Product Advertising API

Developer Guide

API Version 2011-08-01

amazon Product Advertising API

Product Advertising API: Developer Guide

Copyright © 2011 Amazon.com and its affiliates. All rights reserved.

Table of Contents

Welcome

Topics

This is the *Product Advertising API Developer Guide*. This section describes who should read this guide, how the guide is organized, and other resources related to the Product Advertising API.

Audience

This guide is intended for developers who want to build an e-commerce storefront that sells items listed on www.amazon.com, or an application that helps others build e-commerce storefronts.

Required Knowledge and Skills

Use of this guide assumes you are familiar with the following:

- XML (For an overview, go to the *W3 Schools XML Tutorial*)
- Basic understanding of web services (For an overview, go to the *W3 Schools Web Services Tutorial*)

How This Guide Is Organized

This guide is organized into several major sections described in the following table.

Information	Relevant Sections
Describes the changes in this document since its last publication.	What's New (p. 4)
General information about the Product Advertising API	Introduction to Product Advertising API (p. 5)

Information	Relevant Sections
Conceptual information about the Product Advertising API	Terminology and Basic Concepts (p. 21)
Programming Reference--Provides task-oriented descriptions of how to use and implement Product Advertising API operations.	Programming Reference (p. 7)
API Reference--A reference that describes Product Advertising API operations, response groups, locales, and parameter constraints.	API Reference (p. 176)
Typographic and symbol conventions	Document Conventions (p. 451)

Product Advertising API Resources

The following table lists related resources that you'll find useful as you work with this service.

Resource	Description
Product Advertising API Getting Started Guide	The Getting Started Guide provides a quick introduction to getting set up and generating requests to the Product Advertising API.
Product Advertising API Release Notes	The Release Notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
Product Advertising API Developer Resource Center	A central starting point find documentation, code samples, release notes, and other information to help you build innovative applications with the Product Advertising API.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse, etc.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

Other resources are the locales' primary web pages for information about Product Advertising API:

Resource	Description
Product Advertising API information for CA	The primary CA web page for information about Product Advertising API.
Product Advertising API information for DE	The primary DE web page for information about Product Advertising API.
Product Advertising API information for FR	The primary FR web page for information about Product Advertising API.

Resource	Description
Product Advertising API information for JP	The primary JP web page for information about Product Advertising API.
Product Advertising API information for UK	The primary UK web page for information about Product Advertising API.
Product Advertising API information for US	The primary US web page for information about Product Advertising API.

What's New

This What's New is associated with the 2011-08-01 version of the Product Advertising API. This guide was last updated on July 26, 2011.

The following table describes the important changes since the last release of the *Product Advertising API Developer Guide*.

Change	Description	Release Date
Changes to Operations and Response Groups	As part of our efforts to streamline the Product Advertising API, we will be making changes to the currently supported operations and response groups. This guide will be updated to reflect these changes in the next few weeks. For more details about these changes, go to Changes to the Product Advertising API.	26 July 2011

Introduction to the Product Advertising API

Topics

This introduction to Product Advertising API gives you a detailed summary of this web service. After reading this section, you should have a good idea of what it offers and how it can fit in with your business.

Overview of Product Advertising API

Amazon has spent over ten years and hundreds of millions of dollars developing a world-class web service that millions of customers use every day. As a developer, you can build Product Advertising API applications that leverage this robust, scalable, and reliable technology. You get access to much of the data that is used by Amazon, including the items for sale, customer reviews, seller reviews, as well as most of the functionality that you see on www.amazon.com, such as finding items, finding similar items, displaying customer reviews, and product promotions. In short, Product Advertising API operations open the doors to Amazon's databases so that you can take advantage of Amazon's sophisticated e-commerce data and functionality. Build your own web store to sell Amazon items or your own items.

Best of all, Product Advertising API is free. By signing up to become a Product Advertising API developer, you join the tens of thousands of developers who are already realizing financial gains by creating Product Advertising API-driven applications and web stores. In 2006, Product Advertising API developers sold well over $600 million worth of items. Would you like a percentage of that revenue?

Business Model

E-commerce is the practice of conducting business over the Internet. This guide explains in detail how you can use Product Advertising API operations to create storefronts in which you enable Internet customers to search for your items, see pictures of them, find related items, get customer reviews, and purchase items.

With e-commerce, the barrier of distance between the shopper and the store goes away: the local video store must compete with stores across the country. E-commerce levels the playing field: the web site of

an individual seller can appear as sophisticated and intoxicating as that of a major retailer. Product Advertising API is your opportunity to enter the world market where patronage is not limited by the size of your storefront, foot traffic or locality. Welcome to the world of Product Advertising API e-commerce.

Architectural Overview of Product Advertising API

How Product Advertising API works

1	Your application uses the Product Advertising API to supply item descriptions and images, and customer and seller reviews, which you present to your customers.
2	Customers shop on your web site.
3	When the customer is ready to purchase the items in their e-commerce shopping cart, your application sends an HTML form to Product Advertising API and Amazon completes the purchase by getting purchase information, such as payment method and shipping address, and then Amazon fulfills the order by shipping the items.

Programming Guide

The Programming Guide provides task-oriented descriptions of how to use and implement Product Advertising API operations. For a complete description of these operations, refer to the API Reference (p. 176).

The following table describes the sections in the programming guide.

Section	Description
Getting Set Up (p. 8)	Describes the tasks you must complete before using Product Advertising API.
E-Commerce and Web Services (p. 10)	Introduces technologies central to Product Advertising API, such as e—commerce, WSDL, and schemas.
Product Advertising API Terminology and Basic Concepts (p. 21)	Introduces terms and concepts central to Product Advertising API.
Visual Introduction to Product Advertising API (p. 28)	Introduces Product Advertising API functionality by visually comparing it to functionality seen on Amazon's retail web site.
Organization of Items for Sale on Amazon	Describes how items are organized in Amazon databases.
Requests (p. 47)	Introduces you to making Product Advertising API requests and processing Product Advertising API responses.
Finding Items to Buy (p. 88)	Describes the operations and response groups required to find items to buy.
Motivating Customers to Buy (p. 110)	Describes the operations and response groups required to motivate customers to buy. Topics include displaying images, customer review, and seller reviews.
Returning Price And Availability Information (p. 137)	Describes the operations and response groups required to get pricing and availability information of the items for sale.
Working With Remote Shopping Carts (p. 146)	Describes the operations and response groups required to create a shopping cart, add items to it, and modify the number of items in the cart.

Section	Description
Purchasing the Items in a Remote Shopping Cart	Describes the operations and response groups required to purchase the items in a shopping cart.
Handling Errors and Troubleshooting Applications (p. 162)	Describes errors returned by Product Advertising API.
Best Programming Practices (p. 171)	Provides programming tips to maximize your Product Advertising API application's functionality.
Locale Considerations (p. 173)	Addresses locale—specific issues.

Getting Set Up

Topics

To make money using the Product Advertising API to advertise Amazon products in conjunction with the Amazon Associates program, you need to register to become a Product Advertising API developer and join the Amazon Associates Program.

Note

For instructions on how to register to become a Product Advertising API developer and join the Amazon Associates Program, go to the Product Advertising API Getting Started Guide. The remaining tasks that you complete are described in this section.

Becoming a Product Advertising API Developer

The Product Advertising API allows developers to advertise products from all Amazon sites:

- http://www.amazon.com
- http://www.amazon.ca
- http://www.amazon.co.uk/
- http://www.amazon.de
- http://www.amazon.fr
- http://www.amazon.co.jp

To become a Product Advertising API developer, sign up at the page for your locale:

Locale	Developer Sign-up URL
Canada	https://associates.amazon.ca/gp/flex/advertising/api/sign-in.html
France	https://partenaires.amazon.fr/gp/flex/advertising/api/sign-in.html

Locale	Developer Sign-up URL
Germany	https://partnernet.amazon.de/gp/flex/advertising/api/sign-in.html
Japan	https://affiliate-program.amazon.com/gp/flex/advertising/api/sign-in-jp.html
United Kingdom	https://affiliate-program.amazon.co.uk/gp/flex/advertising/api/sign-in.html
United States	https://affiliate-program.amazon.com/gp/flex/advertising/api/sign-in.html

When you sign up, you create a Amazon customer account in the appropriate locale, then provide your account information and review and accept the license agreement.

Becoming an Associate

To become an Associate, go to the Amazon Associates main page for your locale:

Locale	URL
Canada	https://associates.amazon.ca/
France	https://partenaires.amazon.fr/
Germany	http://partnernet.amazon.de
Japan	https://affiliate.amazon.co.jp/
United Kingdom	https://affiliate-program.amazon.co.uk
United States	http://affiliate-program.amazon.com/

Getting Your Key Pair

Once you have registered as an Amazon Associate, you will need a key pair, consisting of an access Key ID and a secret key. The Product Advertising API uses the Amazon Web Services infrastructure for generating key pairs, as well as for other Product Advertising API resources. You can get your keys at http://aws-portal.amazon.com/gp/aws/developer/account/index.html?action=access-key.

Once you have obtained your key pair, you can make calls to the Product Advertising API in all Amazon locales.

Reading the Licensing Agreement

You will need to review and accept the terms and conditions of the license agreement to become a Product Advertising API developer. To read the Product Advertising API' licensing agreement, go to the license agreement link for your locale:

Locale	License Agreement URL
Canada	https://associates.amazon.ca/gp/advertising/api/detail/agreement.html
France	http://partenaires.amazon.fr/gp/advertising/api/detail/agreement.html
Germany	https://partnernet.amazon.de/gp/advertising/api/detail/agreement.html
Japan	https://affiliate.amazon.co.jp/gp/advertising/api/detail/agreement.html

Locale	License Agreement URL
United Kingdom	https://affiliate-program.amazon.co.uk/gp/advertising/api/detail/agreement.html
United States	https://affiliate-program.amazon.com/gp/advertising/api/detail/agreement.html

Note

If you plan to use the Product Advertising API to advertise Amazon products from a locale other than the one you signed up in, please be sure to review the license agreement for that locale. The terms and conditions for each locale apply to any use of the Product Advertising API in that locale.

Please review the Application Best Practices Guide to make sure your application is compliant, scalable, and efficient.

Important

In addition to the Product Advertising API License Agreement, be sure to read your locale's Associates Program Operating Agreement for more details on usage guidelines, policies, and requirements.

E-Commerce and Web Services

Topics

This chapter is for readers who are new to the concepts of e-commerce and web services. If you are familiar with these topics, proceed to the next chapter.

This chapter provides an overview of e-commerce, web services and associated technologies.

What is E-Commerce?

E-commerce (Electronic Commerce) is the practice of using computers to buy, sell, and market goods and services across a network, such as the Internet.

This book assumes that you have at least some familiarity with buying or selling items over the Internet. If this is not the case for you, go to www.amazon.com, find an item to buy and proceed a few steps into the purchase procedure. For a basic understanding of e-commerce, go to www.amazon.com.

In Product Advertising API, e-commerce is facilitated by web services.

What is a Web Service?

This book assumes that you are familiar with the notion of using your computer (the client) to request that some other computer (the server's) perform some task and respond over the Internet. The work a server

does is called a service. The service might be returning a weather forecast, or, in the case of Product Advertising API, returning information about items for sale on Amazon. Some of these request and response interactions over the Internet are considered web services provided by a web server.

A web service is any piece of software that uses REST and/or a standardized XML messaging system, described by a WSDL (Web Service Definition Language) (pronounced "wiz-dal"), to exchange data between applications or systems. A web service must have some simple mechanism for interested parties to locate the service and its public interface using standard network protocols, such as, but not limited to, SOAP over HTTP. Software applications written in various programming languages and running on various platforms can use web services transparently to exchange data over computer networks, like the Internet, because the WSDL serves as the definition of the language used by the computers.

In Product Advertising API:

- Requests and responses occur across the Internet
- Client and server use REST, SOAP, or XML as the means of communication
- Client and server agree on the grammar and syntax used in the requests and responses by specifying a WSDL
- Requests and responses are not tied to a single operating system or programming language.

A simple example of a Product Advertising API request is one that uses the `ItemSearch` operation. This request asks Amazon's web servers to find descriptions of items in its data bases. The web servers carry out their service by finding the item descriptions and then sending those back to the requester.

All Product Advertising API requests are based on REST or SOAP and all of the responses are based on XML.

Note

For more information about REST, see Anatomy Of a REST Request (p. 48).

What is XML?

XML (Extensible Markup Language) is a standard defined by the World Wide Web Consortium (W3C). XML uses tags (identifiers enclosed in brackets, for example, <para>) to label content in text documents. These tags, collectively called the "markup," can be read by humans and computers. XML tags, like SGML tags, encode not only the meaning of the content but also its structure.

XML looks like HTML but XML has nothing to do with the display of the content, which is the central focus of HTML. The following XML tags label the content they enclose as a paragraph.

```
<para>This is a paragraph.</para>
```

The <para> tag has nothing to do with the display of the sentence. The tag, for example, does not take attributes such as font, size, or style.

Unlike HTML, there is not a single set of tags used in all XML documents. XML enables designers to create their own set of tags that are appropriate to their business. For example, Product Advertising API encodes its APIs using an XML document called a WSDL. Other companies create their own XML tags to define their APIs. The tags in these two WSDLs most likely would be completely different. The syntax and rules by which the XML elements are defined in the WSDL, however, are the same. Because a web server handles multiple requests that implement multiple WSDLs, each request must specify the WSDL it is using..

XML Syntax

XML tools enforce XML syntax. XML syntax is very similar to HTML syntax, except, like XHTML, the syntax is strictly enforced. The syntax is:

- Tags are enclosed within angle brackets, for example, <para>
- Opening tags must be paired with closing tags, for example, <para>Sentence</para>
- Opening and closing tags must be nested correctly, for example, the following example is well formed

```
<note><para>Sentence</para></note>
```

The following example is not well formed

```
<note><para>Sentence</note></para>
```

If these syntax rules are followed, the XML document is said to be well formed. That does not mean, however, that the tags in the XML document are necessarily the ones defined in the associated WSDL. If, for example, the WSDL defines <para>, an XML document that uses <Para> would cause an error because "Para" is not defined in the WSDL. When an XML document is well formed and the tags it uses conform to tags defined in the WSDL, the document is said to be valid. Product Advertising API responses always contain valid XML.

XML Structures

Some data in XML documents are one dimensional, for example, <para>Sentence</para>. Some data, however, is structured, as shown in the following XML snippet.

```
<SubTotal>
  <Amount>2998</Amount>
  <CurrencyCode>USD</CurrencyCode>
  <FormattedPrice>$29.98</FormattedPrice>
</SubTotal>
```

This example shows that the subtotal is comprised of three pieces of data: *`Amount`*, *`CurrencyCode`*, and *`FormattedPrice`*. The example also shows that there can be a hierarchy of tags. In this example, Amount is a child tag of Subtotal. These structures represent arrays of values.

Using XML

So, what do you do with the XML response? The good news is that the Product Advertising API returns a well-defined set of tags for each kind of request. That means that you can use parsing mechanisms to pull out of the XML responses the data that you would like to submit in a second request, or, display on a web page. For instance, in the preceding example, you might parse the XML document to find FormattedPrice so that you could display the $29.98 on a web page that shows an item for sale.

XML is not only used to create requests and responses, it is also used to create a WSDL, which defines the XML tags.

What is a WSDL?

Topics

The goal of this section is to give you enough information so that you can read and use the Product Advertising API WSDL. You typically read a WSDL to understand value types, operation definitions, and request and response formats.

A WSDL (Web Service Description Language) is an XML document that defines the operations, parameters, requests, and responses used in web service interactions. You can think of a WSDL as the contract that defines the language and grammar used by web service clients and servers. When you look at the Product Advertising API WSDL, for example, you find in it all of the Product Advertising API operation names, parameters, request and response structures.

There is not a single WSDL. Product Advertising API, for example, has many different versions of its WSDL—the latest one and all of its previous versions. Not only can one company use different versions of a WSDL, every company can use its own WSDL based on its own APIs or business metrics. For that reason, web service requests must identify the WSDL they use so the web servers know how to interpret the requests.

In practice, Product Advertising API developers use the same version of the Product Advertising API WSDL in every request in a session. Over time, they might send requests to the latest Product Advertising API WSDL. The Product Advertising API WSDL is upgraded regularly.

Restricted Parts of the WSDL

The WSDL defines all Product Advertising API operation requests and responses. The majority of the WSDL is generic. There are, however, some small portions of the WSDL that are partner specific. That is, the use of some Product Advertising API operations is restricted to specific Amazon partners. Those operations are marked as restricted by a notation similar to the following.

```
<xs:element name="RestrictedOperation" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:appinfo>
      <aws-se:restricted>
        <aws-se:excludeFrom>public</aws-se:excludeFrom>
        <aws-se:excludeFrom>partner</aws-se:excludeFrom>
      </aws-se:restricted>
    </xs:appinfo>
```

If you try to use a restricted operation and you are not the Amazon partner, Product Advertising API returns an error message.

Anatomy of a WSDL

Topics

Typically, the Binding and Service segments do not change from one WSDL release to the next. In the Definitions segment, only the WSDL version changes. For that reason, when you read a WSDL, you will typically spend most of your time reading the Request and Response Definitions segments.

Definitions

Topics

The Definitions section of the WSDL defines the namespaces used throughout the WSDL, and the name of the service, as shown in the following snippet of the Product Advertising API WSDL.

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://ecs.amazonaws.com/
  AWSECommerceService/2011-08-01"
  targetNamespace="http://ecs.amazonaws.com/
  AWSECommerceService/2011-08-01">
```

This example shows that the:

- Default namespace is xmlns="http://schemas.xmlsoap.org/wsdl/"
- SOAP namespace used is xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
- Schema used is xmlns:xs="http://www.w3.org/2001/XMLSchema"
- Product Advertising API WSDL namespace is "http://ecs.amazonaws.com/AWSECommerceService/2011-08-01"

  The date at the end is the version number. It is the date the WSDL became public.
- TargetNamespace is "http://ecs.amazonaws.com/AWSECommerceService/2011-08-01"

  The TargetNamespace is an XML schema convention that enables the WSDL to refer to itself (as the target). The TargetNamespace value is the Product Advertising API WSDL namespace

Namespaces

Namespaces are collections of parameters and operations in which their names are unique. The advantage of using namespaces is that the WSDL can define terms, like string, just by referring it its namespace, xs. Also, prepending the namespace to a parameter ensures that there is no danger of name collisions.

Each namespace declaration starts with "xmlns:" (XML namespace:) and is followed by the abbreviation for the namespace. For example, in the following namespace declaration, xs becomes the abbreviation for the URL of the schema.

```
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

Throughout the remainder of the WSDL you will see parameters defined in terms of namespace abbreviations, for example:

```
type="xs:string"
ref="tns:HTTPHeaders"
```

These abbreviations provide the namespace in which the parameters are defined.

Versioning

Product Advertising API enables you to specify the version of the WSDL you want to use. This functionality ensures that future enhancements and changes to Product Advertising API WSDLs will not be intrusive to your applications. For example, when Product Advertising API adds new elements to its WSDL, applications that validate against an older Product Advertising API WSDLs will not be affected.

Product Advertising API WSDL version names are based on the date that they become active. The version of the WSDL is specified in the Product Advertising API WSDL namespace declaration. In the preceding example, the version of the WSDL is 2011-08-01.

```
xmlns:tns="http://ecs.amazonaws.com/
  AWSECommerceService/2011-08-01"
```

In reality, the date, here, is the WSDL's file name.

The *`AWSECommerceService`* directory contains all of the Product Advertising API WSDL versions. You use the *`Version`* parameter in REST requests to specify the version of the WSDL you want to use. The default version is 2005-10-05. If you want to use a different WSDL version, including the latest, you must specify it in each request, for example, in REST.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService
&AWSAccessKeyId=[AWS Access Key ID]
&Operation=ItemSearch&
SearchIndex=Books&
Author=Steve%20Davenport&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Note

This version of the *Product Advertising API Developer Guide* describes all of the functionality of the WSDL as of the guide's publication date. To read about older WSDLs, including the default WSDL, refer to the older versions of this guide listed in the AWS Resource Center. Go to http://aws.amazon.com/resources.

SOAP requests always specify a namespace, which includes the WSDL version. To avoid problems due to future WSDL changes, be sure to specify a WSDL version in your SOAP application.

Request Definitions

The Request Definitions segment of the WSDL defines Product Advertising API operation requests, as shown in the following WSDL snippet.

```
<xs:complexType name="ItemSearchRequest">
  <xs:sequence>
  <xs:element name="Actor" type="xs:string" minOccurs="0" />
  <xs:element name="Artist" type="xs:string" minOccurs="0" />
  <xs:element name="Availability" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
```

```
        <xs:enumeration value="Available" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element ref="tns:AudienceRating" minOccurs="0" maxOccurs="unbounded" />
```

This snippet shows some of `ItemSearch` 's input parameters, including *`Actor`*, *`Artist`*, *`Availability`*, and *`AudienceRating`*. The element declarations specify that these parameters are valid in an `ItemSearch` request. Most of the parameters in this example are strings. The type of one, however, *`Availability`*, is a variation on the base class, string. In this case, the variation puts a restriction on the strings that can be valid values for *`Availability`*. For that reason, the restriction keyword is used. The restriction is that the valid values for *`Availability`* are defined by an enumeration. The enumeration, however, has only one valid value, "Available," which means that the parameter, *`Availability`* can be set to only one value.

`minOccurs` refers to the minimum number of times the parameter must appear in an `ItemSearch` request. If the value is zero, the associated parameter is optional. If the value is 1, the associated parameter is required to be included once in every request involving that operation. The default value is 1, that is, if *`minOccurs`* is not included in an element declaration, *`minOccurs`* is 1.

`maxOccurs` defines the maximum number of times the parameter can appear in a request. The default is 1, that is, if *`maxOccurs`* is not included in an element declaration, *`maxOccurs`* is 1 and the parameter can only appear once in a request. In the preceding example, *`maxOccurs`* is "unbounded," which means that the *`AudienceRating`* parameter can appear any number of times in an `ItemSearch` request.

In the preceding example, the parameter types are declared to be simpleTypes. A simple type cannot have child elements or attributes. Complex types can. In practice, any parameter that can take multiple values, such as an array, must be defined as a complex type.

The following snippet shows an example of a complex type.

```
<xs:element name="ItemSearch">
  <xs:complexType>
  <xs:sequence>
    <xs:element name="MarketplaceDomain" type="xs:string" minOccurs="0" />
    <xs:element name="AWSAccessKeyId" type="xs:string" minOccurs="0" />
    <xs:element name="SubscriptionId" type="xs:string" minOccurs="0" />
```

This definition snippet shows three of the parameters that can be part of an `ItemSearch` request.

Response Definitions

The response section defines the responses returned by default by each operation. The following snippet shows some of the specifications of an `ItemSearch` response.

```
<xs:element name="ItemSearchResponse">
  <xs:complexType>
    <xs:sequence>
    <xs:element ref="tns:OperationRequest" minOccurs="0" />
    <xs:element ref="tns:Items" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

The response section shows that an `ItemSearch` response contains two optional (*`minOccurs`*=0) elements, *`OperationRequest`* and Items. Both of these elements are references (*`ref`*=), which means that they are defined further down in the WSDL.

Further down in the WSDL, *`OperationRequest`* is defined, as follows.

```
<xs:element name="OperationRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:HTTPHeaders" minOccurs="0" />
      <xs:element name="RequestId" type="xs:string" minOccurs="0" />
      <xs:element ref="tns:Arguments" minOccurs="0" />
      <xs:element ref="tns:Errors" minOccurs="0" />
      <xs:element name="RequestProcessingTime" type="xs:float" minOccurs="0"
maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

This definition also contains several references. One is *`Arguments`*, which is defined further down in the WSDL. To fully understand the definition of the parts of a request, you keep digging down through the layers of refs. You know that you have reached the end of the definition hierarchy when you no longer have "ref" in the element's definition. Instead, the element definition will have a "name," the name of the element, and "type," which specifies the element's type. The type will be a base type, such as, string, which is defined in the schema (xs:), as shown.

```
<xs:element name="RequestId" type="xs:string" minOccurs="0" />
```

This line defines *`RequestId`* to be of type string, which is defined by the W3C schema.

When you look at a sample response, shown in the following example, you can see how the definition of *`RequestId`* is carried out.

```
<ItemSearchResponse xmlns="
http://ecs.amazonaws.com/AWSECommerceService/2011-08-01">
...
  <OperationRequest>
  ...
    <RequestId>0VFY0HFBRTJGRE6KES74</RequestId>
```

First, you see that the value for *`RequestId`* is string. Secondly, the name of the element is *`RequestId`*. Third, you can see, in the XML hierarchy, how the definition of *`RequestId`* is nested inside the *`OperationRequest`* element, which is nested inside of *`ItermSearchResponse`*. Remember, it was the "ref" keyword that created the nesting in the WSDL.

Response Group Definitions

Response groups, except in the case of a request error, always form part of a response. Each response group is defined in the WSDL. The following snippet from the WSDL shows the definition of the Images response group (p. 50).

```
<xs:complexType name="Image">
  <xs:sequence>
    <xs:element name="URL" type="xs:string" />
    <xs:element name="Height" type="tns:DecimalWithUnits" />
```

```
    <xs:element name="Width" type="tns:DecimalWithUnits" />
    <xs:element name="IsVerified" type="xs:string" minOccurs="0" />
  </xs:sequence>
  </xs:complexType>
```

As you can see, the Image response group returns the elements *URL*, *Height* (height of the image), *Width*, and *IsVerified*. All are required in the response except *IsVerified*, which is optional (*minOccurs*=0). You can see how these elements are displayed in a response.

```
<MediumImage>
  <URL>http://ec1.images-amazon.com/images/P/
    B00005TNFV.01._SCMZZZZZZZ_.jpg</URL>
  <Height Units="pixels">140</Height>
  <Width Units="pixels">99</Width>
```

This example shows how *URL*, *Height*, and *Width* are child elements.

PortType

The association between operation names and their request and response definitions is created by the PortType element in the WSDL, for example:

```
<portType name="AWSECommerceServicePortType">
...
  <operation name="ItemSearch">
    <input message="tns:ItemSearchRequestMsg" />
    <output message="tns:ItemSearchResponseMsg" />
  </operation>
```

In this example, the operation, `ItemSearch`, is associated with its request and response definitions, *ItemSearchRequestMsg* and *ItemSearchResponseMsg*. The keywords, input and output, identify the operation's request and response definitions, respectively.

Binding

The binding segment of the WSDL specifies how operation requests and responses, defined in *PortType*, are actually transmitted over the wire using underlying transport protocols. While this is an interesting portion of the WSDL, it is a section that rarely changes so you need not pay much attention to it.

Binding values include HTTP GET, HTTP POST, and SOAP. SOAP is not tied to a specific transport. SMTP, FTP, HTTP are just some of the options that can transport a SOAP request. HTTP, however, is most commonly used. While both HTTP GET and HTTP POST are allowed, HTTP POST is preferred because many servers place character limits on HTTP GET requests.

Product Advertising API uses SOAP, as shown in the following Product Advertising API WSDL snippet.

```
<binding name="AWSECommerceServiceBinding"
  type="tns:AWSECommerceServicePortType">
...
  <operation name="ItemSearch">
    <soap:operation soapAction="http://soap.amazon.com" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
```

```
      <soap:body use="literal" />
    </output>
  </operation>
```

This binding shows that Product Advertising API uses two SOAP extensions: soap:operation and soap:body.

The soap:operation element specifies that the Product Advertising API operation, `ItemSearch` , in this case, is bound to a specific SOAP implementation. The soapAction attribute specifies that the SOAPAction HTTP header is used to identify the Product Advertising API service, which is the URI value of soapAction, http://soap.amazon.com.. soapAction enables Amazon web servers to determine the intent of the SOAP request without having to examine the message portion of the SOAP payload. Specifying this URI is required to access Product Advertising API web servers.

The soap:body element specifies the input and output details. The value in the Product Advertising API WSDL is "literal," which means that instead of encoding the input and output as a SOAP struct, a literal XML document is used. You have seen that Product Advertising API responses are XML documents.

Service

The Service segment of the WSDL specifies the web service used, which, in this case, is Product Advertising API, as shown in the following WSDL snippet:

```
<service name="AWSECommerceService">
  <port name="AWSECommerceServicePort"
    binding="tns:AWSECommerceServiceBinding">
    <soap:address location= "http://soap.amazon.com/onca/soap?Service=
AWSECommerceService" />
  </port>
</service>
```

This information changes very rarely and so you need not pay much attention to it.

Every Product Advertising API request includes this service declaration, as shown in the following example.

```
http://ecs.amazonaws.comonca/xml?
Service=AWSECommerceService&
Operation=ItemSearch&
SearchIndex=Books&
Keywords=Saving%20Miss%20Oliver's
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

WSDL Location

Product Advertising API publishes its API through its WSDL. You can find the latest WSDL at http://ecs.amazonaws.com/AWSECommerceService/AWSECommerceService.wsdl

The WSDL contains all the API endpoints. To select the required endpoint, see your SOAP framework.

What is a Schema?

A schema is similar to a WSDL in that both are XML documents. Whereas the WSDL defines the web service language used by computers to converse, the schema defines the data types used in the WSDL.

You do not have to create schemas to use Product Advertising API. Those have already been created. It is helpful, however, to understand schemas so that you can determine the data types returned in responses.

The W3C defines the base data types, which include, for example, int, string, and float. While these data types are useful, they are not very descriptive. For example, defining every occurrence of text in an XML document as being of type string hides the differences between text that is, for example, a paragraph and a note. In such an application where paragraphs and notes are used, a schema would contain an extension of the string base class so that paragraph (<para>) and note (<note>) could be used as tags in XML documents.

Schemas enable you to create your own data types for the purpose of identifying the content in an XML document. All data types that you create must be based on the base data types defined by the W3C. This is the schema namespace defined in the WSDL example.

```
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

The data types that can be created are either simple or complex. Complex types can have sub elements and attributes; simple types cannot.

In the WSDL section of this chapter, you saw that complex types are declared as complexType. In the following example, the element, SearchBinSet, is defined as having two child elements, Bin and NarrowBy.

```
<xs:element name="SearchBinSet">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:Bin" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="NarrowBy" type="xs:string" use="required" />
  </xs:complexType>
</xs:element>
```

The NarrowBy attribute is defined in terms of a base type, string. The Bin parameter, however, is not. That means that Bin is defined elsewhere in the schema. Schema syntax, like WSDL syntax, calls for using the keyword "ref" if the element is defined elsewhere in a schema.

The next step in understanding Bin would be to see its definition in the schema. It might be that all of Bin's sub elements are defined by base types. In that case, the research would be over; you would have the full definition of SearchBinSet and it's child elements. If, however, Bin contains more "ref's," you would repeat the search for the child elements until you reached element type definitions that used base types, as shown in the following example.

```
name="BinItemCount" type="xs:string"
```

Schema Locations

Product Advertising API provides schemas for validating the XML in SOAP requests and for specifying item attribute types in responses. XML schemas are available by version and by locale. The following table shows the location of the latest Product Advertising API XML schema, by locale.

Locale	URL
US	http://ecs.amazonaws.com/AWSECommerceService/AWSECommerceService.xsd
UK	http://ecs.amazonaws.com/AWSECommerceService/UK/AWSECommerceService.xsd

Locale	URL
DE	http://ecs.amazonaws.com/AWSECommerceService/DE/AWSECommerceService.xsd
JP	http://ecs.amazonaws.com/AWSECommerceService/JP/AWSECommerceService.xsd
FR	http://ecs.amazonaws.com/AWSECommerceService/FR/AWSECommerceService.xsd
CA	http://ecs.amazonaws.com/AWSECommerceService/CA/AWSECommerceService.xsd

To access an older schema

- Insert the schema version between the last two path elements.
  For example, to retrieve the 2011-08-01 schema, use the following URI.`http://ecs.amazonaws.com/AWSECommerceService/2011-08-01/AWSECommerceService.xsd`. This specification would be for the 2011-08-01 version of the US schema. For the schema used in a different locale, insert the locale name after the version number. For example, to retrieve the 2011-08-01 JP schema, use the following URI.`http://ecs.amazonaws.com/AWSECommerceService/2011-08-01/JP/AWSECommerceService.xsd` If the locale is omitted, the US schema is used by default.

Product Advertising API Terminology and Basic Concepts

Topics

This chapter introduces you to Product Advertising API terminology and concepts. Many of the concepts introduced in this chapter are explored in greater depth in later chapters. The concepts are briefly presented here so that you have a broad understanding of Product Advertising API and have a context in which you can place the specifics that are presented in later chapters.

The Marketplace

Amazon started in a two-bedroom house in Seattle. The location was chosen because of its proximity to Ingram, a book wholesaler and because of the wealth of talented programmers in the area. In the garage of that house, Jeff Bezos, the CEO of Amazon, stockpiled books and set up three Sun Microsystems computers to take book orders online. In 1995, Jeff opened his online bookstore to the public. At that time, there was only one kind of seller, in fact, there was only one seller, Jeff . He only sold one type of product: books. Soon after the start of Amazon, the garage became too small to run the business.

Since 1995, the number of sellers has grown, the types of sellers have grown, and the product offerings have grown. All of these sellers come together to sell their items on www.amazon.com, in what is called the "marketplace."

Individual sellers cannot have their own storefront in the marketplace. Pro Merchant Sellers and Merchant@ vendors can. Individual sellers can, however, set up their own storefronts outside of the marketplace. These storefronts can be accessed through www.amazon.com. Amazon refers to these stores as zShops.

Product Advertising API requests cannot access zShops. This usually has little impact on product searches because zShop owners can also list their items for sale in the Amazon marketplace.

What Is the Product Advertising API?

The Product Advertising API gives you much of the functionality that you see in action on Amazon's retail web site, *http://www.amazon.com*. That functionality includes:

- Finding items to buy.

  These items are for sale by Amazon or other merchants.
- Finding information about those items.

  This item information includes such things as the titles of tracks on a CD, the rating of a movie, the kind of metal used in a piece of jewelry, accessories that go with an item, the author of a book, or the composer of a piece of music.
- Getting customer reviews of items.

  Show customers what others think about the items on sale.
- Create a fully-functional shopping cart.

  Add items that are immediately available or ones that will become available in the future, such as in a pre-sale of a book.
- Adding, removing, and otherwise modifying the items in the shopping cart.

  Have full control over the contents of their shopping cart.
- Getting information about the company selling the item.

  Show customers what others think about the merchant selling the item.
- Finding similar items for sale.

  Generate additional sales by suggesting other items similar to the ones the customers are buying.
- Purchasing the items in the shopping cart.

  Once the customer decides to buy the contents in their shopping cart, Amazon takes care of the shipping, payment, and order fulfillment, or notifies you to take care of the same.
- Find items on a friend's wishlist, wedding registry or baby registry and purchase those items.

Product Advertising API offers you the tools to create a complete, rich online shopping experience for customers.

How Do I Make Money Using the Product Advertising API?

You can make money when you use the Product Advertising API to advertise Amazon products in conjunction with the Amazon Associates Program. How? You earn referral fees when you join the Amazon Associates program and the users you refer to Amazon sites buy qualifying products.

You need to get an Amazon Associates account and ensure that you include your associate tag in API requests. That way the URLs returned by the API contain your Associate tag and when a user visits an Amazon site through a tagged link and buys a product, the developer earns referral fees. For more information about getting an Amazon Associates account, see Becoming an Associate (p. 9).

International Locales

Product Advertising API hosts multiple locales so that developers around the world can take advantage of Amazon's multi-language support. Each locale uses the language, customs, and formats, such as date, time, and money formats, particular to that locale. You can see how this plays out by viewing the homepages for Amazon's different locales:

- Canada—www.amazon.ca
- Germany—www.amazon.de
- France—www.amazon.fr
- Japan—www.amazon.co.jp
- UK—www.amazon.co.uk
- US—www.amazon.com

Items for Sale

www.amazon.com lists hundreds of thousands of items for sale. Most of these items are warehoused and sold by Amazon or other large merchants, such as Nordstrom. All of these items are part of the Amazon marketplace.

In addition to large merchants selling items, individuals and companies also sell items that can be found using www.amazon.com. Some of these sellers have their own storefront on Amazon and their items can only be found in their store. Most sellers, however, list their items in the Amazon marketplace as well as in their stores.

Product Advertising API operations only enable you to retrieve items that are stored by Amazon. Product Advertising API does not enable you to add items for sale to Amazon. Sellers, such as Amazon Advantage sellers, who do add items to Amazon's catalog do so under a separate agreement.

Item Identifiers

All items for sale on www.amazon.com have identifiers. There are two major kinds. The first identifies items regardless of whether or not they can be purchased:

- **ASIN—**Amazon Standard Item Number
  An alphanumeric token that uniquely identifies items in the Amazon marketplace.
  All items in the marketplace have an ASIN. This is by far the most common identifier. ASINs are used by the majority of Product Advertising API operations.

The second identifies items that can actually be purchased:

- **OfferListingId—**An alphanumeric token that uniquely identifies items in the Amazon marketplace that can be purchased
  All items in the marketplace that can be purchased have an OfferListingId. This is by far the most common identifier for items that can be purchased.

Other Item Identifiers

Amazon has other item identifiers but these are rarer and they can be used in fewer Product Advertising API operations:

- **UPC—**Universal Product Code
  A 12-digit item identifier used in the US and CA locales. The UPC is identifier used in barcodes.
- **EAN—**European Article Number
  A 13-digit equivalent of the UPC that is used in Europe for products and barcodes.
- **JAN—**Japanese Article Number
  The equivalent of the EAN that is used in Japan for products and barcodes.
- **ISBN—**International Standard Book Number
  An alphanumeric token that uniquely identifies a book. A book's EAN is typically set equal to the book's ISBN.
- **SKU—**Stock Keeping Unit
  A merchant-specific identifier for a purchasable good, like a shirt or chair. Amazon's version of the SKU is the ASIN.

  Amazon assigns items with SKUs an ASIN. If you had to search by SKU only, you would have to search each merchant's items independently because SKUs are not unique across all merchants. Several Product Advertising API operations enable you to search by SKU but the merchant must also be identified in the same request.
- **EISBN—**Electronic International Standard Book Number
  A token that uniquely identifies a digital book.

All items in the Amazon marketplace have an ASIN even if they also have one of the preceding identifiers. Only a few Product Advertising API operations can use these identifiers. That functionality is presented as a convenience function. You would typically only use one of these identifiers if you knew it but did not know the item's ASIN.

The validity of these identifiers varies by search index, as shown in the following table.

Search Index	UPC	EAN	SKU
Apparel	Y	Y	Y
Automotive	Y	Y	Y
Automotive	Y	Y	Y
Baby	Y	Y	Y
Beauty	Y	Y	Y
Books	Y	Y	Y
Classical	Y	Y	Y
DigitalMusic	Y	N	N
DVD	Y	Y	Y

Search Index	UPC	EAN	SKU
Electronics	Y	Y	Y
HealthPersonalCare	Y	Y	Y
HomeGarden	Y	Y	Y
Industrial	Y	Y	Y
Jewelry	Y	Y	Y
Kitchen	Y	Y	Y
Magazines	N	Y	Y
Miscellaneous	Y	Y	Y
Music	Y	Y	Y
MusicalInstruments	Y	Y	Y
MusicTracks	N	N	N
OfficeProducts	Y	Y	Y
OutdoorLiving	Y	Y	Y
PCHardware	Y	Y	Y
PetSupplies	Y	Y	Y
Photo	Y	Y	Y
Software	Y	Y	Y
Tools	Y	Y	Y
Toys	Y	Y	Y
VHS	Y	Y	Y
Video	Y	Y	Y
Watches	Y	Y	Y

The following table shows the valid identifiers by locale.

Locale	Valid Item Identifiers
CA	ASIN, EAN, SKU, EISBN
DE	ASIN, EAN, SKU, EISBN
FR	ASIN, SKU, UPC, EISBN
JP	ASIN, EAN, JAN, SKU, EISBN
UK	ASIN, SKU, UPC, EISBN
US	ASIN, ISBN, SKU, UPC, EISBN

The default value of *IdType* is ASIN. For non-ASIN searches, including searches by ISBN, JAN, SKU, UPC, EAN, and EISBN, a variety of additional parameters become mandatory, including a value for *IdType*.

Summary of Product Advertising API Operations

Topics

In Product Advertising API, the term "operation" is synonymous with the terms function and method. Product Advertising API operations are included in every request and they cause Product Advertising API web servers to take some action, for example, to find items in Amazon's databases or to find similar items. Think of each Product Advertising API operation as a different means of querying Amazon's databases for information. Product Advertising API does not offer any operations that place information on Amazon databases.

The following sections provide an overview of Product Advertising API operations.

Find Items

Use the following Product Advertising API operations to return a list of items that satisfy your search criteria.

Product Advertising API Operation	Description
`ItemSearch`	Find items that are sold on www.amazon.com by merchants and most sellers.
`SimilarityLookup`	Find items that are similar to ones you've already found..

Find Out More About Specific Items

Once you use the Product Advertising API operations in the previous section to get a list of items that match your search criteria, you can use the following Product Advertising API operations to return additional information about specific items.

Product Advertising API Operation	Description
`ItemLookup`	Returns descriptions of specified items.

Shopping Cart

Once customers find items they want to purchase, they put them in a shopping cart. The following Product Advertising API operations enable you to implement a fully-featured e-commerce shopping cart.

Product Advertising API Operation	Description
`CartCreate`	Create a shopping cart and add an item(s).
`CartAdd`	Add items to the shopping cart
`CartModify`	Add to or remove items that are already in a shopping cart.
`CartClear`	Remove all of the items from a shopping cart
`CartGet`	Retrieve the contents of a shopping cart with updated price and availability information.

Other Operations

The remaining operations cover a variety of functions.

Product Advertising API Operation	Description
`BrowseNodeLookup`	Amazon uses browse nodes as a means of organizing the millions of items in their inventory. One browse node, for example, might be "Carving Knives." All items associated with this browse node would have something to do with carving knives. This browse node might be the child of a more general browse node, "Cutlery." `BrowseNodeLookup` enables you to find a browse node, its ancestors and progeny.

Product Advertising API Developer Community

There are many people developing applications using the Product Advertising API. That means that there is a wealth of knowledgeable people who can help you with questions you might have. To connect with other Product Advertising API developers, go to the Product Advertising API Developer Forum at http://developer.amazonwebservices.com/connect/forum.jspa?forumID=9.

Product Advertising API Developer Aids

Topics

Product Advertising API has been around along enough that some developers have created tools that make the job of creating your online, Product Advertising API-driven store easier to build. The following sections provide links to some of those web sites.

Store Builders

Store builder applications create part or all of a web site for an Amazon Associate, Marketplace Seller, or Merchant. Store builders make it easy for less technical vendors or web site owners to create a customized web experience for their customers. The following sites offer store builders or tools that help you enhance your store:

- Associate Engine: http://www.c3scripts.com/amazon

- Amazon Products Feed: http://www.mrrat.com/aws/index.html
- Associates Shop: http://www.associatesshop.filzhut.de/
- Mr. Rat: http://www.mrrat.com

Integration Software Services

Perhaps you want to work with a company to help integrate Product Advertising API into your online stores. The following development groups offer this service:

- Mercent: http://www.mercent.com
- Monsoon Retail: http://www.monsoonretail.com

Visual Introduction to Product Advertising API

Topics

The previous chapter gave an overview of all Product Advertising API operations. Sometimes it is helpful to see them in action. Product Advertising API functionality is implemented on www.amazon.com. This chapter provides a series of screenshots from www.amazon.com. The parts of the web pages that demonstrate Product Advertising API functionality are labeled.

The labels show three kinds of implementations:

- Operations
- Response groups
- Elements in the response

Product Advertising API only provides data; it does not provide HTML. 'We show one way of displaying Product Advertising API on our retail web site, www.amazon.com. You might find a better way.

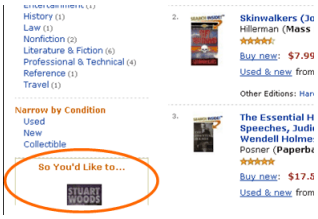
Tools to Find Items

Typically, the first task a customer undertakes is searching for an item. The following web page from www.amazon.com shows the implementation of this task using, in particular, `ItemSearch` and ItemLookup requests.

1	ItemSearch SearchIndex parameter	9	Images response group
2	ItemSearch Keywords parameter	10	CustomerReviews response group
3	BrowseNodeLookup	11	Availability element returned by ItemsSearch Availability parameter
4	ItemLookup	12	SearchBins response group
5	Author ItemSearch parameter	13	ItemSearch Title parameter
6	Offers response group	14	ItemSearch Condition parameter
7	ItemSearch Sort parameter		
8	Child Browsenodes		

The top entry fields show a drop down menu equal to search indices, and a text entry box for a keyword. Both of these parameter values are required for an `ItemSearch` request. The result, shown in this page, displays many of the elements returned by the ItemAttributes response group.

The following figure shows the functionality not supported by Product Advertising API.



Tools to Find Out More About an Item

If the customer clicks on the first image in the previous screenshot, the equivalent of a Product Advertising API `ItemLookup` request is sent. The following page shows the elements returned by such a request

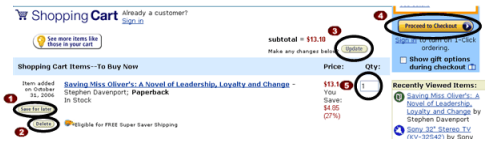
and some of the functionality implemented to move the customer to the next stage, such as getting seller information and adding the item to a Product Advertising API remote shopping cart.

1	MediumImage element	6	Availability element
2	Title element	7	ItemLookup with Images response group
3	Author element	8	TotalItems element
4	FormattedPrice element	9	SmallImage element
5	CartAdd operation	10	SimilarityLookup

The following figure shows the functionality not supported by Product Advertising API.

Tools to Implement a Shopping Cart

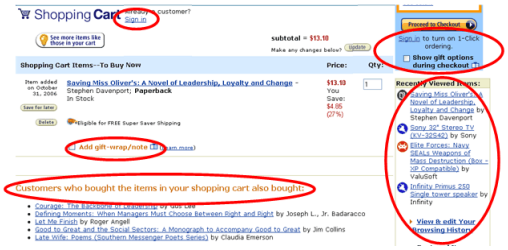
If the customer clicked the "Add to Cart" button in the previous screenshot, their cart would be displayed, as shown in the following figure. Most of the functionality on this page deals with modifying (`CartModify`) the items in the cart.



1	CartModify, Action=SaveForLater	4	PurchaseURL

2	CartModify, Quantity=0	5	CartModify, Quantity=1
3	CartModify		

The following figure shows the functionality not supported by Product Advertising API.



Organization of Items for Sale on Amazon

Topics

For a vast majority of customers, their first task when coming to an e-commerce web site is finding something to buy. They might want to buy a camera, a book, or a piece of jewelry. If you were to walk into a store with relatively few items for sale, the task of finding what you want might be easy. If, however, the store you walk into has thousands of shelves and hundreds of thousands of items for sale, finding what you want presents its own challenge. In such a store, a good store owner would arrange items in a way that would help customers quickly find the items they want to buy. Such is the case with Amazon.

Amazon has literally hundreds of thousands of items for sale. Product Advertising API operations and response groups give you the tools you need to find the items in the store that you want quickly. Before understanding the search mechanisms that Product Advertising API operations provide and the filtering mechanisms that Product Advertising API response groups provide, it is important to first understand the way in which Amazon groups items for sale.

Item Organization

The mechanisms used to organize items for sale in Amazon are:

- Browse nodes
- Search indices
- Variations
- Collections
- Lists
- Accessories

Note

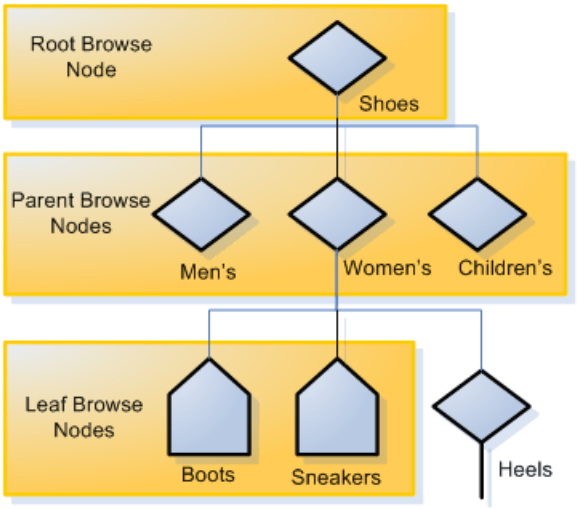
You will sometimes see in responses another organizing tool called `ProductGroup`. This is an older concept in Amazon's database design and it has been superseded by browse nodes and search indices. None of the Product Advertising API operations use `ProductGroup` as an input parameter

Browse Nodes

Topics

Amazon uses a hierarchy of nodes to organize its items for sale. Each node represents a collection of items for sale, such as Harry Potter books, not the items themselves. Product Advertising API calls the nodes, browse nodes because the customer can browse through the nodes to find the collection of items that interests them. For example, the customer might be interested in the browse nodes Literature & Fiction, Medicine, Mystery & Thrillers:, Nonfiction:, or Outdoors & Nature.

Browse nodes are related in a hierarchical structure; each browse node can be a leaf node or a parent node. A leaf node has no children nodes, a parent node does, as shown in the following figure.



Note

The figure is representational in nature and should not be construed to be the real browse node hierarchy used by Amazon.

As you can see in this example, the different levels of the hierarchical tree of nodes provides an organizational principle that is used to catalog and find items. The nodes progress from general to specific. For example, a top level browse node might be "Shoes." It's child browse nodes might be "Men's Shoes," "Women's Shoes," and "Children's Shoes." Child browse nodes are subsets of the parent's product

category. Navigating down the tree refines the search for items from the general to the specific. Going up the tree generalizes the search from the child browse node toward the root node.

Browse Node Properties

Browse nodes properties include:

- **Name—**Describes the items associated with the browse node, such as "Mystery & Thrillers"
- **ID—**A positive integer, for example, Literature & Fiction: (17), Medicine: (13996), Mystery & Thrillers: (18), Nonfiction: (53), Outdoors & Nature (290060)
- **Child nodes—**Nodes that are subsets of the current node
- **Parent nodes—**Nodes that are supersets of the current node

For example, the following XML response shows a browse node whose ID is 163357 and name is "Comedy."

```
<BrowseNode>
  <BrowseNodeId>163357</BrowseNodeId>
  <Name>Comedy</Name>
  <Children>
    <BrowseNode>
      <BrowseNodeId>599826</BrowseNodeId>
      <Name>Boxed Sets</Name>
    </BrowseNode>
    <BrowseNode>
      <BrowseNodeId>538712</BrowseNodeId>
      <Name>African American Comedy</Name>
    </BrowseNode>
  </Children>
  <Item>
    <Ancestors>
      <BrowseNode>
        <BrowseNodeId>549726</BrowseNodeId>
        <Name>Performing Arts</Name>
      </BrowseNode>
    </Ancestors>
  </Item>
```

In this example, the Comedy browse node has two child browse nodes, "Boxed Sets" and "African American Comedy," and one parent node, "Performing Arts."

Root Category

Search results can return with items listed under multiple browse nodes. Some nodes, however, are more relevant that others. The *`IsCategoryRoot`* response tag identifies which browse node is the most relevant for an item in a specific marketplace. This functionality helps vendors classify items for sale.

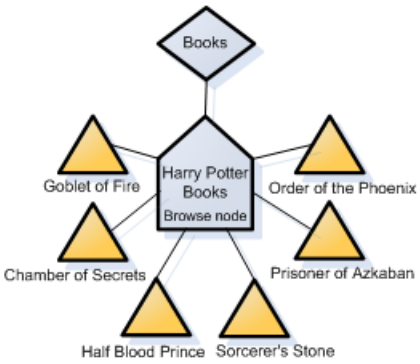
The element applies to the marketplace specified in the request. So, it is possible for the *`IsCategoryRoot`* value to be different across marketplaces.

Note

The *`IsCategoryRoot`* value is not useful in the Books search index.

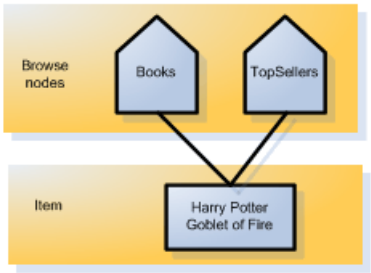
Browse Nodes and Items

Browse nodes are categories into which items for sale are organized. A single node might have many items associated with it, as shown in the following figure.



In this example, six items (Harry Potter book titles) are associated with the browse node: Harry Potter Books. This node is a leaf node and a child of the Books parent node. As the figure shows, many items can be associated with a single browse node. What all of the items have in common is described by the name of the browse node.

On the other hand, a single item can belong to more than one browse node, for example, a book might belong to the Books and TopSellers browse nodes.



At a later time, when the book comes off of the top sellers list, the book will be removed from its association with the TopSellers browse node. In this way, you can see that the association between items and browse nodes is dynamic.

Browse nodes are created and deleted as items demand. When, for example, a new toy or group of books starts selling briskly, a node would be created for it. For example, when pet rocks were popular, a node would have been created for pet rock items. When the sales of pet rocks declined significantly, the node would have been deleted. As you can see, some nodes are volatile by nature. For example, the items associated with the browse node, "Top Sellers," change frequently according to sales figures. Other browse nodes, such as Pet Rocks, exist only for a brief time.

Some browse nodes, however, are much longer lived. Top level nodes, for example, "Books" and "Apparel," have remained unchanged for years. So are the browse nodes associated with cities, as shown in the following table.

City	Browse Node
Boston	917982
Chicago	917984

City	Browse Node
New York	917976
San Francisco	917980
Seattle	917978
Washington, D.C.	917986

Browse Node IDs

Browse node IDs are positive integers that uniquely identify product collections, for example, Literature & Fiction: (17), Medicine: (13996), Mystery & Thrillers: (18), Nonfiction: (53), Outdoors & Nature: (290060). Amazon uses over 120,000 browse node IDs in the US locale alone.

While top level browse node values tend to remain the same, the values of others change often. Browse nodes are created and eliminated without notification. For that reason, it is advisable not to hard code browse node IDs into applications.

Browse node IDs are unique in one locale only, that is, the same browse node ID might be used in two locales but the names and purposes of those browse nodes might be unrelated.

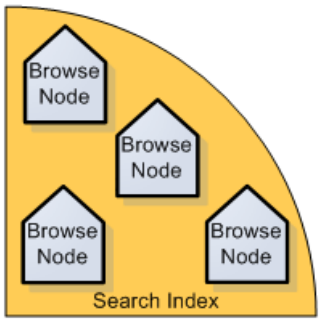
Browse node IDs are assigned internally by Amazon. There are no general rules for number assignments. You cannot assume, for example, that all IDs in the 2000s are related in any way.

If you have a browse node ID, you can find the name of the browse node by putting the ID into one of the following URLs, depending on the locale:

- http://www.amazon.com/exec/obidos/tg/browse/-/[Browse Node ID]
- http://www.amazon.com.ca/exec/obidos/tg/browse/-/[Browse Node ID]
- http://www.amazon.com.uk/exec/obidos/tg/browse/-/[Browse Node ID]
- http://www.amazon.com.fr/exec/obidos/tg/browse/-/[Browse Node ID]
- http://www.amazon.com.de/exec/obidos/tg/browse/-/[Browse Node ID]
- http://www.amazon.com.co.jp/exec/obidos/tg/browse/-/[Browse Node ID]

Browse Nodes and Search Indices

A search index is a more general classification than a browse node. For example, one search index is Books; a browse node within that search index might be Harry Potter Books.



As you can imagine, there can be many browse nodes within each search index. Some Product Advertising API search operations require a search index to limit the scope of the search. Specifying a browse node

in addition to a search index returns more targeted search results. For example, a search operation, ItemSearch, that looked in the Books search index for book titles and descriptions that contained the word "dragon," would return thousands of titles. The same ItemSearch request with the Harry Potter browse node specified would return only those Harry Potter books that had dragons in them.

Finding Browse Nodes

Product Advertising API offers several means of finding browse node IDs:

- `BrowseNodes` response group
- `BrowseNodeInfo` response group

The `BrowseNodes` response group returns the browse node that an item belongs to as well as the ancestry of that browse node. The following response shows the ancestral browse nodes of High School. As you read down the response, the browse nodes ascend the browse node hierarchy. The last browse node, Books, in the response is the root browse node.

```
<Item>
  <ASIN>0976925524</ASIN>
  <BrowseNodes>
    <BrowseNode>
      <BrowseNodeId>69825</BrowseNodeId>
      <Name>High School</Name>
      <Ancestors>
        <BrowseNode>
          <BrowseNodeId>10605</BrowseNodeId>
          <Name>Education</Name>
          <Ancestors>
            <BrowseNode>
            <BrowseNodeId>53</BrowseNodeId>
            <Name>Nonfiction</Name>
            <Ancestors>
              <BrowseNode>
              <BrowseNodeId>1000</BrowseNodeId>
              <Name>Subjects</Name>
              <Ancestors>
                <BrowseNode>
                <BrowseNodeId>283155</BrowseNodeId>
                <Name>Books</Name>
```

There is, however, one caveat. When a node has more than one parent node, the BrowseNodes response group only returns one of the parents. There is no logic that determines which of the parent nodes it follows up the ancestral tree. Running the request multiple times, therefore might return a different set of ancestors for a node.

You can use the `BrowseNodes` response group with `ItemLookup`, `ItemSearch`, and `SimilarityLookup`.

The `BrowseNodeInfo` response group returns browse node names, IDs, children and parent browse nodes, as shown in the following response snippet.

```
<BrowseNodeId>11232</BrowseNodeId>
<Name> Social Sciences</Name>
<Ancestors>
  <BrowseNode>
    <BrowseNodeId>53</BrowseNodeId>
```

```
    <Name>Nonfiction</Name>
    <Ancestors>
      <BrowseNode>
        <BrowseNodeId>1000</BrowseNodeId>
        <Name>Subjects</Name>
        <Ancestors>
          <BrowseNode>
            <BrowseNodeId>283155</BrowseNodeId>
            <Name>Books</Name>
          </BrowseNode>
        </Ancestors>
      </BrowseNode>
    </Ancestors>
  </BrowseNode>
</Ancestors>
<Children>
  <BrowseNode>
    <BrowseNodeId>11233</BrowseNodeId>
    <Name>Anthropology</Name>
  </BrowseNode>
  <BrowseNode>
    <BrowseNodeId>11242</BrowseNodeId>
    <Name>Archaeology</Name>
  </BrowseNode>
  <BrowseNode>
    <BrowseNodeId>3048861</BrowseNodeId>
    <Name>Children's Studies</Name>
  </BrowseNode>
</Children>
```

This response shows that the Social Sciences browse node has three child browse nodes: Anthropology, Archaeology, and Children's Studies, and an ancestry that starts with the parent node, Nonfiction and ends with the root browse node, Books.

You can use this response group only with `BrowseNodeLookup`. Typically use `BrowseNodeLookup` to navigate the browse node tree. With every response, you use the browse node IDs returned to refine your search until you reach the desired browse node.You might, for example, navigate down the tree to refine a search or retrieve the root browse node to return, for example, the top sellers in the product category. To do that, you would use the ID of the root browse node found in the BrowseNodeInfo response as the value for the *BrowseNode* parameter in an **`ItemSearch`** request. That request would include the TopSellers response group so that the top sellers of the product category are returned.

Note: If a browse node has multiple ancestors, only one of them is returned in the response.

In contrast, `BrowseNodeLookup` only returns child browse nodes that are the direct descendant of the browse node in the request. You could use any of the browse node IDs to either find additional, related browse nodes by using another `BrowseNodeLookup` request, or to focus an `ItemSearch` request.

Use `ItemSearch` and `ItemLookup` and specify the BrowseNodes response group.

Find a browse node ID that is similar to the one you want and use `BrowseNodeLookup` to investigate its child and ancestor browse nodes.

Related Topics

- BrowseNodes Response Group (p. 230)

Search Indices

Topics

One of the reasons that customers enjoy shopping on Amazon is that Amazon has hundreds of thousands of items for sale—far more than any one store could contain. On Amazon, you can find just about anything—from a book to a $100,000 piece of jewelry. Ironically, the job of finding the items you want to buy is made more difficult by having so many items for sale. Returning too many items in a response is almost as bad as returning no items at all. Amazon addresses this difficulty by placing every item it sells into a product category, called a search index.

You can think of search indices as dividing into groups everything for sale on Amazon, as shown in the following figure.

By selecting one slice in a request , you avoid returning items in all of the other slices. This functionality makes requests more targeted and reduces search times. For example, when looking for a specific book, it is best to specify the Books search index in the request.

```
SearchIndex=Books
```

Titles and keywords used to find items often occur in multiple search indices. For example, if you were searching using "Harry%20Potter" as your keyword, you would get results in many search indices, including Books, DVD, Video, and Music.By specifying the search index you are interested in, your request becomes much better targeted.

Specifying the wrong search index leads to no results or results that do not meet the customer's search criteria. For example, if your keyword was carburetor and the search index was Kitchen, you might not get any items that match the search criteria. Change the search index to Automotive and you'd get many matches.

Amazon uses the following search indices to divide up all of the items for sale.

• All • Apparel • Automotive • Baby • Beauty • Blended • Books • Classical • DigitalMusic • DVD • Electronics • ForeignBooks • GourmetFood • Grocery • HealthPersonalCare	• Hobbies • HomeGarden • Industrial • Jewelry • KindleStore • Kitchen • Magazines • Miscellaneous • MP3Downloads • Music • MusicalInstruments • MusicTracks • OfficeProducts • OutdoorLiving • PCHardware	• PetSupplies • Photo • Software • SoftwareVideoGames • SportingGoods • Tools • Toys • VHS • Video • VideoGames • Watches • Wireless • WirelessAccessories

Search Indices and Locales

Some search indices work in all locales; some search indices do not. Supported search indices in a locale often changes over time. For example, a search index, such as Automotive, that works in the US locale can be added to the JP locale.

Support for search indices varies by locale often because of the data that Amazon does or does not have. For example, the search index, Grocery, is currently supported in the US locale only because grocers in other countries have not yet teamed up with Amazon to provide grocery items for sale. As the data grows over time, more and more search indices will be supported in all locales.

When you specify a search index in a request, make sure it is supported in the locale of interest. For a list of the search indices supported in different locales, see Search Index Support by Locale (p. 443).

Combined Search Indices

Topics

As a convenience, some of the search indices are combinations of other search indices, for example:

- **All—**Searches through all search indices. Only five pages of items can be returned where each page contains up to five items
- **Blended—**Combines the following search indices: DVD, Electronics, Toys, VideoGames, PCHardware, Tools, SportingGoods, Books, Software, Music, GourmetFood, Kitchen, and Apparel search indices
- **Music—**Combines Classical, DigitalMusic, and MusicTracks search indices
- **Video—**Combines DVD and VHS search indices

These search index combinations are helpful when you are not sure which search index to specify. The downside of using one of these combined search indices is that they might return quite a few items in the response.

More commonly, you will specify individual search indices in requests, for example,

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=All&
ResponseGroup=Images&
SearchIndex=Books&
Title=Harry%20Potter
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

All Search Index

You can use the All search index to do an `ItemSearch` search through all search indices. There are, however, a number of restrictions placed on this request. The only parameter that you can use in the request is *`Keywords`*. You cannot, for example, sort results. Results are restricted to the first five pages of results. Each page can have up to five results.

Note

The list of all available search indexes by locale can be found on the search indexes page.

Variations

Topics

Often, an item comes in a variety of sizes and colors. A shirt, for example, might come in four different sizes and six different colors.

Each color and size combination is called a variation. Each variation, such as a medium, blue shirt, is an item that a customer can buy. For that reason, each variation has its own ASIN. For example, if a shirt came in four sizes and six colors, there would be twenty-four variations and twenty-four corresponding ASINs.

Variation Parents

The abstraction of the variations is called the variation parent. The title element of the variation parent names the variations, for example, "Long Sleeve Classic Pocket Tee." Because the parent ASIN is an abstraction, it cannot be purchased, that is, it is not associated with an offer. The following request uses the Offers response group in an `ItemLookup` of a parent variation.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B00006XYAB&
IdType=ASIN&
Condition=All
ResponseGroup=ItemAttributes,Offers
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

As you can see in the following response snippet, no offers are returned.

```
<Item>
  <ASIN>B00006XYAB</ASIN>
  <ItemAttributes>
    <Binding>Apparel</Binding>
    <Brand>Eddie Bauer</Brand>
    <Department>mens</Department>
    <FabricType>leather</FabricType>
    <Feature>Cotton.</Feature>
    <Feature>Single chest pocket.</Feature>
    <Feature>Side seamed to prevent twisting.</Feature>
    <Feature>Coverstitched seams add strength.</Feature>
    <ProductGroup>Apparel</ProductGroup>
    <Title>LongSleeve Classic Pocket Tee</Title>
  </ItemAttributes>
  <OfferSummary>
    <TotalNew>0</TotalNew>
    <TotalUsed>0</TotalUsed>
    <TotalCollectible>0</TotalCollectible>
    <TotalRefurbished>0</TotalRefurbished>
  </OfferSummary>
  <Offers>
    <TotalOffers>0</TotalOffers>
    <TotalOfferPages>0</TotalOfferPages>
  </Offers>
</Item>
```

Returning Variations

By default, `ItemSearch` only returns parent variations.

To return the associated variations and their images

- Include in the request the Variations and VariationImages response groups.

For example, when you add Variations to the preceding request, the response includes, in addition to what is shown, variations and their offers, as shown in the following response snippet.

```
<Offer>
  ...
  <OfferListing>

<OfferListingId>[illegible]</OfferListingId>
```

```
...
</Offer>
<Offer>
  ...
    <OfferListing>
<OfferListingId>[illegible]</OfferListingId>
```

Each of these offers is associated with an Item element and each of these items is a variation. The following snippet, which comes from the same response, shows how each variation is differentiated by size.

```
<Item>
  <ItemAttributes>
    <Brand>Eddie Bauer</Brand>
    ...
  </ItemAttributes>
  <VariationAttributes>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>Small Regular<Value>
    </VariationAttribute>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>Medium Regular</Value>
    </VariationAttribute>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>Large Regular<Value>
    </VariationAttribute>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>X-Large Regular<Value>
    </VariationAttribute>
  </VariationAttributes>
  ...
</Item>
```

Returning Variation Dimension Information Only

The Variations and VariationSummary response groups return a great deal of information about each item in the response.

To see the values of the variation dimensions

- Use the VariationMatrix response group in an `ItemSearch` request.

```
<Item>
  <ASIN>B0008G23PQ</ASIN>
  <Variations>
  <VariationDimensions>
    <VariationDimension>ClothingSize</VariationDimension>
    <VariationDimension>Color</VariationDimension>
  </VariationDimensions>
<Item>
  <ASIN>B0008EOA9U</ASIN>
```

```
  <VariationAttributes>
    <VariationAttribute>
      <Name>Color</Name>
      <Value>Grey<Value>
    </VariationAttribute>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>29W x 30L</Value>
    </VariationAttribute>
  </VariationAttributes>
</Item>
<Item>
  <ASIN>B0008EO9J6</ASIN>
  <VariationAttributes>
    <VariationAttribute>
      <Name>Color</Name>
      <Value>Navy<Value>
    </VariationAttribute>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>29W x 30L</Value>
    </VariationAttribute>
  </VariationAttributes>

</Item>
```

This response snippet shows that the VariationMatrix response group returns the names of the dimensions along with their values for each returned item. For more information, see VariationMatrix Response Group (p. 299).

Related Topics

- Variations Response Group (p. 294)
- VariationSummary Response Group (p. 304)
- VariationMatrix Response Group (p. 299)

Variation Images

Each variation has its own set of images and those images are returned by the VariationImages response group, for example:

```
  <SmallImage>
    <URL>  http://images.amazon.com/images/P/B99999999A.01._SCTHUMBZZZ_.jpg
    </URL>
    <Height Units="pixels">60</Height>
    <Width Units="pixels">60</Width>
  </SmallImage>

  <MediumImage>
    <URL> http://images.amazon.com/images/P/B99999999A.01._SCMZZZZZZZ_.jpg
    </URL>
    <Height Units="pixels">140</Height>
    <Width Units="pixels">140</Width>
  </MediumImage>
```

These images are small and medium size images of the same child variation. For more information about image sets and variation images, see Motivating Customers to Buy (p. 110).

Variation Dimensions

Variations can differ from one another in a variety of ways. Size and color are common ways for apparel variations to differ. The ways in which variations differ are called dimensions. Parent variations relay that information with the following response elements:

- VariationDimensions
- VariationDimension

The values encapsulated by these elements in the parent variation response specify the variation dimensions for the child variations, for example:

```
<VariationDimensions>
    <VariationDimension>ClothingSize</VariationDimension>
    <VariationDimension>Color</VariationDimension>
</VariationDimensions>
```

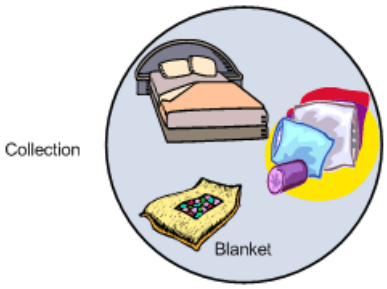
The following response snippet from one of the associated child variations shows that the dimensions specified in the parent variation are used as variation attributes in the child variation.

```
<Item>
  ...
  <VariationAttributes>
    <VariationAttribute>
      <Name>Color</Name>
      <Value>Grey<Value>
    </VariationAttribute>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>29W x 30L</Value>
    </VariationAttribute>
  </VariationAttributes>
</Item>
```

Collections

Topics

A collection is a group of items that are related thematically. For example, all of the linens that go into a bedroom might be associated in a bedding linens collection.



The Collections response group returns all of the items in a collection. You can use `ItemLookup`, `ItemSearch`, and `SimilarityLookup` with the Collections response group to return collections. If, for example, `ItemSearch` returns four items and each of them are in a collection, the Collections response group returns the ASINs and titles of all the items in all four collections (even if some are duplicates), as shown in the following example.

```
<Collections>
  <Collection>
    <CollectionParent>
      <ASIN>B0006PLAOE</ASIN>
      <Title>Fieldcrest® Classic Bedding Collection GarnetIvory</Title>
    </CollectionParent>
    <CollectionItem>
      <ASIN>B00067IV8U</ASIN>
      <Title>Fieldcrest® Classic Solid Sheets Garnet</Title>
</CollectionItem>
    <CollectionItem>
      <ASIN>B000673NE2</ASIN>
      <Title>Fieldcrest® Classic Solid Pillowcases Set of 2 Garnet</Title>
    </CollectionItem>
    <CollectionItem>
      <ASIN>B00065WTJY</ASIN>
      <Title>Fieldcrest® Classic Dobby Pillowcases Set of 2 Garnet</Title>
    </CollectionItem>
  </Collection>
</Collections>
```

This response snippet shows that there are four items in this collection: the collection name, Fieldcrest® Classic Bedding Collection GarnetIvory, and the items in the collection that are for sale: solid sheets, solid pillowcases, and Dobby pillowcases.

Product Advertising API does not offer a means of creating a collection. Collections are created internally by Amazon only.

Related Topics

- Collections Response Group (p. 239)

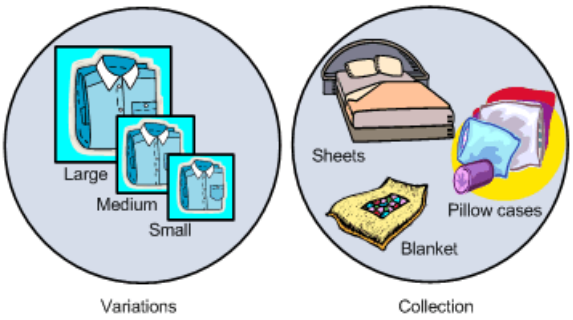
Parent and Child Collection Items

In the preceding response, one of the items is the Collection Parent. All collections have one parent item (CollectionParent) and one or more child items (CollectionItem). The parent item has an ASIN and title but cannot be purchased. The title of the parent item names the collection. For example, in the previous response snippet, the collection is named, "Fieldcrest® Classic Bedding Collection GarnetIvory."

Child items usually can be purchased. The only exception is when a child item is a variation parent ASIN. For more information about variation parents, see Variations.

Variations and Collections are Different

At first, collections might seem to be the same as variations but there are important differences. Items that are variations of one another are essentially the same item but in a slightly different form. For example, the variation of a blue, large shirt would be the same shirt in a different size or color. Items in a collection are fundamentally different from one another, for example, bed sheets and pillow cases could belong to the same collection, as shown in the following figure.



Accessories

Some items for sale have associated accessories. For example, a camera might have, as accessories, a camera case, flash card, and battery, as shown in the following figure.

Each of these accessories has an item ID, such as an ASIN, as shown in the following response snippet. In this example, the main item, B00008OE6I, the camera, returned in the response comes with two accessories, B00003G1RG, a compact flash card, and B00004WCCT, a leather camera case.

```
<Item>
  <ASIN>B00008OE6I</ASIN>
  <Accessories>
    <Accessory>
      <ASIN>B00003G1RG</ASIN>
      <Title>Viking 128 MB CompactFlash Card (CF128M)</Title>
    </Accessory>
    <Accessory>
      <ASIN>B00004WCCT</ASIN>
      <Title>Canon Soft Leather Case for Canon Digital ELPH
Cameras(Black)</Title>
    </Accessory>
```

The following figure shows those items.

As you can see in this example, Amazon groups accessories with the main item for sale. Product Advertising API makes it easy to retrieve all of the accessories associated with a main item by using the Accessories response group in an **`ItemLookup`** or **`ItemSearch`** request. As you can see in the previous response snippet, each accessory listed in the response includes the accessory's title and item identifier, such as an ASIN.

Note

When you have the item ID of the main item, the Accessories response group returns the item's accessories. The reverse, however, is not true, that is, if you have the item ID of an accessory, you cannot use the Accessories response group to return the main item or the other accessories associated with the main item.

Related Items

The *`RelatedItems`* response group returns information about items related to the one specified in an ItemLookup request. The item is specified in an `ItemLookup` request. Digital items include downloadable music (search index: MP3Downloads), downloadable video (search index: UnboxVideo), and digital books (search index: KindleStore). A related item could be, for example, all of the shows in a TV series that are available separately, or, for example, all of the songs on a CD.

The basis upon which the item(s) are related is specified by the *`RelationshipType`* parameter. The *`RelatedItems`* response group requires that you include in the `ItemLookup` request the *`RelationshipType`* parameter. Sample values include Episode, Season, Tracks, and Variation. For a complete list of values, see ItemLookup. (p. 200)

Each `ItemLookup` request can return, at most, ten related items. To return additional items, use the *`RelateditemsPage`* parameter. A value of 2, for example, returns the second set of ten related items.For more information, go to the `ItemLookup`. page.

Requests

Topics

The Product Advertising API web service supports REST requests for remotely calling Product Advertising API operations hosted by Amazon servers. REST requests are simple HTTP requests, using either the GET method with parameters in the URL, or the POST method with parameters in the POST body. The response is an XML document that conforms to a schema.

You might use REST requests because they are more intuitive than their SOAP counterpart or because a SOAP toolkit is not available for your platform. The example requests used throughout this guide are in REST.

Anatomy of a REST Request

Product Advertising API REST requests are URLs, as shown in the following example.

```
http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&Operation=ItemSearch&
AWSAccessKeyId=[Access Key D]&AssociateTag=[ID]&SearchIndex=Apparel&
Keywords=Shirt&Timestamp=[YYYY-MM-DDThh:mm:ssZ]&Signature=[Request Signature]
```

If you substituted real IDs in this request and put the entire example in a browser, you would be sending Product Advertising API a request.

Although the preceding example is in the form you would enter in a browser, it is difficult to read. For this reason, this guide presents the same request as follows.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
Operation=ItemSearch&
AWSAccessKeyId=[Access Key ID]&
AssociateTag=[ID]&
SearchIndex=Apparel&
Keywords=Shirt
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

General Request Format

Part of every Product Advertising API request is the same, the other part of the request changes according to the parameters used in the request, as shown in the following figure.

Request Terms that Remain the Same

The first two lines in the preceding example contain the endpoint, *http://ecs.amazonaws.com/onca/xml*, and the service name, AWSECommerceService.

Amazon hosts many web services in addition to Product Advertising API, including Mechanical Turk and S3. The service name in the request specifies that the request should be sent by the web servers to

Product Advertising API. This line is always the same in every Product Advertising API request, regardless of locale.

```
Service=AWSECommerceService&
```

The endpoint value varies by locale, but there are only two endpoints per locale. One endpoint in a locale is the secure version of the other endpoint. The following table lists the endpoints to use in Product Advertising API requests.

Locale	Endpoint
CA	http://ecs.amazonaws.ca/onca/xml https://ecs.amazonaws.ca/onca/xml
DE	http://ecs.amazonaws.de/onca/xml https://ecs.amazonaws.de/onca/xml
FR	http://ecs.amazonaws.fr/onca/xml https://ecs.amazonaws.fr/onca/xml
JP	http://ecs.amazonaws.jp/onca/xml https://ecs.amazonaws.jp/onca/xml
UK	http://ecs.amazonaws.co.uk/onca/xml https://ecs.amazonaws.co.uk/onca/xml
US	http://ecs.amazonaws.com/onca/xml https://ecs.amazonaws.com/onca/xml

The endpoint remains the same for all Product Advertising API requests, as shown in the following example.

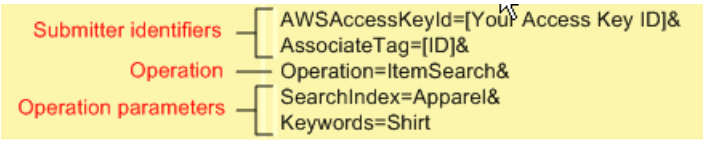
```
http://ecs.amazonaws.com/onca/xml
```

The third and fourth lines identify the request submitter. The AWSAccessKey Id is required; it helps identify the request submitter. You receive an AWS Access Key ID when you sign up with Product Advertising API.

The other identifier, *AssociateTag*, is required. It is an ID for an Associate. If you are an Associate, you must include your Associate tag in each request to be eligible to receive a referral fee for a customer's purchase.

Request Terms that Change

The remaining terms in the request vary significantly according to the operation chosen. The terms, however, follow a pattern, as shown in the next figure.



The *Operation* parameter is required. Its value is one of the Product Advertising API operations. These operations are described throughout this guide.

The last lines, operation parameters, are representative of parameters that the operation requires, and optional parameters that the operation can use. Requests can contain zero or more operation (up to ten) parameters. These parameters are described in the discussion of each operation in the *Product Advertising API Reference Guide*.

Response Groups

A special parameter that is optional for all Product Advertising API operations is *ResponseGroup*. Response groups control the kind of information returned by the request. For example, the Large response group returns a great deal of information about the items included in a response, whereas the Medium and Small response groups return less.

Besides these generic response groups, there are very specific ones. For example, if you want to return images of the items included in a response, you would include the Image response group in the request. If you wanted pricing information, you would include the Offer response group in the request. To get browsenode information, you'd include the BrowseNode response group. The specificity of the response groups enables you to return only the information you want.

Each Product Advertising API operation can only work with a subset of all Product Advertising API response groups. The valid response groups that each Product Advertising API operation can use is listed in the *Product Advertising API Reference Guide*.

All Product Advertising API operations use some response group by default. So, specifying additional response groups is optional. Every Product Advertising API operation uses the Request response group, which echoes operation name and the input parameters sent in the request. The other response groups used by default vary by the operation. For example, `CartCreate`, `CartAdd`, and `CartModify` use, by default, the Cart response group, which provides detailed information about the items in a cart. The *API Reference Guide* lists the default response groups used by each operation.

Now, when you read requests in this guide, your eye should jump to the *`Operation`* parameter and all of the required and optional parameters associated with the specified Product Advertising API operation.

Request Limitations

Some Product Advertising API operations have many parameters and, as you will learn in a future section, you can combine multiple single requests in one, longer batch request. The upper limit is bounded by the maximum number of characters that can be in a request. The maximum number differs by browser. For example, the limit for a URL in Microsoft's Internet Explorer is a little more than 2000 characters. It would be unusual to have a request that approached this upper boundary.

REST Syntax

Topics

One of the values of using REST is that its syntax is simple, which makes REST requests easy to read. This section summarizes all of the REST syntax rules that you must keep in mind when creating a REST request.

Spaces in Requests

Because a REST request is a URL, there can be no spaces between the parts of a request. A browser will stop reading when it runs across the first space. For example, if the last parameter read, `Keywords=Blue Shirts`, the request would end on "Blue." "Shirts" would never be read. If you have key words, such as names, that do have spaces in them, you must URL-encode the space using %20. For the preceding example to work, you would include a URL-encoded space, as follows.

```
Keywords=Blue%20Shirts
```

The same problem occurs if you put spaces between the parameters in a request, as shown in the following example.

```
SearchIndex=Apparel& Keywords=Shirt
```

In this example, the request would end with "Apparel&." Often, this kind of mistake returns an error because parameters required by the operation are not read. So, make sure to remove all spaces within a request.

Separator Characters

The question mark (?) and ampersand (&) separate the terms in a REST request. The first term in the request must always be the endpoint, which, in the preceding example, is,*http://ecs.amazonaws.com/onca/xml*. A question mark always follows the endpoint. The question mark tells the Product Advertising API web servers to start parsing the request for parameters.

Ampersands separate all of the other parameter name-value pairs in the request. The order of the parameter name-value pairs is inconsequential, as long as they all occur after the question mark.

Setting Parameter Values

Topics

Request parameter values are set using the format.

```
ParameterName=value
```

The following example is a parameter/value pair.

```
Operation=ItemSearch
```

Parameter values must be URL-encoded. There are some characters, such as an asterisk or space, that cannot go into a URL. There are equivalents of these characters that you use in requests instead. For example, the URL encoded equivalent of a space is %20. So, instead of writing `Name=John Smith` you would write `Name=John%20Smith`.

Parameter Names and Values are Case Sensitive

Parameter names and values are case sensitive. For example, the following declaration is fine.

```
SearchIndex=Apparel
```

The following examples return errors because the capitalization is incorrect.

```
Searchindex=Apparel
SearchIndex=apparel
```

As you can see in these examples, parameter names and values start with capitals. If the name or value is a compound word, the beginning of each new word is capitalized, for example, in the parameter name, SearchIndex, the "I" is capitalized.

Compound Parameters

The vast majority of parameters can be specified in a REST request using just the name of the parameter and an appropriate value, with the value URL-encoded to make the request a valid URL, as shown in the following example.

```
Author=Steve%20Davenport
```

Some parameters, however, can be repeated in a request. In that case, the parameter names are differentiated by adding a period (.) after the parameter name and then a sequence number, as shown in the following example.

```
Item.1=1234&
Item.2=2345
```

Other parameters can be repeated but, in addition, have associated parameters. These parameters extend the preceding example by adding another period and the associated parameter name, as shown in the following example.

```
Item.1.ASIN=3456789123&
Item.1.Quantity=2
```

In this example, the item being added to a shopping cart has an identifier, the ASIN, and a quantity value. The equivalent expression in an XML document is, as shown in the following example.

```
<Item>
  <ASIN>3456789123</ASIN>
  <Quantity>2</Quantity>
</Item>
```

The sequence number associates the identifier and its quantity value. The following example shows two compound parameters.

```
Item.1.ASIN=3456789123&
Item.1.Quantity=2&
Item.2.ASIN=123456&
Item.2.Quantity=1
```

The sequence numbers associate the ASIN and quantity values. This means that the compound parameters can be written in a different order without causing a problem.

```
Item.1.ASIN=3456789123&
Item.2.ASIN=123456&
Item.1.Quantity=2&
Item.2.Quantity=1
```

SOAP Requests

Product Advertising API supports the SOAP message protocol for calling Product Advertising API operations over an HTTP connection. The easiest way to use the SOAP interface with your application is to use a SOAP toolkit appropriate for your platform. SOAP toolkits are available for most popular programming languages and platforms.

The service's Web Services Definition Language (WSDL) file describes the operations and the format and data types of their requests and responses. Your SOAP toolkit interprets the WSDL file to provide your application access to the operations. For most toolkits, your application calls a service operation using routines and classes provided or generated by the toolkit.

For more information, see WSDL Locations (p. 19).

The Structure of a SOAP Request

A SOAP request is an XML data structure generated by a SOAP toolkit that is sent to a web service. The root element of this structure is named after the operation and contains the values for the operation's parameters.

The root element of every request must contain:

- A value for AWSAccessKeyId to authenticate the request
  For more information, see AWS Access Key ID
- An endpoint, which is the destination for the request
  For more information, see Request Terms That Remain the Same (p. 48)
- A request element, which contains the values for the operation's parameters

The API Reference describes all of the Product Advertising API operations and their parameters. The Product Advertising API WSDL shows how the operation parameters appear in the XML request generated by your toolkit.

SOAP Endpoints

SOAP requests use endpoints in their requests. The endpoint value varies by locale but there are only two endpoints per locale. One endpoint in a locale is the secure version of the other endpoint. The following table lists the endpoints to use in Product Advertising API SOAP requests.

Locale	Endpoint
CA	http://ecs.amazonaws.ca/onca/soap https://aws.amazonaws.ca/onca/soap
DE	http://ecs.amazonaws.de/onca/soap https://aws.amazonaws.de/onca/soap

Locale	Endpoint
FR	http://ecs.amazonaws.fr/onca/soap https://aws.amazonaws.fr/onca/soap
JP	http://ecs.amazonaws.jp/onca/soap https://aws.amazonaws.jp/onca/soap
UK	http://ecs.amazonaws.co.uk/onca/soap https://aws.amazonaws.co.uk/onca/soap
US	http://ecs.amazonaws.com/onca/soap https://aws.amazonaws.com/onca/soap

The XML Message for an ItemSearch SOAP Request

The following example is the XML for a SOAP message that calls the `ItemSearch` operation. While you will probably not be building the SOAP message for a service request manually, it is useful to see what your SOAP toolkit produces when provided with the appropriate values. Many SOAP toolkits require that you build a request data structure similar to the XML to make a request.

The *`ItemSearch`* element contains the parameters common to all requests. The *`Request`* element contains the `ItemSearch` parameters, SearchIndex and Keywords.

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope
     xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ItemSearch
xmlns="http://ecs.amazonaws.com/AWSECSCommerce/onca/soap">
      <AWSAccessKeyId>0PAP1H1P8JJVZEXAMPLE</AWSAccessKeyId>
      <Request>
        <SearchIndex>Books</SearchIndex>
        <Keywords>Harry%20Potter</Keywords>
      </Request>
    </ItemSearch>
  </soapenv:Body>
</soapenv:Envelope>
```

Request Authentication

Topics

The topics in this section describe how the Product Advertising API authenticates your requests. In this section you can learn about the basics of authentication, how your AWS account and identifiers are used to support authentication, and how to create an HMAC-SHA1 signature. This section also covers the request authentication requirements for Query and SOAP requests.

Important

You have until August 15, 2009 to authenticate requests sent to the Product Advertising API. After August 15, 2009, messages that aren't authenticated will be denied.

What Is Authentication?

Authentication is a process for identifying and verifying who is sending a request. The following table shows a simplified version of an authentication process.

General Process of Authentication

1	The sender obtains the necessary credential.
2	The sender sends a request with the credential to the recipient.
3	The recipient uses the credential to verify the sender truly sent the request.
4	If yes, the recipient processes the request. If no, the recipient rejects the request and responds accordingly.

During authentication, Product Advertising API verifies the identity of the sender.

For further discussion of authentication, go to the TechEncylopedia.com entry for authentication. For definitions of common industry terms related to authentication, go to the RSA Laboratories Glossary.

The subsequent sections describe how the Product Advertising API implements authentication to protect your data.

Your Product Advertising API Account

To access the Product Advertising API, you must first create an account at http://aws.amazon.com. An AWS account is simply an Amazon.com account that is enabled to use AWS products; you can use an existing Amazon.com account login and password when creating the AWS account.

Alternately, you could create a new AWS-enabled Amazon.com account by using a new Amazon.com login and password. The e-mail address you provide as the account login must be valid. You'll be asked to provide a credit card or other payment method to cover the charges for any AWS products you use.

From your AWS account you can view your AWS account activity, view usage reports, and manage your AWS account access identifiers.

Your AWS Identifiers

When you create an AWS account, AWS assigns you a pair of related identifiers:

- Access Key ID (a 20-character, alphanumeric sequence)
  For example: 022QF06E7MXBSH9DHM02
- Secret Access Key (a 40-character sequence)
  For example: kWcrlUX5JEDGM/LtmEENI/aVmYvHNif5zB+d9+ct

These are your *AWS access key identifiers*.

Caution

Your Secret Access Key is a secret and only you and AWS should know it. It is important to keep it confidential to protect your account. Never include it in your requests to AWS, and never e-mail it to anyone. Do not share it outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your Secret Access Key.

Important

If you developed your application prior to 2006, you may be using a subscription ID instead of an AWS Access Key ID. You can only use your AWS Access Key ID in authenticated requests. Please change your application to use the AWS Access Key ID. For information about viewing your AWS Access Key ID, see Viewing Your AWS Identifiers (p. 56).

The Access Key ID is associated with your AWS account. You include it in AWS service requests to identify yourself as the sender of the request.

The Access Key ID is not a secret, and anyone could use your Access Key ID in requests to AWS. To provide proof that you truly are the sender of the request, you must also include a digital signature. For all requests except those using SOAP with WS-Security, you calculate the signature using your Secret Access Key. AWS uses the Access Key ID in the request to look up your Secret Access Key and then calculates a digital signature with the key. If the signature AWS calculates matches the signature you sent, the request is considered authentic. Otherwise, the request fails authentication and is not processed.

Related Topics

- HMAC-SHA Signatures for REST Requests (p. 57)
- Authenticating REST Requests (p. 59)

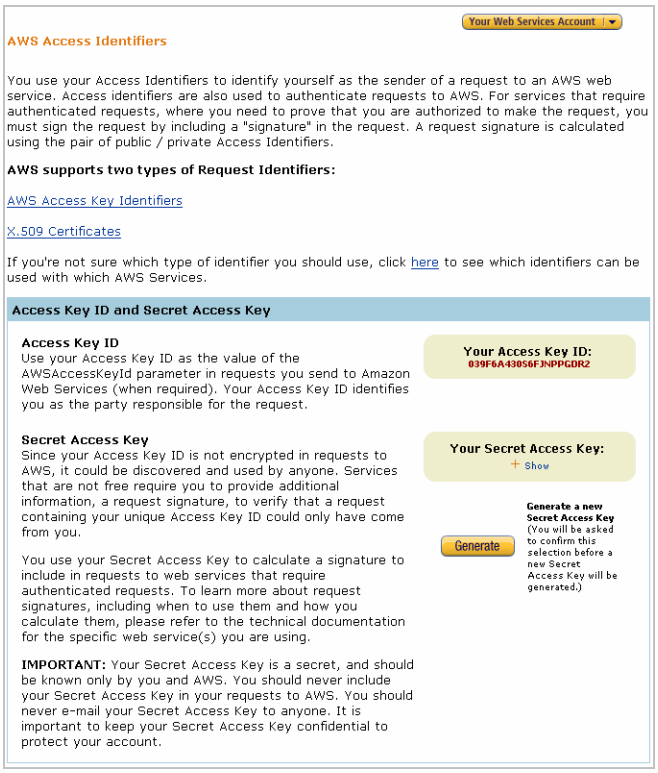
Viewing Your AWS Identifiers

Your Access Key ID and Secret Access Key are displayed to you when you create your AWS account. They are not e-mailed to you. If you need to see them again, you can view them at any time from your AWS account.

To view your AWS access identifiers

1. Go to the Amazon Web Services web site at http://aws.amazon.com.
2. Point to **Your Account** and click **Access Identifiers**.
3. Log in to your AWS account.

Your Access Key ID and Secret Access Key are displayed on the resulting **Access Identifiers** page. Following is an example of this page.



Related Topics

- Your Product Advertising API Account (p. 55)
- Your AWS Identifiers (p. 55)

HMAC-SHA Signatures for REST Requests

Topics

- Authentication Parameters (p. 57)
- Basic Authentication Process (p. 58)

The topics in this section describe how Product Advertising API uses HMAC-SHA signatures to authenticate REST requests.

Authentication Parameters

The following parameters are used by Product Advertising API for REST authentication:

- **Signature—**Required. There is no default value. A signature is created by using the request type, domain, the URI, and a sorted string of every parameter in the request (except the Signature parameter itself) with the following format `<parameter>=<value>&`. Once properly formatted, you create a base64-encoded HMAC_SHA256 signature using your AWS secret key. For a detailed example of this process, see Example REST Requests (p. 63).
- **Timestamp—**Required. There is no default value. The time stamp you use in the request must be a `dateTime` object, with the complete date plus hours, minutes, and seconds (for more information, go to http://www.w3.org/TR/NOTE-datetime).This is a fixed -length subset of the format defined by ISO 8601, represented in Universal Time (GMT): `YYYY-MM-DDThh:mm:ssZ` (where T and Z are literals).

Important

If you are using .NET you must not send overly specific time stamps, due to different interpretations of how extra time precision should be dropped. To avoid overly specific time stamps, manually construct `dateTime` objects with no more than millisecond precision.

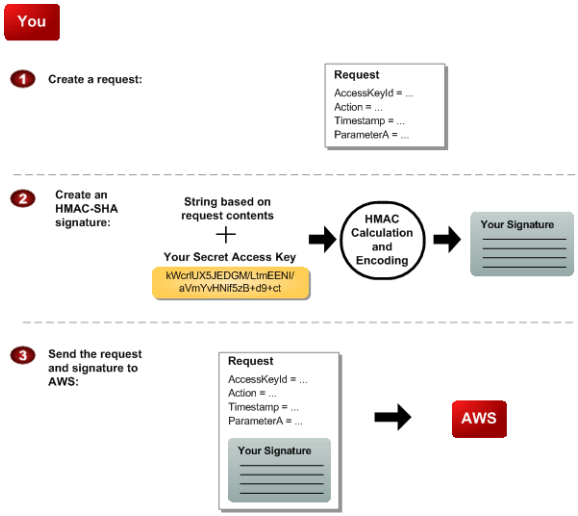
Related Topics

- Your AWS Identifiers (p. 55)

Basic Authentication Process

The following figures and text describe the series of tasks required to authenticate requests to AWS using an HMAC-SHA request signature. It is assumed you have already created an AWS account and received an Access Key ID and Secret Access Key. For more information about those, see Your Product Advertising API Account (p. 55) and Your AWS Identifiers (p. 55).

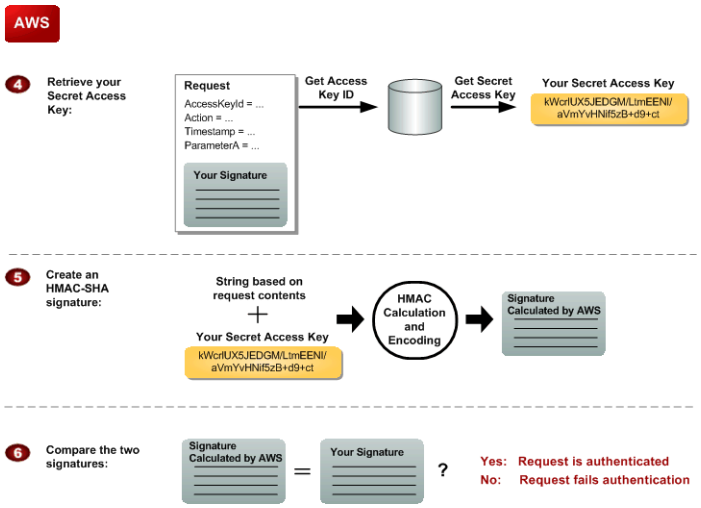
You perform the first three tasks.



Process for Authentication: Tasks You Perform

1	You construct a request to AWS.
2	You calculate a keyed-hash message authentication code (HMAC-SHA) signature using your Secret Access Key (for information about HMAC, go to http://www.faqs.org/rfcs/rfc2104.html)
3	You include the signature and your Access Key ID in the request, and then send the request to AWS.

AWS performs the next three tasks.



Process for Authentication: Tasks AWS Performs

4	Product Advertising API uses the Access Key ID to look up your Secret Access Key.
5	Product Advertising API generates a signature from the request data and the Secret Access Key using the same algorithm you used to calculate the signature you sent in the request.
6	If the signature generated by AWS matches the one you sent in the request, the request is considered authentic. If the comparison fails, the request is discarded, and AWS returns an error response.

Authenticating REST Requests

This section describes how to create a signature. The Product Advertising API only supports Signature Version 2.

To create the signature

1. Create the canonicalized query string that you need later in this procedure:

   a. Sort the UTF-8 query string components by parameter name with natural byte ordering. The parameters can come from the GET URI or from the POST body (when `Content-Type` is `application/x-www-form-urlencoded`).

   b. URL encode the parameter name and values according to the following rules:

      - Do not URL encode any of the unreserved characters that RFC 3986 defines. These unreserved characters are A-Z, a-z, 0-9, hyphen (-), underscore (_), period (.), and tilde (~).
      - Percent encode extended UTF-8 characters in the form %XY%ZA....
      - Percent encode the space character as %20 (and not +, as common encoding schemes do).
      - Percent encode all other characters with %XY, where X and Y are hex characters 0-9 and uppercase A-F.

Perl Note:

The commonly used URI::Escape CPAN module uses RFC 2396. This has five additional reserved characters: asterisk (*), left and right parenthesis ((and)), single quote (‘) and exclamation (!). To follow RFC 3986 use:

```
URI::Escape::uri_escape( $parameter_value, "^A-Za-z0-9\-_.~" )
```

Java Note:

URLEncoder uses + for space, and won’t encode asterisk (*) , and encodes tilda (~) when not necessary. To follow RFC 3986 use:

```
URLEncoder.encode(value, UTF_8_Encoding).replace("+",
"%20").replace("*", "%2A").replace("%7E", "~");
```

C# Note:

Use uppercase hex characters.

Tip

Currently all Product Advertising API service parameter names use unreserved characters, so you don't need to encode them. However, you might want to include code to handle parameter names that use reserved characters, for possible future use.

c. Separate the encoded parameter names from their encoded values with the equals sign (=) (ASCII character 61), even if the parameter value is empty.
d. Separate the name-value pairs with an ampersand (&) (ASCII code 38).

2. Create the string to sign according to the following pseudo-grammar (the `"\n"` represents an ASCII newline).

```
StringToSign = HTTPVerb + "\n" +
    ValueOfHostHeaderInLowercase + "\n" +
    HTTPRequestURI + "\n" +
    CanonicalizedQueryString <from the preceding step>
```

The HTTPRequestURI component is the HTTP absolute path component of the URI up to, but not including, the query string. If the HTTPRequestURI is empty, use a forward slash (/).

Note

HTTPRequestURI is always “/onca/xml” for Product Advertising API. HTTPVerb is either GET or POST.

3. Calculate an RFC 2104-compliant HMAC with the string you just created, your Secret Access Key as the key, and SHA256 as the hash algorithm.
   For more information, go to http://tools.ietf.org/html/rfc2104.
4. Convert the resulting value to base64.
5. Use the resulting value as the value of the *`Signature`* request parameter.

Important

The final signature you send in the request must be URL encoded as specified in RFC 3986 (for more information, go to http://tools.ietf.org/html/rfc3986). If your toolkit URL encodes your final request, then it handles the required URL encoding of the signature. If your toolkit doesn't URL encode the final request, then make sure to URL encode the signature before you include it in the request. Most importantly, make sure the signature is URL encoded *only once*. A common mistake is to URL encode it manually during signature formation, and then again when the toolkit URL encodes the entire request.

Java Sample Code for Calculating Signature Version 2 Signatures

The following Java code sample shows how to calculate a signature (version 2). This sample has code for creating the canonical string, for base64 encoding, and for HMAC encoding for generating the signature.

```
package com.amazon.associates.sample;

import java.io.UnsupportedEncodingException;

import java.net.URLDecoder;
import java.net.URLEncoder;

import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;

import java.text.DateFormat;
import java.text.SimpleDateFormat;

import java.util.Calendar;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.SortedMap;
import java.util.TimeZone;
import java.util.TreeMap;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

import org.apache.commons.codec.binary.Base64;

public class SignedRequestsHelper {
  private static final String UTF8_CHARSET = "UTF-8";
  private static final String HMAC_SHA256_ALGORITHM = "HmacSHA256";
  private static final String REQUEST_URI = "/onca/xml";
  private static final String REQUEST_METHOD = "GET";

  private String endpoint = "ecs.amazonaws.com"; // must be lowercase
  private String awsAccessKeyId = "YOUR AWS ACCESS KEY";
```

```
private String awsSecretKey = "YOUR AWS SECRET KEY";

private SecretKeySpec secretKeySpec = null;
private Mac mac = null;

public SignedRequestsHelper() {
  byte[] secretyKeyBytes = awsSecretKey.getBytes(UTF8_CHARSET);
  secretKeySpec =
    new SecretKeySpec(secretyKeyBytes, HMAC_SHA256_ALGORITHM);
  mac = Mac.getInstance(HMAC_SHA256_ALGORITHM);
  mac.init(secretKeySpec);
}

public String sign(Map<String, String> params) {
  params.put("AWSAccessKeyId", awsAccessKeyId);
  params.put("Timestamp", timestamp());

  SortedMap<String, String> sortedParamMap =
    new TreeMap<String, String>(params);
  String canonicalQS = canonicalize(sortedParamMap);
  String toSign =
    REQUEST_METHOD + "\n"
    + endpoint + "\n"
    + REQUEST_URI + "\n"
    + canonicalQS;

  String hmac = hmac(toSign);
  String sig = percentEncodeRfc3986(hmac);
  String url = "http://" + endpoint + REQUEST_URI + "?" +
  canonicalQS + "&Signature=" + sig;

  return url;
}

private String hmac(String stringToSign) {
  String signature = null;
  byte[] data;
  byte[] rawHmac;
  try {
    data = stringToSign.getBytes(UTF8_CHARSET);
    rawHmac = mac.doFinal(data);
    Base64 encoder = new Base64();
    signature = new String(encoder.encode(rawHmac));
  } catch (UnsupportedEncodingException e) {
    throw new RuntimeException(UTF8_CHARSET + " is unsupported!", e);
  }
  return signature;
}

private String timestamp() {
  String timestamp = null;
  Calendar cal = Calendar.getInstance();
  DateFormat dfm = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'");
  dfm.setTimeZone(TimeZone.getTimeZone("GMT"));
  timestamp = dfm.format(cal.getTime());
  return timestamp;
}
```

```
  private String canonicalize(SortedMap<String, String> sortedParamMap)
{
    if (sortedParamMap.isEmpty()) {
      return "";
    }

    StringBuffer buffer = new StringBuffer();
    Iterator<Map.Entry<String, String>> iter =
      sortedParamMap.entrySet().iterator();

    while (iter.hasNext()) {
      Map.Entry<String, String> kvpair = iter.next();
      buffer.append(percentEncodeRfc3986(kvpair.getKey()));
      buffer.append("=");
      buffer.append(percentEncodeRfc3986(kvpair.getValue()));
      if (iter.hasNext()) {
        buffer.append("&");
      }
    }
    String cannoical = buffer.toString();
    return cannoical;
  }

  private String percentEncodeRfc3986(String s) {
    String out;
    try {
      out = URLEncoder.encode(s, UTF8_CHARSET)
      .replace("+", "%20")
      .replace("*", "%2A")
      .replace("%7E", "~");
    } catch (UnsupportedEncodingException e) {
      out = s;
    }
    return out;
  }

}
```

Example REST Requests

This section shows the steps to sign a request, using sample AWS access identifiers.

If you follow these steps using these identifiers, you will generate the same signature strings in the examples. Although this can help verify your request generation code, you cannot make actual requests with these example requests.

Tip

If you aren't familiar with REST requests, read Anatomy of a REST Request (p. 48) before continuing with this example.

The following is an example of an ItemLookup (p. 200) request:

```
http://webservices.amazon.com/onca/xml?Service=AWSECommerceServic
e&AWSAccessKeyId=00000000000000000000&Operation=ItemLookup&ItemId
```

```
=0679722769&ResponseGroup=ItemAttributes,Offers,Images,Reviews&Ve
rsion=2009-01-06
```

Steps to Sign the Example Request

1. Enter the timestamp. For this example, we'll use the UTC time **2009-01-01T12:00:00Z**

```
http://webservices.amazon.com/onca/xml?Service=AWSECommerceServic
e&AWSAccessKeyId=0000000000000000000&Operation=ItemLookup&ItemId
=0679722769&ResponseGroup=ItemAttributes,Offers,Images,Reviews&Ve
rsion=2009-01-06&Timestamp=2009-01-01T12:00:00Z
```

2. URL encode the request's comma (,) and colon (:) characters, so that they don't get misinterpreted. For more information about converting to RFC 3986 specifications, see documentation and code samples for your programming language.

```
http://webservices.amazon.com/onca/xml?Service=AWSECommerceServic
e&AWSAccessKeyId=0000000000000000000&Operation=ItemLookup&ItemId
=0679722769&ResponseGroup=ItemAttributes%2COffers%2CImages%2CRevi
ews&Version=2009-01-06&Timestamp=2009-01-01T12%3A00%3A00Z
```

Important

Be sure that you do not double-escape any characters.

3. Split the parameter/value pairs and delete the ampersand characters (&) so that the example looks like the following:

```
Service=AWSECommerceService
AWSAccessKeyId=00000000000000000000
Operation=ItemLookup
ItemId=0679722769
ResponseGroup=ItemAttributes%2COffers%2CImages%2CReviews
Version=2009-01-06
Timestamp=2009-01-01T12%3A00%3A00Z
```

4. Sort your parameter/value pairs by byte value (not alphabetically, lowercase parameters will be listed after uppercase ones).

```
AWSAccessKeyId=00000000000000000000
ItemId=0679722769
Operation=ItemLookup
ResponseGroup=ItemAttributes%2COffers%2CImages%2CReviews
Service=AWSECommerceService
Timestamp=2009-01-01T12%3A00%3A00Z
Version=2009-01-06
```

5. Rejoin the sorted parameter/value list with ampersands. The result is the canonical string that we'll sign:

```
AWSAccessKeyId=0000000000000000000&ItemId=0679722769&Operation=I
temLookup&ResponseGroup=ItemAttributes%2COffers%2CImages%2CReview
s&Service=AWSECommerceService&Timestamp=2009-01-01T12%3A00%3A00Z&
Version=2009-01-06
```

6. Prepend the following three lines (with line breaks) before the canonical string:

```
GET
webservices.amazon.com
/onca/xml
```

7. The string to sign:

```
GET
webservices.amazon.com
/onca/xml
AWSAccessKeyId=0000000000000000000&ItemId=0679722769&Operation=I
temLookup&ResponseGroup=ItemAttributes%2COffers%2CImages%2CReview
s&Service=AWSECommerceService&Timestamp=2009-01-01T12%3A00%3A00Z&
Version=2009-01-06
```

8. Calculate an RFC 2104-compliant HMAC with the SHA256 hash algorithm using the string above with our "dummy" Secret Access Key: **1234567890**. For more information about this step, see documentation and code samples for your programming language.

```
Nace+U3Az4OhN7tISqgs1vdLBHBEijWcBeCqL5xN9xg=
```

9. URL encode the plus (+) and equal (=) characters in the signature:

```
Nace%2BU3Az4OhN7tISqgs1vdLBHBEijWcBeCqL5xN9xg%3D
```

10. Add the URL encoded signature to your request and the result is a properly-formatted signed request:

```
http://webservices.amazon.com/onca/xml?AWSAccessKeyId=00000000000
000000000&ItemId=0679722769&Operation=ItemLookup&ResponseGroup=It
emAttributes%2COffers%2CImages%2CReviews&Service=AWSECommerceServ
ice&Timestamp=2009-01-01T12%3A00%3A00Z&Version=2009-01-06&Signatu
re=Nace%2BU3Az4OhN7tISqgs1vdLBHBEijWcBeCqL5xN9xg%3D
```

Examples of Other Signed Requests

The previous section went through a detailed process for signing an ItemLookup request. This section presents examples of unsigned requests, the string to sign generated from them, and the final signed request. These examples can make excellent test cases for your software. All examples use the "dummy" AWS access identifiers used earlier in the Steps to Sign the Example Request (p. 64).

ItemSearch Examples

The following examples show the ItemSearch (p. 207) request without a signature, with the string to sign, and the signed request.

Original Unsigned Request

```
http://ecs.amazonaws.co.uk/onca/xml?Service=AWSECommerceService&A
WSAccessKeyId=00000000000000000000&Operation=ItemSearch&Actor=Joh
nny%20Depp&ResponseGroup=ItemAttributes,Offers,Images,Reviews,Var
iations&Version=2009-01-01&SearchIndex=DVD&Sort=salesrank&Associa
teTag=mytag-20
```

String to Sign

```
GET
ecs.amazonaws.co.uk
/onca/xml
AWSAccessKeyId=00000000000000000000&Actor=Johnny%20Depp&Associate
Tag=mytag-20&Operation=ItemSearch&ResponseGroup=ItemAttributes%2C
Offers%2CImages%2CReviews%2CVariations&SearchIndex=DVD&Service=AW
SECommerceService&Sort=salesrank&Timestamp=2009-01-01T12%3A00%3A0
0Z&Version=2009-01-01
```

Signed Request

```
http://ecs.amazonaws.co.uk/onca/xml?AWSAccessKeyId=00000000000000
000000&Actor=Johnny%20Depp&AssociateTag=mytag-20&Operation=ItemSe
arch&ResponseGroup=ItemAttributes%2COffers%2CImages%2CReviews%2CV
ariations&SearchIndex=DVD&Service=AWSECommerceService&Signature=T
uM6E5L9u%2FuNqOX09ET03BXVmHLVFfJIna5cxXuHxiU%3D&Sort=salesrank&Ti
mestamp=2009-01-01T12%3A00%3A00Z&Version=2009-01-01
```

CartCreate Examples

The following examples show the CartCreate (p. 187) request without a signature, with the string to sign, and the signed request.

Original Unsigned Request

```
http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&AWS
AccessKeyId=00000000000000000000&Operation=CartCreate&Version=200
9-01-01&Item.1.OfferListingId=j8ejq9wxDfSYWf2OCp6XQGDsVrWhl08GSQ9
m5j%2Be8MS449BN1XGUC3DfU5Zw4nt%2FFBt87cspLow1QXzfvZpvzg%3D%3D&Ite
m.1.Quantity=3&AssociateTag=mytag-20
```

String to Sign

```
GET
ecs.amazonaws.com
/onca/xml
AWSAccessKeyId=00000000000000000000&AssociateTag=mytag-20&Item.1.
OfferListingId=j8ejq9wxDfSYWf2OCp6XQGDsVrWhl08GSQ9m5j%2Be8MS449BN
1XGUC3DfU5Zw4nt%2FFBt87cspLow1QXzfvZpvzg%3D%3D&Item.1.Quantity=3&
Operation=CartCreate&Service=AWSECommerceService&Timestamp=2009-0
1-01T12%3A00%3A00Z&Version=2009-01-01
```

Signed Request

```
http://ecs.amazonaws.com/onca/xml?AWSAccessKeyId=0000000000000000
0000&AssociateTag=mytag-20&Item.1.OfferListingId=j8ejq9wxDfSYWf2O
Cp6XQGDsVrWhl08GSQ9m5j%2Be8MS449BN1XGUC3DfU5Zw4nt%2FFBt87cspLow1Q
XzfvZpvzg%3D%3D&Item.1.Quantity=3&Operation=CartCreate&Service=AW
SECommerceService&Signature=cF3UtjbJb1%2BxDh387C%2FEmS1BCtS%2FZ01
taykBCGemvUU%3D&Timestamp=2009-01-01T12%3A00%3A00Z&Version=2009-0
1-01
```

BrowseNodeLookup Examples

The following examples show the BrowseNodeLookup (p. 177) request without a signature, with the string to sign, and the signed request.

Original Unsigned Request

```
http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&AWS
AccessKeyId=00000000000000000000&Operation=BrowseNodeLookup&Versi
on=2009-01-01&BrowseNodeId=465600&AssociateTag=mytag-20&ResponseG
roup=BrowseNodeInfo,TopSellers,NewReleases,MostWishedFor,MostGift
ed
```

String to Sign

```
GET
ecs.amazonaws.com
/onca/xml
AWSAccessKeyId=00000000000000000000&AssociateTag=mytag-20&BrowseN
odeId=465600&Operation=BrowseNodeLookup&ResponseGroup=BrowseNodeI
nfo%2CTopSellers%2CNewReleases%2CMostWishedFor%2CMostGifted&Servi
ce=AWSECommerceService&Timestamp=2009-01-01T12%3A00%3A00Z&Version
=2009-01-01
```

Signed Request

```
http://ecs.amazonaws.com/onca/xml?AWSAccessKeyId=0000000000000000
0000&AssociateTag=mytag-20&BrowseNodeId=465600&Operation=BrowseNo
deLookup&ResponseGroup=BrowseNodeInfo%2CTopSellers%2CNewReleases%
2CMostWishedFor%2CMostGifted&Service=AWSECommerceService&Signatur
e=kEXxAIqhh6eBhLhrVMz2gt3ocMaH%2FOBVPbjvc9TG8ao%3D&Timestamp=2009
-01-01T12%3A00%3A00Z&Version=2009-01-01
```

SimilarityLookup Examples

The following examples show the SimilarityLookup (p. 218) request without a signature, with the string to sign, and the signed request.

Original Unsigned Request

```
http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&AWS
AccessKeyId=00000000000000000000&Operation=SimilarityLookup&ItemI
d=B0011ZK6PC,B000NK8EWI&Version=2009-01-01&AssociateTag=mytag-20&
```

```
ResponseGroup=Offers,ItemAttributes&SimilarityType=Intersection&C
ondition=New&Merchant=Amazon
```

String to Sign

```
GET
ecs.amazonaws.com
/onca/xml
AWSAccessKeyId=00000000000000000000&AssociateTag=mytag-20&Conditi
on=New&ItemId=B0011ZK6PC%2CB000NK8EWI&Merchant=Amazon&Operation=S
imilarityLookup&ResponseGroup=Offers%2CItemAttributes&Service=AWS
ECommerceService&SimilarityType=Intersection&Timestamp=2009-01-01
T12%3A00%3A00Z&Version=2009-01-01
```

Signed Request

```
http://ecs.amazonaws.com/onca/xml?AWSAccessKeyId=0000000000000000
0000&AssociateTag=mytag-20&Condition=New&ItemId=B0011ZK6PC%2CB000
NK8EWI&Merchant=Amazon&Operation=SimilarityLookup&ResponseGroup=O
ffers%2CItemAttributes&Service=AWSECommerceService&Signature=I2pb
qxuS%2FmZK6Apwz0oLBxJn2wDL5n4kFQhgYWgLM7I%3D&SimilarityType=Inter
section&Timestamp=2009-01-01T12%3A00%3A00Z&Version=2009-01-01
```

Authenticating SOAP Requests

Topics

The Product Advertising API accepts SOAP requests sent over an HTTPS connection only. We prefer that you use WS-Security because it is the most secure. However, if your SOAP toolkit does not support WS-Security, you can authenticate SOAP requests with your AWS secret key instead.

Using SOAP with WS-Security

WS-Security, which is officially called Web Services Security: SOAP Message Security, is an open standard published by OASIS that defines mechanisms for signing and encrypting SOAP messages. Product Advertising API supports version 1.0 of the WS-Security specification. For more information and a link to the WS-Security 1.0 specification, go to the OASIS-Open web site for WS-Security.

Tip

The easiest way to comply with the WS-Security requirements is to use a SOAP toolkit that supports WS-Security 1.0 and X.509 certificates.

Caution

For the purposes of request security and integrity, you must send all SOAP requests over HTTPS.

X.509 Certificates

When using SOAP with WS-Security, you must use an X.509 certificate for authentication (as opposed to your AWS Secret Access Key). An X.509 certificate is a security token designed to carry a public key and bind that key to an identity. X.509 certificates are used in a public key infrastructure (PKI), which is a framework for allowing trusted third parties to vouch for a party's identity. PKIs are commonly used in situations that require authentication. For more information about PKIs and X.509 certificates, go to the techencyclopedia.com entries for digital certificate and PKI.

Note

Product Advertising API does not implement a full public key infrastructure. The certificate information is used only to authenticate requests to Product Advertising API. Product Advertising API uses X.509 certificates only as carriers for public keys and does not trust or use in any way any identity binding that might be included in an X.509 certificate.

The WS-Security 1.0 specification requires you to sign the SOAP message with the private key associated with the X.509 certificate and include the X.509 certificate in the SOAP message header. Specifically, you must represent the X.509 certificate as a `BinarySecurityToken` as described in the WS-Security X.509 token profile (also available if you go to the OASIS-Open web site).

You can use your own X.509 certificate or one generated by AWS. Following are the procedures for uploading your own certificate to AWS and obtaining an AWS-generated certificate. To obtain an X.509 certificate generated by an external certificate authority, follow the instructions provided by that CA.

Using Your Own Certificate

If you have an X.509 certificate you want to use, you can upload the certificate to AWS (without the private key value). Uploading the certificate automatically associates it with your AWS account.

AWS accepts any syntactically and cryptographically valid X.509 certificate. Certificates can be self-signed or signed by any key. The certificate must be in Privacy Enhanced Mail (PEM) format and include a base64 encoded Distinguished Encoding Rules (DER) certificate body.

Important

When you upload the certificate, AWS checks the certificate's contents to confirm that the certificate has not expired. AWS doesn't check certificate revocation lists (CRLs) to determine if the certificate has been revoked, nor does AWS validate the certificate with a certificate authority (CA) or any trusted third parties.

To upload your own X.509 certificate

1. Go to the Amazon Web Services web site at http://aws.amazon.com.
2. Point to **Your Web Services Account** to display a list of options.
3. Click **View Access Key Identifiers** and log in to your AWS account.
   The **AWS Access Key Identifiers** page is displayed.
4. Scroll down to the **X.509 Certificate** area of the page and click **Upload.**
5. Follow the instructions on the subsequent pages to upload your certificate.

Using a Certificate Generated by AWS

If you don't already have an X.509 certificate, or if you want a new certificate to use with AWS, you can have AWS generate one and automatically associate it with your AWS account. Certificates generated by AWS are signed by an AWS internal certificate authority.

To have AWS create an X.509 certificate for you

1. Go to the Amazon Web Services web site at http://aws.amazon.com.
2. Point to **Your Web Services Account** to display a list of options.
3. Click **View Access Key Identifiers** and log in to your AWS account.
   The **AWS Access Key Identifiers** page is displayed.
4. Scroll down to the **X.509 Certificate** area of the page and click **Create New.**
   The **Create a New Certificate** page is displayed.
5. Read the information on the page and click **Yes** to create your certificate.
   The **Create Success** page is displayed.
6. Download your private key file and X.509 certificate file.

Important

Don't close the browser window before you download the certificate, otherwise you'll have to create a new certificate.

What Needs to Be Signed

Each SOAP request must be signed with the private key associated with the X.509 certificate. To create the signature, you sign the `Timestamp` element, and if you're using WS-Addressing, we recommend you also sign the `Action` header element. In addition, you can optionally sign the `Body` and the `To` header element.

Example Request to Use When Troubleshooting

The following example shows the initial portion of a SOAP request that uses WS-Security with an X.509 certificate. If you're using a SOAP toolkit that supports WS-Security and X.509 certificates, the toolkit constructs the request automatically for you, so you don't have to create a request like this yourself. The example is included here as a reference to use if you're troubleshooting authentication issues with your SOAP requests. Several requirements are listed following the example; the numbers highlight where in the example the requirements are satisfied.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">

  <wsse:BinarySecurityToken
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
    (1) EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
soap-message-security-1.0#Base64Binary"
    (2) ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
x509-token-profile-1.0#X509v3"
    wsu:Id="CertId-1064304">
      (3) [Your base64 encoded X.509 certificate…]
  </wsse:BinarySecurityToken>
```

```
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>

   <ds:CanonicalizationMethod 4 Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></ds:CanonicalizationMethod>

     <ds:SignatureMethod 5 Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"></ds:SignatureMethod>

     <ds:Reference URI="#id-17984263">
        <ds:Transforms>

          <ds:Transform 6 Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></ds:DigestMethod>
        <ds:DigestValue>0pjZ1+TvgPf6uG7o+Yp3l2YdGZ4=</ds:DigestValue>
     </ds:Reference>

    <ds:Reference URI="#id-15778003">
      <ds:Transforms>

        <ds:Transform 6 Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></ds:Transform>
      </ds:Transforms>

      <ds:DigestMethod 7
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></ds:DigestMethod>
      <ds:DigestValue>HhRbxBBmc2OO348f8nLNZyo4AOM=</ds:DigestValue>
    </ds:Reference>

    </ds:SignedInfo>


<ds:SignatureValue>bmVx24Qom4kd9QQtclxWIlgLk4QsQBPaKESi79x479xgbO9PEStXMiHZuBAi9luuKdNTcfQ8UE/d

      jjHKZKEQRCOlLVy0Dn5ZL1RlMHsv+OzJzzvIJFTq3LQKNrzJzsNe</ds:SignatureValue>


    <ds:KeyInfo Id="KeyId-17007273">

      8 <wsse:SecurityTokenReference
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
        wsu:Id="STRId-22438818">
        <wsse:Reference URI="#CertId-1064304"
           ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
x509-token-profile-1.0#X509v3">
        </wsse:Reference>
      </wsse:SecurityTokenReference>
    </ds:KeyInfo>

  </ds:Signature>


  <wsu:Timestamp
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
```

```
    9 wsu:Id="id-17984263">
    <wsu:Created>2011-09-24T10:57:35Z</wsu:Created>
    <wsu:Expires>2011-09-24T11:02:35Z</wsu:Expires>
  </wsu:Timestamp>

  </wsse:Security>
</SOAP-ENV:Header>
```

Requirements for BinarySecurityToken and Signatures

1. The `EncodingType` attribute for the `BinarySecurityToken` element must be:

```
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-
security-1.0#Base64Binary
```

2. The `ValueType` attribute for the `BinarySecurityToken` element must be:

```
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509v3
```

or:

```
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509PKIPathv1
```

3. The `BinarySecurityToken` element must contain the base64 encoding of the leaf X.509 certificate if the `ValueType` is `#X509v3`, or it must contain the base64 encoding of the full X.509 certificate chain if the `ValueType` is `#X509PKIPathv1`.
4. The `Algorithm` attribute of the `CanonicalizationMethod` element must be:

```
http://www.w3.org/2001/10/xml-exc-c14n#
```

5. The `Algorithm` attribute of the `SignatureMethod` element must be:

```
http://www.w3.org/2000/09/xmldsig#rsa-sha1
```

6. The `Algorithm` attribute of the `Transform` element for each `Reference` element must be either:

```
http://www.w3.org/2001/10/xml-exc-c14n#
```

or:

```
http://www.w3.org/TR/2001/REC-xml-c14n-20010315
```

7. The `Algorithm` attribute of the `DigestMethod` element for each `Reference` element must be:

```
http://www.w3.org/2000/09/xmldsig#sha1
```

8. The `KeyInfo` element must contain a `SecurityTokenReference` element. The `SecurityTokenReference` element must contain a `Reference` element with a `URI` attribute. The

`URI` attribute must use a local particle reference to identify the `BinarySecurityToken` element that contains the X.509 certificate (for example: the `URI` attribute equals `#CertId-1064304` in the preceding example request).

9. You must include a `wsu:Id` attribute in any message elements that you sign. You can sign any SOAP header and the entire SOAP `Body`. Do not sign any other elements (such as children of the `Body` element). AWS ignores those elements for the purposes of signature validation, even if you include a `wsu:ID` attribute in them. If you sign elements that shouldn't be signed, the signature validation will fail.

Using SOAP without WS-Security

This section describes how to authenticate SOAP requests without using WS-Security. The topics describe the basic requirements, the required authentication information, and where to place the information in the SOAP request.

General Requirements

If you plan to use SOAP without WS-Security:

- You can use either SOAP 1.1 or SOAP 1.2
- You must use HTTPS with your requests

Required Authentication Information

Authentication of SOAP requests without WS-Security uses your AWS identifiers and an HMAC-SHA256 signature. The request must include the parameters listed in the following table.

Parameter	Description
`AWSAccessKeyId`	Your AWS Access Key ID. For more information, see Your AWS Identifiers (p. 55).
`Timestamp`	This is a required parameter if you include the Signature parameter. Otherwise it is optional. There is no default value. The time stamp you use in the request must be a `dateTime` object, with the complete date plus hours, minutes, and seconds (for more information, go to http://www.w3.org/TR/NOTE-datetime).This is a fixed -length subset of the format defined by ISO 8601, represented in Universal Time (GMT): `YYYY-MM-DDThh:mm:ssZ` (where T and Z are literals). **Important** If you are using .NET you must not send overly specific time stamps, due to different interpretations of how extra time precision should be dropped. To avoid overly specific time stamps, manually construct `dateTime` objects with no more than millisecond precision.
`Signature`	The HMAC-SHA256 signature calculated from the concatenation of the *`Action`* and *`Timestamp`* parameters, using your AWS Secret Access Key as the key (for information about authentication with HMAC signatures, see HMAC-SHA Signatures for REST Requests (p. 57)). For example, for a request to create a queue, the value of the `Signature` element would be the HMAC-SHA256 digest of a string like this: `ItemLookup2011-09-24T00:00:00Z`

To calculate the signature

1. Concatenate the values of the *Action* and *Timestamp* request parameters, in that order. The string you've just created is the string you'll use when generating the signature.
2. Calculate an RFC 2104-compliant HMAC-SHA256 signature, using the string you just created and your Secret Access Key as the key.
3. Convert the resulting value to base64.
4. Pass this final value in the *Signature* parameter of the SOAP request.

Location of Authentication Information in the Request

With version 2011-08-01, you must provide the authentication information as elements in the SOAP header (using the namespace http://security.amazonaws.com/doc/2007-01-01/), as in the following example.

```
<?xml version="1.0"?>
<soap:Envelope
   xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
   soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header
   xmlns:aws="http://security.amazonaws.com/doc/2007-01-01/">
   <aws:AWSAccessKeyId>1D9FVRAYCP1VJS767E02EXAMPLE</aws:AWSAccessKeyId>
   <aws:Timestamp>2008-02-10T23:59:59Z</aws:Timestamp>
   <aws:Signature>SZf1CHmQnrZbsrC13hCZS061ywsEXAMPLE</aws:Signature>
</soap:Header>
...
</soap:Envelope>
```

Batch Requests

Topics

The requests presented so far contain only one operation. Product Advertising API enables you to improve performance by submitting more than one request at the same time. There are two ways to do this:

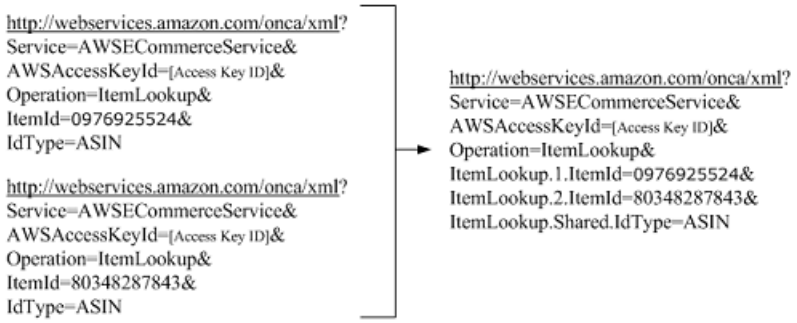
- **Batch request—**A request that uses one operation with up to two sets of parameters.
- **Multiple ItemIds—**In an `ItemLookup` request, you can include up to ten comma-separated values for *ItemId*.

Batch Requests

Topics

There might be times when you want to use the same operation in two requests, each one using different input parameters. For example, you might want to run an `ItemLookup` request several times, each time looking up a different item. To accomplish this task, you can submit two requests or one batch request.

The following figure shows how two simple requests can be combined into one batch request.



The new syntax introduced in the batch request centers on the ways in which the operation parameters are defined. Either the parameter values are different or they are shared between the simple requests in the batch request.

Batch requests can contain up to two sets of parameters.

Note

The Product Advertising API supports `ItemLookup` batch requests only when the *`ItemType`* is an EAN (European Article Number), ISBN (International Standard Book Number), EISBN (Electronic International Standard Book Number), or ASIN (Amazon Standard Item Number).

Parameters That Differ

Parameters that differ between the simple requests in a batch request are identified by a reference number. The following parameter declaration syntax shows where the reference number goes.

```
OperationName.ReferenceNumber.Parameter=Value
```

The following request snippet shows an example of a reference number, 1.

```
ItemLookup.1.ItemId=0976925524
```

The *`ReferenceNumber`* is a positive integer that associates the parameters in the simple request. For example, the following parameters are part of the same simple request because they use the same reference number.

```
ItemLookup.1.ItemId=0976925524&
ItemLookup.1.ResponseGroup=Images
```

These parameters might be different from a second simple request, which is part of the same batch request.

```
ItemLookup.2.ItemId=0485935524&
ItemLookup.2.ResponseGroup=Collections
```

So, if the previous two request snippets were part of one batch request, the `ItemLookup` operation would be executed twice, each time with a different *`ItemId`* and *`ResponseGroup`*.

In one batch request, you can have up to two different reference numbers. For example, the following request snippet exceeds the number of allowed reference numbers in one batch request.

```
ItemLookup.1.ItemId=0976925524&
ItemLookup.2.ItemId=0485935524&
ItemLookup.3.ItemId=0792335535&
```

Shared Parameters

There are times when the parameter values for the simple requests in a batch request are the same, for example, `ItemType=ASIN`. In this case, instead of using a ReferenceNumber for each simple request, you can substitute the special value, "Shared,", as shown in the following example.

```
ItemLookup.Shared.ItemType=ASIN
```

The following parameter declarations show the equivalent of the preceding declaration.

```
ItemLookup.1.ItemType=ASIN&
ItemLookup.2.ItemType=ASIN
```

Both forms of the request work but using the special value, Shared, reduces the amount of typing required. Generally, you only use the Shared value with required parameters. Optional parameters do not need to be included in the request and their default values, if any, are assumed and thus shared across the simple requests in the batch request.

Performing Multiple ItemLookups in One Request

The other way to execute multiple `ItemLookup` requests in one request is to use a comma-separated list of *`ItemIds`*, for example:

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B00008OE6I,B35987036I,B0002546I,B25468OE6I,B09788OE6I,B00453OE6I&
IdType=ASIN&
ResponseGroup=OfferFull&
Condition=All&
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The preceding example shows six *`ItemId`*s in the request, which means `ItemLookup` will be executed six times, one for each *`ItemId`*. You can specify up to ten *`ItemId`*s.

Tip

Be careful not to introduce spaces before or after the commas in the comma-separated list of *`ItemIds`*.

Parameters Common to All Product Advertising API Requests

Topics

Product Advertising API operations use a variety of required and optional parameters. The API Reference Guide describes the parameters that each operation can use.

There are, however, a number of parameters that all operations can use. These parameters can be grouped into types.

The following sections describe these parameter types.

Required Parameters

The parameters in the following table are required in every request.

Parameter	Value	Description
Service	AWSECommerceService	Specifies the Product Advertising API service
AWSAccessKeyId	Your Amazon-assigned Access Key ID	To register for an Access Key ID from the Amazon Web site, go to http://aws.amazon.com. Every Product Advertising API 4.0 request must contain either an Access Key ID or a Subscription ID but not both.
SubscriptionId	Your Amazon-assigned subscription ID	Every Product Advertising API 4.0 request must contain either an Access Key ID or a Subscription ID but not both. Starting with version 2005-10-05, Product Advertising API stopped distributing Subscription IDs. If you already have one, it will continue to work.
AssociateTag	An Amazon-assigned Associate ID	The *`AssociateTag`* enables product URLs returned by Product Advertising API to be tagged as originating from your Associates website. For your requesst to work and to receive a referral credit for a sale, you must include an AssociateTag value in all your requests. Be sure to specify the value correctly; no error is generated for incorrect values.
Operation	Operation you wish to perform, for example, `ItemLookup`	One of the Product Advertising API operation types.

Related Topics

- Operations (p. 176)

General, Optional Parameters

The parameters in the following table can be used optionally in any Product Advertising API request.

Parameter	Value	Description
MerchantId	An optional parameter that can be used to filter search results and offer listings to include only items sold by Amazon. By default, the API will return items sold by various merchants including Amazon.	The only valid optional value is “Amazon”.
ResponseGroup (p. 224)	Various	Specifies what subset of data to return. The API Reference Guide specifies the response groups that can be used with each operation..
Version	Various	Version of the Product Advertising API 4.0 WSDL. The default is 2005-10-05. If you want another version, including the latest, you must specify it in your request.

XSL Parameters

If you are using an XSL stylesheet to specify the output returned by Product Advertising API, you may need to include some or all of the following (optional) parameters.

Parameter	Value	Description
Style	"XML", the default, or the URL of an XSL stylesheet	The `Style` parameter applies to REST requests only. Use the `Style` parameter to control the format of the data returned by Product Advertising API. Set this parameter to "XML" to generate a pure XML response. Set this parameter to the URL of an XSL stylesheet to have Product Advertising API transform the XML response to another format you specify.
ContentType	"text/xml", the default, or "text/html"	The `ContentType` parameter is valid for REST requests only. The `ContentType` set in your request is returned as the content type in the HTTP headers of the response that Product Advertising API returns. Generally `ContentType` should only be changed when it is being used in conjunction with an XSLT stylesheet specified with the `Style` parameter. When you use a stylesheet to transform your Product Advertising API response to HTML, set this parameter to text/html.

For more information, go to Transforming Product Advertising API Responses into HTML Using XSLT (p. 86)

XML-Encoding Parameters

Another optional parameter is provided, that you may need in some environments, as described in the following table.

Parameter	Value	Description
XMLEscaping	"Single", the default, or "Double"	*XMLEscaping* specifies whether responses are XML-encoded in a single pass or a double pass. By default, *XMLEscaping* is "Single," and Product Advertising API responses are encoded only once in XML. For example, if the response data includes an ampersand character (&), the character is returned in its regular XML encoding (&). If *XMLEscaping* is "Double," the same ampersand character is XML-encoded twice (&). The Double value for *XMLEscaping* is useful in some clients, such as PHP, that do not decode text within XML elements.

Debugging Parameters

Process to Debut a Request

1	Use the *Validate* parameter described in the following table.

Parameter	Value	Description
Validate	"False", the default, or "True"	Use the *Validate* parameter to have Product Advertising API test your request without actually executing it. When present, *Validate* must equal "True." If the request is valid, the response will contain an element called IsValid with a value of True. If the request is invalid, the response will contain 1) an element called IsValid with a value of False and 2) the errors that would be returned if the request were actually executed. Notes: Since the request is not actually executed, only a subset of the errors for the request may be returned. This is because some errors (e.g., no_exact_matches) are only generated during execution of a request.

We also recommend that you include the *Request* response group for help with debugging. The Request response group echoes back the parameters used to generate the response and allows you to verify that your request was received correctly.

Responses

In response to a request, the Product Advertising API web service returns an XML document that contains the results of the call in an XML data structure. This data conforms to a WSDL and schema.

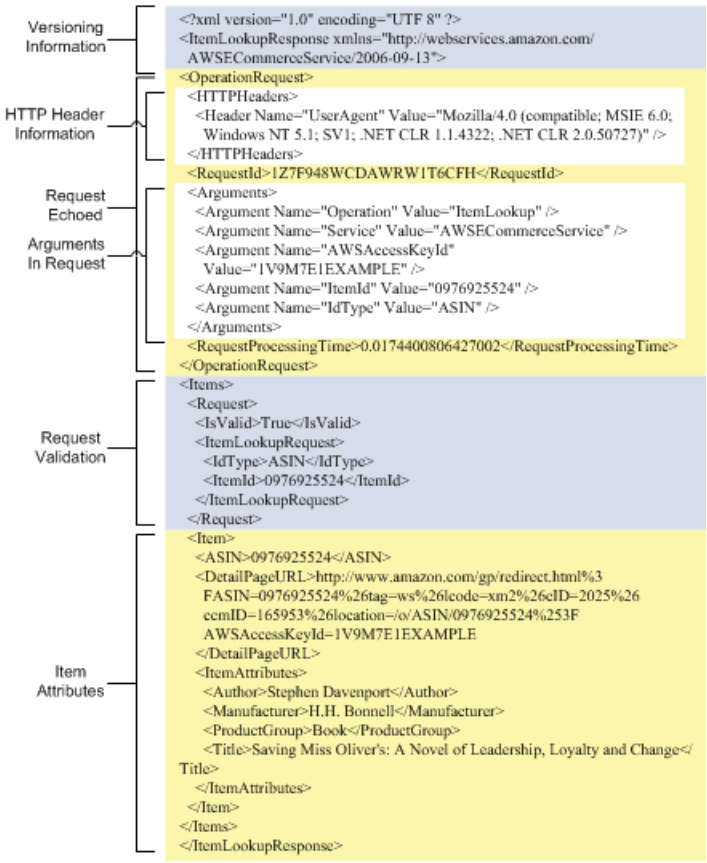
For REST requests, this data structure is simply the body of the HTTP response. You can use a data binding method for REST responses, or use an XML parser directly to process the information.

For SOAP requests, the data structure is the SOAP message body of the response. SOAP toolkits typically convert the response data into structures for use with your programming language, or allow you to specify your own data bindings.

Other than the use of a message envelope in the SOAP response, REST and SOAP responses are identical. Both conform to the Product Advertising API schema. The SOAP WSDL imports an XSD file to define the response messages. REST users can access the XSD file directly. For more information, see WSDL Location (p. 19).

Anatomy of a Response

Product Advertising API responses vary according to the operations and parameters submitted in the request. There are, however, segments of the response that are common across all responses, as shown in the following figure.



The following sections describe each segment in this response.

Version Information Segment

The first two lines of every response contain version information. The first line details the XML version used, which, in this example, is 1.0. The first line also includes the character encoding, which, in this example, is UTF-8.

The second line of every response contains the name of the response, which is named after the operation in the request. In this example, the `ItemLookup` operation returns a response element named `ItemLookupResponse`.

The second line also contains the version of Product Advertising API used to create the response. In this example, the version number is 2011-08-01. If a version is not specified in a request, Product Advertising API uses the default version, 2005-10-05. Make sure to specify the latest version number if you want the latest Product Advertising API functionality.

Operation Request Segment

The second segment of a response is always the Operation Request. For the most part, it returns the operation's parameters, called arguments, that were used in the request. There is, however, some additional information. The first large subsection is the HTTPHeader section, which specifies the network agent used to send the request. Typically, the agent, called the user agent, is a web browser and the information returned describes that browser, such as its name, version, and the operating system of its host.

The second large subsection contains the arguments, or operation parameters, used in the request. These values are unprocessed echoes of the values submitted in the request. This information is echoed for the benefit of troubleshooting requests.

Finally, interspersed around the large subsections in the Operation Request segment is Product Advertising API processing information, including the Request ID and the time it took to process the request. This is interesting information for troubleshooting. You will not, however, have to use either the Request ID or the processing time in future requests.

Request Validation Segment

The segments described so far are typically less interesting than the ones that follow. The Request Validation segment contains error information if there was an error in the request. It is also the last segment that Product Advertising API returns in a response if there was an error.

In the preceding example, the request was valid, as shown in the IsValid element.

```
<IsValid>True</IsValid>
```

The IsValid element pertains only to the validity of the request. It reports whether all of the required elements of a request are present, whether parameter restrictions and constraints have been obeyed, and whether parameter values are legal. IsValid does not assure that a valid request will be processed by Product Advertising API. For example, an *ItemId*, such as, 1234ABCD, is in the correct form, but no items in Amazon's catalog match that *ItemId*, so, the request would be valid but would return an error.

There are many reasons why a request can fail, as described in the following table.

Problem	Description
Incorrect parameters supplied	The request is missing parameters that an operation requires, or the combination of parameters is not allowed, or the Service parameter was omitted. In this case, the IsValid value is "False" and the request is not processed by Product Advertising API. Common mistakes that cause these errors are spaces in the URL that prevent Product Advertising API from reading the entire request, or misspelled and mis-capitalized parameter names or values. If you inadvertently enter a space in a request, your browser automatically enters the URL-encoded form of a space: %20. For example, if you paste, ItemId= B12345, into your browser as part of a request, the URL field in the browser shows, ItemId=%20B12345.
Invalid values	The request might be syntactically correct but the values for one or more parameters might be invalid. For example, you might supply an ItemId value that does not match any item sold on Amazon. In this case, the IsValid value is "True" but an error message is returned that says that the value is "not a valid value for ItemId. Please change this value and retry your request."
Empty result	The request is valid but there is no data that matches the search criteria. In this case, the IsValid value is "True." Remember that parameter names and values are case sensitive.
Network or processing error	The request is valid but a network outage or processing failure has caused Product Advertising API to return an empty response. In this case, the IsValid value is "True."
Partial request errors	Product Advertising API returns as much information as possible. For example, if you use a batch request and Product Advertising API finds items for one request but not the other, Product Advertising API returns all of the information for the found item and an error message for the other request.

When one of these errors occurs, the IsValid value is False and Product Advertising API returns an error element in the Request Validation segment. The error element includes:

- **Error code—**The title of the error
- **Error message—**A brief description of the error condition

The following example shows an error message.

```
<Errors>
  <Error>
    <Code>missing_service_parameter</code>
    <message>Your request is missing the Service parameter. Please add the
Service parameter to your request and retry. Valid values for the Service
parameter include AWSEcommerceService.</message>
  </Error>
</Errors>
```

The more difficult error to assess is a request that is valid, that is processed, but contains an error in logic. For example, a request might use default values such that only items in new condition are returned when the requester meant to return items in all conditions.

The remainder of the Request Validation segment is a subsection named after the operation used in the request. In this example, the subsection is named `ItemLookup`Request. This subsection echoes the parameter values used in the request. These values are returned for the purpose of troubleshooting the request.

Item Attributes Segment

Assuming that your request is valid, the final segment of the response that contains the item attributes is typically the part of the response that contains the most value. This segment contains all of the descriptive information about the items that satisfied the request. In the preceding example, one item was found that satisfied the search criteria. Many attributes are returned that describe that item, including, the title, "Saving Miss Oliver's: A Novel of Leadership, Loyalty and Change."

The item attributes returned in a response vary according to the response group used and the items found. Merchants do not always provide Amazon with a single set of item attributes. For example, one merchant might supply the name of a book's publisher, another might not. A response contain both of these books would therefore contain a different set of item attributes. Item attributes that do not have values are not returned. The API Reference Guide describes the item attributes that can be returned by each response group.

Paging and Sorting Through Responses

Topics

The only drawback of having so many items at your fingertips is the possibility of receiving too many in a response. Product Advertising API handles this problem in several ways:

- Results are returned on page, generally, up to ten results per page
- The Sort parameter orders results

Paging Through Results

It is possible to create a request that returns many thousands of items in a response. This is problematic for several reasons. Returning all of the item attributes for those items would dramatically impact the performance of Product Advertising API in a negative way. Also, posting a thousand responses on a web page is impractical.

For that reason, Product Advertising API developed the strategy of returning results a little at a time. The good news is that you can return any page of results. For example, the first request can return the last

page of results. To do that, you have to specify the desired page of results using one of the parameters that enable you to return result pages.

To page through results

- Use the appropriate paging parameter in the request. Operations have their own paging parameters. For example, the following `ItemSearch` request uses ItemPage to ask for the fourth page of results.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Keywords=Potter&
SearchIndex=Books&
ItemPage=4
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The following snippet of the response shows that the fourth page of results has been returned.

```
<ItemSearchRequest>
  <ItemPage>4</ItemPage>
  <Keywords>Potter</Keywords>
  <SearchIndex>Books</SearchIndex>
</ItemSearchRequest>
</Request>
<TotalResults>9729</TotalResults> <TotalPages>973</TotalPages>
```

This example shows that 9729 items matched the search criteria. Also, it shows that those results are on 973 (~9729/10) pages. You might try putting in an *ItemPage* value over 10. If you do, Product Advertising API returns the following error.

```
<Error>
  <Code>AWS.ParameterOutOfRange</Code>
  <Message>The value you specified for ItemPage is invalid. Valid values must
be between 1 and 10.</Message>
</Error>
```

So, how do you get that 973rd page? You cannot. A better approach is to submit a new request that is more targeted and yields fewer items in the response.

Maximum Number Of Returned Pages

The last example showed that Product Advertising API returns only so many pages for any one request. This is because performance must be optimized for the tens of thousands of Product Advertising API developers and customers. For that reason, many Product Advertising API operations have pagination parameters and associated maximum values, as shown in the following table.

Operation	Parameter Name	Maximum Page Number
`ItemLookup`	VariationPage	150
`ItemLookup`	ReviewPage	20
`ItemSearch`	ItemPage	10

These parameters are defined in the section of the API reference that describes the associated operation.

Sorting Results

The following Product Advertising API operations have a *Sort* parameter that arranges results:

- `ItemSearch`

To Sort Results

1. Consult the appendix, APPNDX_SortValuesArticle Sort Values, to determine available sort values. Available sort values vary by locale and search index.
2. Add the Sort parameter to a request that uses one of the preceding operations.

For example, the following request returns books with "Harry Potter" in their title or description in alphabetical order.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=1MEXAMPLEZBG2&
Operation=ItemSearch&
Keywords=Harry%20Potter&
SearchIndex=Books&
Sort=titlerank&
ItemPage=4&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

A small snippet of the response shows that the book titles are returned in alphabetical order.

```
<Title>Aventuras de Harry Potter, Las</Title>
...
<Title>Beacham's Sourcebook For Teaching Young Adult Fiction: Exploring Harry
Potter</Title>
...
<Title>Beatrix Potter to Harry Potter: Portraits of Children's Writers</Title>
```

Product Advertising API provides many different sorting criteria, for example, price (high to low, or low to high), salesrank (best to worst selling, or worst to best selling), publication date, review rank, and release date. Valid sort parameters vary by search index, for example, the DigitalMusic search index can be sorted by UploadedDate. That value for *Sort* would not make sense in the Automotive search index, for example. Sort parameters also differ by locale.

For more information about sort values by locale and search index, see the appendix, ItemSearch Sort Values By Locale (p. 325).

Default Sort Values

There are many sort values. The majority are not applied unless the *`Sort`* parameter is included in the request. There are two sort values, however, that are used by default.

- For `ItemSearch` requests that do not use the *`BrowseNode`* parameter, results are sorted by Relevance

- For `ItemSearch` requests that do use the *`BrowseNode`* parameter, results are sorted by BestSeller ranking

Transforming Product Advertising API Responses into HTML Using XSLT

Topics

Perhaps you want to use a different set of tags in Product Advertising API responses than those that are returned by default. For example, because you want to display responses on a web page, you want to turn the responses into HTML. You have two choices: you can receive the default Product Advertising API response and then transform it into HTML (or another set of XML tags) or you can tell Product Advertising API to do the transformation for you so that the result is ready for you to use. To make Product Advertising API do the work for you, you just need to reference an XSL stylesheet in your Product Advertising API request.

Product Advertising API provides an XSLT (Extensible Stylesheet Language Transformation) service to ensure that even novice developers can produce rich content without complex parsing or programming. XSL is an XML-based language for transforming XML tags into HTML or any other set of XML tags. To use the Product Advertising API XSLT service, the request must be in REST, and the XSL style sheet must be referenced using the Style input parameter.

To transform the response

- Include the *`Style`* parameter in your request.

  The referenced stylesheet must be publicly accessible.

For example, the following `ItemSearch` request specifies the XSLT stylesheet *http://ecs.amazonaws.com/xsl/aws4/item-search.xsl*.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
IdType=ASIN&
ItemId=B00008OE6I&
ResponseGroup=Large&
Style=http://ecs.amazonaws.com/xsl/aws4/item-search.xsl
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

XSLT Service URLs

For security reasons, Amazon Web Services use a different set of domains for processing XSLT requests. XSLT requests sent to the standard domains (for example, *http://ecs.amazonaws.com*) are re-directed. If your web client does not support re-directs, you must send your XSLT requests directly to one of the domains, as shown in the following table.

Locale	URL
CA	xml-ca.amznxslt.com
DE	xml-de.amznxslt.com
FR	xml-fr.amznxslt.com
JP	xml-jp.amznxslt.com
UK	xml-uk.amznxslt.com
US	xml-us.amznxslt.com

Creating an XSL Style Sheet

It is beyond the scope of this book to explain in detail how to create an XSL style sheet. To see a sample style sheet, go to http://ecs.amazonaws.com/xsl/aws4/item-search.xsl.

You use "xmlns" to define a namespace prefix at the top of the style sheet. You may name the namespace prefix anything, however, the namespace URL must match the namespace URL that is returned in the Product Advertising API response. The preceding example uses "aws" as the namespace prefix, as follows.

```
xmlns:aws="http://xml.amazon.com/AWSECommerceService/2011-08-01">
```

Once you have created a namespace prefix, use it to match elements in the Product Advertising API response. For example, if you are trying to match an element called "`ItemLookup`Response," and your prefix is "aws," the matching string would be aws:`ItemLookup`Response.

The following example shows how and where the namespace prefix should be used.

```
<xsl:template match="/">
  <xsl:apply-templates select="aws:Items/aws:Item"/>
</xsl:template>
<xsl:template match="aws:Items/aws:Item">
  <tr>
    <td style="border-bottom:C0C0C0 dotted 1px;padding:10px">
      <table cellpadding="0" cellspacing="0" style="width: 90%;padding:5px">
        <tr>
          <xsl:if test="aws:SmallImage/aws:URL">
            <td valign="top" width="50">
              <img>
                <xsl:attribute name="src">
                  <xsl:value-of select="aws:SmallImage/aws:URL" />
                </xsl:attribute>
                <xsl:attribute name="border">0</xsl:attribute>
              </img>
            </td>
          </xsl:if>
          <td valign="top">
            <xsl:value-of select="aws:ItemAttributes/aws:Title" />
            <br />
            <span style="font-size:10px">
              <xsl:if test="aws:ItemAttributes/aws:Author">
                by <xsl:value-of select="aws:ItemAttributes/aws:Author" />
```

```
              </xsl:if>
              <xsl:if test="aws:ItemAttributes/aws:Artist">
                by <xsl:value-of select="aws:ItemAttributes/aws:Artist" />
              </xsl:if>
              <xsl:if test="aws:ItemAttributes/aws:Director">
                by <xsl:value-of select="aws:ItemAttributes/aws:Director" />
              </xsl:if>
              <xsl:if test="aws:ItemAttributes/aws:Composer">
                by <xsl:value-of select="aws:ItemAttributes/aws:Composer" />
              </xsl:if>
              <xsl:if test="aws:ItemAttributes/aws:Manufacturer">
               from <xsl:value-of select="aws:ItemAttributes/aws:Manufacturer"
 />
              </xsl:if>
            </span>
            <br />
            <br />
            <span style="font-size:11px;">
              List Price: <xsl:value-of
                select="aws:ItemAttributes/aws:ListPrice/aws:FormattedPrice"
/>
            </span>
          </td>
        </tr>
      </table>
    </td>
  </tr>
</xsl:template>
```

Finding Items to Buy

Topics

In the previous chapter, you learned how Amazon groups and catalogs items for sale. In the following sections, you learn how to use Product Advertising API operations to find those items.

Search Operations

The following table describes the Product Advertising API operations that find items.

Operation	Description
ItemSearch	To find items sold by merchants and sellers. This operation is the one used most often to find items for sale.

Operation	Description
BrowseNodeLookup	To find items associated with browse nodes by navigating through the browse node hierarchy. This operation does not return items directly, only the browsenodes associated with items.

If you do not know whether an item is for sale by a merchant or seller, search first using `ItemSearch` for two reasons:

- Most of the items sold on Amazon are sold by merchants.
- Sellers often list their items for sale in Amazon's marketplace as well as in their own stores. Anything sold in the marketplace can be found by `ItemSearch`.

Items You Cannot Buy

You might assume that any item returned by one of the search operations is for sale. This, however, is not the case. Some items, such as Variation parents and Collection parents, are never for sale. Before presenting items to customers, you must make sure the items can be purchased.

Product Advertising API provides several ways to determine if an item can be purchased, for example, `ItemSearch` has an *Availability* parameter, which specifies whether an item is available for purchase. The best way, however, to determine whether or not an item can really be purchased is by using one of the following response groups in a request:

- Offers Response Group (p. 267)
- OfferFull Response Group (p. 263)
- Large Response Group (p. 250)

Note

Large is a parent of the Offers response group.

All of the search operations listed, except `BrowseNodeLookup`, can use these response groups. These response groups return an OfferListingId element for each item that can be purchased. Items that do not have an OfferListingId cannot be purchased.

The following response snippet shows that an item can be purchased because the response contains an *`OfferListingId`*.

```
<Item>
  ...
  <Offer>
    ...
    <OfferListing>
      <OfferListingId>g7CWSnBZmVESKy%2BCNS</OfferListingId>
      <Price>
        <Amount>625</Amount>
        <CurrencyCode>USD</CurrencyCode>
        <FormattedPrice>$6.25</FormattedPrice>
      </Price>
      <Availability>Usually ships in 1-2 business days
      </Availability>
    </OfferListing>
```

```
  </Offer>
</Item>
```

This response shows how the *OfferListingId* element is a child of the Item element.

The Availability element in the preceding example is not a reliable means of determining whether or not an item is truly available for purchase. Its message concerns how quickly an item can ship when it is in stock. A value returned for Availability does not guarantee that the item is in stock.

Tip

The next chapter talks about adding items to a customer's shopping cart. While it is possible to add an item to a cart using an ASIN, the preferred means of adding an item is by specifying an *OfferListingId* because an *OfferListingId*, not an ASIN, guarantees that an item can be purchased.

Availability Parameter

The *Availability* parameter filters out of ItemSearch results those items that are unavailable. The availability of an item can change rapidly. There is typically a discrepancy between an item's availability as reported by ItemSearch and the item's true availability, as reported by Amazon's web site. For this reason, the availability of items reported by ItemSearch and by Amazon's web site will be slightly different. Items that are "available" are classified on Amazon's retail web site as:

- Currently for sale
- Pre-orders
- Special orders
- New releases
- E-mail me when items become available
- Items available for in-store pickup
- Items for sale by third parties

Parameter Restrictions for Availability

The following table describes the ItemSearch parameters that must be included to return available items (only).

ItemSearch Parameter	Description
Availability	Must be set to "Available." When the Availability parameter is not set, ItemSearch returns available and unavailable items. "Available" is the only valid value for *Availability*. Setting it to another value returns an error message. Parameter values are case sensitive. When the Availability parameter is set to "Available", the only optional constraint that can be present is MerchantId.

ItemSearch Parameter	Description
MerchantId	An optional parameter that can be used to filter search results and offer listings to only include items sold by Amazon. By default, the API will return items sold by various merchants including Amazon. The only valid optional value for MerchantId is "Amazon". When the Availability parameter is used and MerchantId is set to "Amazon," the availability results for Amazon, Toys R Us, and Target are merged.

The following search indices do not work with the *`Availability`* parameter:

- Items available for in-store pickup
- Items for sale by third parties

In both cases, because Amazon does not warehouse the items for sale, Amazon cannot determine the availability of them.

Items Sold by Merchants

By far, most items on Amazon are sold by merchants. Amazon itself is one. For that reason, ItemSearch (p. 207) is one of the most heavily used Product Advertising API operations. Because there are so many items for sale on Amazon, `ItemSearch` has many input parameters that help target a request. `ItemSearch` requests can also use many response groups, which can tailor the type of information returned with each item..

At first, the number of ItemSearch input parameters might seem overwhelming. This section explains the parameters most commonly used. By mastering these, you can create highly targeted requests. For a description of all of the `ItemSearch` parameters, see ItemSearch.

Required ItemSearch Parameters

`ItemSearch` can potentially return any item sold by a merchant and most sellers. Because there are so many items, `ItemSearch` requires that you specify a search index and at least one additional parameter, as shown in the following example.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
SearchIndex=Books&Keywords=Saving%20Miss&20Oliver's
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

In this request, `ItemSearch` looks for "Saving Miss Oliver's" in the titles and descriptions of all of the items in the Books search index. The response would include all matches.

The previous example used the *Keywords* parameter as the second parameter, which is one of the most popular parameters to use. The following list shows all of the `ItemSearch` parameters that you could have used instead.

- Actor
- Artist
- AudienceRating
- Author
- Brand
- BrowseNode
- City
- Composer
- Conductor
- Director
- Keywords
- Manufacturer
- MusicLabel
- Neighborhood
- Orchestra
- Power
- Publisher
- Title

A cursory look at the list of parameters shows you that most are very specific, which makes it easy to determine whether or not they should be used in a request. The Publisher parameter, for example, is used only to specify the publisher of a book. Once you know the item you are looking for, choosing which parameter to use becomes easy.

Parameter Support by Search Index

Each search index supports only a subset of all `ItemSearch` parameters. For example, in the US locale, when you specify the Blended search index, you can only use one parameter: *Keywords*, in a request. Using any of the other parameters in that request would return an error.

All other search indices support multiple parameters. For example, when you specify the Beauty search index, in the US locale, you can include one or more of the following parameters in the request.

- Brand
- BrowseNode
- Condition
- ItemPage
- Keywords
- Manufacturer
- MaximumPrice
- MinimumPrice
- Sort
- Title

The search index—parameter combinations, for the most part, make sense. For example, if you use the Automotive search index, the *`Author`* parameter is invalid in the request. This restriction makes intuitive sense. So, in general, it is not the case that you need to memorize all of the parameters that can be used in each search index.

Search Indices, Parameters, and Locales

There is one further restriction on the use of parameters in an `ItemSearch` request. The search index values that can be used in an ItemSearch (p. 207) request vary by locale. For example, the Baby and Beauty search indices are available in the US but not in the UK locale. An error is returned if you use a search index value that is not supported in a locale. The limitations on the use of search indices by locale therefore also limits the use of `ItemSearch` parameters.

For a complete list of `ItemSearch` parameters supported in each search index and in each locale, see the Search Index and ItemSearch Parameter Combinations (p. 392).

Common ItemSearch Parameters

Topics

- Availability Parameter Restrictions and Constraints (p. 95)
- Searching Across Indices (p. 95)

Once you decide on a search index value and check Search Index and ItemSearch Parameter Combinations (p. 392)to determine what ItemSearch (p. 207) input parameters are valid for that search index, you should set values for as many valid parameters as you can to increase the accuracy of the search results.

The most commonly used input parameters are the ones that can be used with the most search indices, including:

- Availability—Specifies that the item must be available for purchase. The only valid value for the parameter is "Available."
- BrowseNode—Enables you to search a specified browse node for associated items.
- Condition—Enables you to specify the condition of an item. Valid values are "All," "New," "Used," "Collectible," and "Refurbished." The default is "New." Condition does not restrict the total number of items returned. It does, however, restrict the offers returned to those items that are in the specified condition.
- Keywords—A word or phrase (words separated by xml-encoded spaces, %20) used as a search criteria. The titles and descriptions of items are searched for keywords.
- MaximumPrice—The maximum price that an item can cost.
- MinimumPrice—The minimum price that an item can cost.
- Title—A word or phrase (words separated by xml-encoded spaces, %20) used as a search criteria. The titles of items are searched for these words.

The definitions for all `ItemSearch` parameters are in the API Reference. The following sections describe in greater detail some of the commonly-used parameters.

Returning Only Available Items

`ItemSearch` returns available and unavailable items. Unavailable items are, for example, items that are temporarily out of stock. By setting the *Availability* parameter to "Available," `ItemSearch` can filter out most of the items that are unavailable. The following `ItemSearch` request returns shirts that are available.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=All&
Availability=Available&
SearchIndex=Apparel&
Keywords=Shirt
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

"Available" is the only valid value for the parameter. Setting it to any other value returns an error message. By default, the value is not set.

Items that are “available” are classified on Amazon’s retail web site as:

- Currently for sale
- Pre-orders
- Special orders

- New releases
- E-mail me when items become available
- Items available for in-store pickup
- Items for sale by third parties

The availability of an item can change rapidly. There is typically a discrepancy between an item's availability as reported by `ItemSearch` and the item's true availability, as reported by Amazon's web site. For this reason, the availability of items reported by `ItemSearch` and by Amazon's web site can be slightly different. These differences are most evident for items, such as software, whose availability changes often.

Note

Do not confuse the *Availability* parameter with the return element, Availability. The latter is an element in a response that describes how soon an item can be shipped, such as, "Ships in 48 hours." While this is a helpful value to display, it does not guarantee that an item is truly available to be purchased. For that determination, test for an OfferListingId, which is provided by the Offers, OfferFull, and Large response groups.

Inaccurate Availabilities

The *Availability* parameter does a good job of filtering out of the response items that are unavailable. The parameter, however, is not foolproof. It is possible for an item to be unavailable for purchase even if:

- The item is returned in a response
- The response has been filtered using the *Availability* parameter
- The Availability element in the response says the item can be shipped in twenty-four hours

For example, some items, including Variations (p. 40) parents and Collections (p. 44) parents, are never available for purchase because they do not represent real items; they are abstractions. These parent items, however, are often returned as being available. The following request returns a Collection parent item even though the *Availability* parameter is used.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B0006PLAOE&
Availability=Available&
ResponseGroup=ItemAttributes,Offers&
Condition=All
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

A snippet of the response shows that the Collection parent item is returned, so you might assume that it is available. However, the item is not associated with any offers, so it cannot be purchased.

```
<Item>
  <ASIN>B0006PLAOE</ASIN>
  ...
  <Offers>
    <TotalOffers>0</TotalOffers>
```

The contradiction of an item being available but not something you can buy shows that the *Availability* parameter is helpful but not the final arbiter in determining what items a customer can actually purchase. Instead, as was mentioned at the beginning of this chapter, a better test for availability is whether or not an item has an OfferListingId. Also, in the preceding example, you can see that you can test the value of TotalOffers to determine if an item has an offer.

Process to Include Item's Availability

1	Include in an ItemSearch request `Availability = Available`.
2	Include in the request `ResponseGroup = Offers`.
3	Test to see that the value in the response for *`TotalOffers`* is greater than zero.

Even though the *`Availability`* parameter is not foolproof, it does have value. The advantage gained by using the parameter is that it prevents the return of items that certainly do not have an *`OfferListingId`*. This filtering reduces the amount of testing you need to do to make sure the returned items have an *`OfferListingId`*.

Availability Parameter Restrictions and Constraints

The *`Availability`* parameter carries with it a restriction and some constraints. The restriction is that the *`Availability`* parameter cannot be used with the Blended search index. When the Availability parameter is set to "Available", the only optional constraint that can be present is MerchantId=Amazon.

Parameter	Description
MerchantId	An optional parameter that can be used to filter search results and offer listings to only include items sold by Amazon. By default, the API will return items sold by various merchants including Amazon. The only valid optional value for MerchantId is "Amazon".

Searching Across Indices

`ItemSearch` requests require that you specify a search index. This is because searching across the millions of products in Amazon databases would take too long. Product Advertising API does, however, enable you to search across multiple search indices using the All or Blended search indices.

All Search Index

You can use the All search index to do an `ItemSearch` search through all search indices. There are, however, a number of restrictions placed on this request: the only parameter that you can use in the request is *`Keywords`*, and you cannot, for example, sort results.

Note

You cannot use the All search index in an `ItemLookup` request.

Blended Searches

`ItemSearch` searches through a specified search index, or *`SearchIndex`* can be set to "Blended." A blended search always searches through the following search indices (only).

• Apparel • Automotive • Books • DVD • Electronics • GourmetFood	• Kitchen • Music • PCHardware • PetSupplies • Software • SoftwareVideoGames	• SportingGoods • Tools • Toys • VHS • VideoGames

Power Searches

Topics

`ItemSearch`'s *Power* parameter, which can only be used when the search index equals Books, enables you to use simple or complex queries to perform book searches. A simple power search would be to return all books written by a specified author. While this query is valuable, it does not offer anything more than a similar search using the *Author* parameter in an `ItemSearch` request.

The true value of power searches comes in the ability to specify multiple criteria to search for books. For example, you could create a query that returned all books by a specific author, published during a specified year, in a specified language, and printed in a specified format.

Power Search Syntax

The individual parts of a power search query are of the following form

```
key:value
```

For example, the following query string searches for books authored by Davenport.

```
author:Davenport
```

Values are not case sensitive. So, "Davenport" returns the same books as "davenport."

The individual parts of a power search query are linked together using a variety of operators. The following example searches for all books authored by Davenport that were published during 2006.

```
author:Davenport and pubdate:during 2006
```

Other operators, such as parentheses, can be used to create additional specificity. For example, a query of `subject:history and (spain or mexico) and not military and language:spanish` would return a list of books in the **Spanish** language on the **subject** of either **Spanish** or **Mexican history**, excluding all items with **military** in their description.

These parameter values would be part of an `ItemSearch` request that used the *Power* parameter, as shown in the following example.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
```

```
AWSAccessKeyId=[ID]&
Operation=ItemSearch&
SearchIndex=Books&Power=subject:history%20and%20(spain%20or%20mexico)%20and%20not%20military%20and%20language:spanish
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Note that spaces must be URL-encoded (%20) as shown.

Power Search Keywords

Power search queries are created using one or more keyword:value pairs. The keywords you can use in a power search are described in the following table.

Keyword	Description
after	Restricts the results to books published after the specified date. This keyword is used with pubdate, for example, pubdate:after 2006.
ASIN	An alphanumeric token that uniquely identifies a book.
author	Book's author.
author-begins	Author's name begins with this value.
author-exact	Authors name is exactly this value.
binding	Searches for books with the specified binding. Values for binding include abridged and large print.
during	Searches for books that were published during the time period specified. This keyword is used with pubdate, for example, pubdate:during 2006.
EISBN	Electronic International Standard Book Number of the digital book.
ISBN	International Standard Book Number of the book.
keywords	Words that can be found in the title or description of a book.
keywords-begin	Search for all words tat begin with this value.
language	Language, such as Spanish, that the book is written in.
pubdate	Book's publication date.
publisher	Name of the book's publisher.
subject	Find books with the specified word in their subject description.
subject-begins	Find books whose subject description begins with this word or character.
subject-words-begin	Find books whose subject description begins with these words
title	Title of the book.
title-begins	Title of the book begins with this value exactly. This value does not mean "begins with," it means "begins." For the example, if the value is "I", the book, "I Lied," would be returned; the book, "Isabel," would not be returned.
title-words-begin	Title of the book begins with this word.

Power Search Operators

The following table describes the Power search queries operators that you use to build queries.

Header 1	Header 2
not	Excludes the following parameter from the results, for example, subject:history and not military, excludes military history in the results.
and	Specifies that both values must be true to be selected. For example, subject:history and (Spanish and Mexican), requires that the books selected contain both Spanish and Mexican history.
or	Exclusive or which means one of either item but not both. For example subject:history and (Spanish or Mexican), means the subject matter can be about the history of Spain or Mexico, but not both.
colon (:)	Used as an equals sign, for example, subject:history, searches for books whose subject matter is history.
parenthesis	Groups terms to clarify operations, for example, `subject: history and (spain or mexico)` Without the parentheses, you would search for books about Spanish history or Mexico. With parentheses, you search for books about Spanish history and Mexican history.
asterisk (*)	Stands for zero or more alphanumeric characters, for example, `keywords:high%20tech*` Some of the keywords used in a search would be "high tech," "high technology," and "high technologies." Use the asterisk to generalize your search.
Quotation marks (“ “)	Specifies an exact match of the word(s) within the quotes, for example, `keywords:"Saving Miss Oliver's"`

Example Power Searches

The following five Power Search examples demonstrate many of the key words and operators.

```
author: ambrose and binding: (abridged or large print) and pubdate: after 11-
1996
subject: history and (Spain or Mexico) and not military and language: Spanish
(subject: marketing and author: kotler) or (publisher: harper and subject: "high
 technology")
keywords: "high tech*" and not fiction and pubdate: during 1999
isbn: 0446394319 or 0306806819 or 1567993850
```

Power Search Tips

Topics

The following tips help you avoid common problems when using power search key words.

Exact Matches

title-begins require exact matches. So, the following examples return completely different results.

```
title-begins:I
title-begins:I*
```

The first example would return the book, "I Lied," but it would not return the book, "Isabel." The second example would return both books. This behavior also applies to:

- author
- author-begins
- author-exact
- title
- title-words-begin

Discarded Search Words

The search engine automatically discards small words, such as "A" and "The." So, for example, if you created queries using either of the following key:value pairs. `title-begins:A` `title-begins:The` you would get zero results because the search engine discards those words when searching through titles. This behavior is also true for:

- author
- author-begins
- author-exact
- title
- title-words-begin

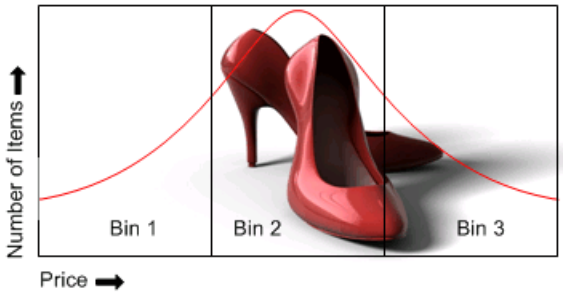
Using Search Bins to Find Items

Topics

Up to this point, you have seen how you can use operation input parameters and response groups to filter out unwanted responses. The SearchBins response group provides a different means of refining results. It enables you to filter results based on values returned in a response.

The SearchBins response group categorizes the items returned by `ItemSearch` into groups, called bins. The grouping is based on some criteria, depending on the search index. For example, a set of bins can be based on a set of price ranges for an item. In the case of women's shoes, for example, SearchBins

might return a bin that contains ASINs for shoes that cost between $0 and $50, a second bin for shoes that cost $50 to $100, and a third bin for shoes that cost more than $100.



The advantage of using search bins is that the response group divides the items into bins without you having to return or parse item attributes. You can then submit a second `ItemSearch` request and return only the items in one bin.

You cannot create bins nor can you specify the criteria used to divide the items into groups. The SearchBins response group does that automatically.

Some search indices support more than one kind of bin. For example, apparel items can be divided in to bins according to price range and brand. In this case, the response would return multiple sets of bins, called SearchBinSets, in which the items would be divided according to different criteria.

The criteria used to divide the returned items into bins is called the NarrowBy value.

Note

Search bins are only available in the US locale.

Related Topics

- SearchBins Response Group (p. 283)

NarrowBy Attribute

The basis on which returned items are split into bins is specified by the *`NarrowBy`* attribute in the SearchBinSet element. In the following example, the NarrowBy attribute shows that the bins are based on price range.

```
<SearchBinSet NarrowBy="PriceRange">
```

For another product category, the *`NarrowBy`* attribute might be different, for example:

```
<SearchBinSet NarrowBy="BrandName>
```

You cannot specify *`NarrowBy`* values nor can you specify the values they encompass. When the SearchBins response group is included in a request, `ItemSearch` automatically divides the `ItemSearch` results into bins.

Search Bin Example

The following request uses the SearchBins response group to return search bins.

```
Service=AWSECommerceService&
AWSAccessKeyId=[Access Key ID]&
Operation=ItemSearch&
SearchIndex=Baby&
Keywords=pants&
Availability=Available&
Condition=All&
ResponseGroup=SearchBins
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The following xml is a snippet from the response.

```
<SearchBinSets>
  <SearchBinSet NarrowBy="PriceRange">
  <Bin>
    <BinName>$0$24</BinName>
    <BinItemCount>1645</BinItemCount>
    <BinParameter>
      <Name>MinimumPrice</Name>
      <Value>0</Value>
    </BinParameter>
    <BinParameter>
      <Name>MaximumPrice</Name>
      <Value>2499</Value>
    </BinParameter>
  </Bin>
  <Bin>
    <BinName>$25$49</BinName>
    <BinItemCount>647</BinItemCount>
    <BinParameter>
      <Name>MinimumPrice</Name>
      <Value>2500</Value>
    </BinParameter>
    <BinParameter>
      <Name>MaximumPrice</Name>
      <Value>4999</Value>
    </BinParameter>
  </Bin>
  <Bin>
    <BinName>$50$99</BinName>
    <BinItemCount>173</BinItemCount>
    <BinParameter>
      <Name>MinimumPrice</Name>
      <Value>5000</Value>
      </BinParameter>
    <BinParameter>
      <Name>MaximumPrice</Name>
      <Value>9999</Value>
    </BinParameter>
  </Bin>
```

This response snippet shows the first three bins in the response. The NarrowBy value shows that the items were divided up based on price range. The BinName element names the bin. The names are descriptive of the price ranges that each bin represents. For example, the BinName, $50$99, contains items that cost between $50 and $99.99, which you can see by the values returned for MinimumPrice

and MaximumPrice in that bin. The BinItemCount element shows how many items are in each bin, for example, there are 173 items in the last bin.

The BinParameter/Value elements show the values used to create the bins. In this example, the parameters are the minimum and maximum prices of the items in that bin. For example, in the last bin, the minimum price of an item in that bin is $50.00 and the maximum value is $99.99.

The BinParameter/Name value, such as MaximumPrice, is an `ItemSearch` parameter name. This means that you can use the <Value> as the value for the parameter named by <Name> in a subsequent `ItemSearch` request. In this example, *MinimumPrice* is the `ItemSearch` parameter and, in the last bin, the value is 5000. By submitting a second request using `ItemSearch`'s parameters, *`MinimumPrice`* and *`MaximumPrice`* , you could return the item attributes for only the items in that bin.

As you can see from this example, the SearchBins response group enables you to narrow your search without you having to parse through item attributes.

NarrowBy Values

The following table describes the NarrowBy values.

NarrowBy Values	Description
BrandName	Brands, such as Levi's, Reebok, and Nike, that create the item. Use the name of a brand to filter out similar items made by other companies.
PriceRange	Minimum and maximum prices for a bin of items. Use the minimum and maximum price values in each bin to filter out items outside of the price range you want.
SpecialSize	Uncommon sizes an item comes in. Examples are "Plus Size & Tall," "Misses," "Maternity," "Husky," "Petites," and "Big & Tall."
Subject	BrowseNode IDs of all topics related to items returned by `ItemSearch`. For example, searching for books about dogs returns, in the Subject bins, BrowseNodes for "Home & Garden," "Animal Care & Pets," "Dogs," and "Educational."

All search indices return the Subject bin. Most return all of the bins.

NarrowBy Values Per Search Index

The following table shows which NarrowBy values are returned by each search index:

Search Index	NarrowBy
Apparel	Subject, BrandName, PriceRange, SpecialSize
Baby	Subject, BrandName, PriceRange, SpecialSize
Beauty	Subject, BrandName, PriceRange, SpecialSize
Blended	Not supported
Books	Subject
Classical	Subject
DVD	Subject

Search Index	NarrowBy
Electronics	Subject, BrandName, PriceRange, SpecialSize
Garden	Subject, BrandName, PriceRange, SpecialSize
GourmetFood	Subject, BrandName, PriceRange, SpecialSize
HealthPersonalCare	Subject, BrandName, PriceRange, SpecialSize
Jewelry	Subject, BrandName, PriceRange, SpecialSize
Kitchen	Subject, BrandName, PriceRange, SpecialSize
Magazines	Subject
Marketplace	Subject
Miscellaneous	Subject, BrandName, PriceRange, SpecialSize
Music	Subject
MusicalInstruments	Subject, BrandName, PriceRange, SpecialSize
OfficeProducts	Subject, BrandName, PriceRange, SpecialSize
PCHardware	Subject, BrandName, PriceRange, SpecialSize
Photo	Subject, BrandName, PriceRange, SpecialSize
Showtimes	Theater
Software	Subject, BrandName, PriceRange, SpecialSize
SportingGoods	Subject, BrandName, PriceRange, SpecialSize
Theatrical	Subject
Tools	Subject, BrandName, PriceRange, SpecialSize
Toys	Subject, BrandName, PriceRange, SpecialSize
Travel	Subject
VHS	Subject
Video	Subject
VideoGames	Subject, BrandName, PriceRange, SpecialSize
Wireless	Subject
WirelessAccessories	Subject

Results Refined by Repeating ItemSearch Requests

One value of using search bins is that you can divide items into groups according to criteria without having to parse item attributes. Based on the search bins returned, you can then submit a second request using the `ItemSearch` parameter value that helps target your results, but how?

The names of bins and the parameters that describe the bins vary according to the bin. The following sample shows a bin based on price.

```
<SearchBinSetNarrowBy="PriceRange">
<Bin>
  <BinName>$0$24</BinName>
  <BinItemCount>1645</BinItemCount>
  <BinParameter>
    <Name>MinimumPrice</Name>
    <Value>9</Value>
  </BinParameter>
  <BinParameter>
    <Name>MaximumPrice</Name>
    <Value>2499</Value>
  </BinParameter>
</Bin>
```

The response shows the minimum and maximum price for items in the bin, $0$24, and the number of items in it, 1645.

Other NarrowBy values name bins differently.

As you can see from these examples, BinParameter names are the same as `ItemSearch` input parameter names. This correspondence means that you can create a second `ItemSearch` request using the search bin results as values for `ItemSearch` parameter values. For example, *`MinimumPrice`* and *`MaximumPrice`* are returned in search bins based on *`PriceRange`*. You could take the values of the search bin and put them directly into `ItemSearch` parameters. Using the preceding PriceRange example, you could write the following `ItemSearch` request to retrieve items only in the first search bin.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
SearchIndex=Baby&
Keywords=pants&
Availability=Available&
Condition=All&
MinimumPrice=0& MaximumPrice=2499&
ResponseGroup=SearchBins
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

`ItemSearch` divides the results of this request into another set of search bins because the SearchBins response group was used again. This means that the price range of the first search bin in the first response is split into multiple search bins in the response to the second request. The second response enables you to present more granularity in price ranges. For example, from the first response, you could return all items that cost between $0 and $24.99. In the second response, you are able to provide a much smaller price interval, for example, $10 to $14.99.

To refine search results using SearchBin

1. Include the SearchBins response group in an ItemSearch request.

   The response includes the narrowby value and results grouped by bins.

2. Use related ItemSearch parameters to send a second request to return results related to only those items within the specified bin.

   For example, if the *`NarrowBy`* value is *`PriceRange`*, the related `ItemSearch` parameters are *`MinimumPrice`* and *`MaximumPrice`*.

3. If you want to refine the results further, use the SearchBins response group in a second request and repeat the procedure.

The process of using search bin results for `ItemSearch` parameter values can be iterative. You can, for example, submit a third request using the SearchBins response group to divide one search bin into more search bins. This process can be repeated until the level of granularity you desire is reached. At that point, you can send a last request using other response groups of your choosing.

Alternatively, you could refine the search results in a different way. Some search indices return more than one set of search bins. In those cases, you can use the values from more than one set of search bins in an `ItemSearch` request. Using the preceding example, if the response also included a search bin based on BrandName, which is the NarrowBy value, you could use brand and price range values in an `ItemSearch` request.

```
Brand=Levi's&
MinimumPrice=0&
MaximumPrice=2499&
```

The response would then only include shirts by Levi's that cost under $25. You could continue to drill down by adding additional parameters to the request.

ItemSearch Request Tips

Here are some tips to help you create accurate ItemSearch (p. 207) requests.

- If you want to find only items sold by Amazon, set the MerchantId parameter to "Amazon". (Note: Amazon sells only new items).
- The *`Keywords`* parameter searches for word matches in an item's title and description. If you know a word is part of the title of an item, use the Title parameter because, in this case, it often returns fewer but more accurate results than the Keywords parameter.
- To use Boolean values, such as AND, NOT, or OR, in an `ItemSearch` request, use the Power parameter. You can create relatively sophisticated search criteria using this parameter. For more information, see ItemSearch (p. 207).

Finding Items Using Browse Nodes

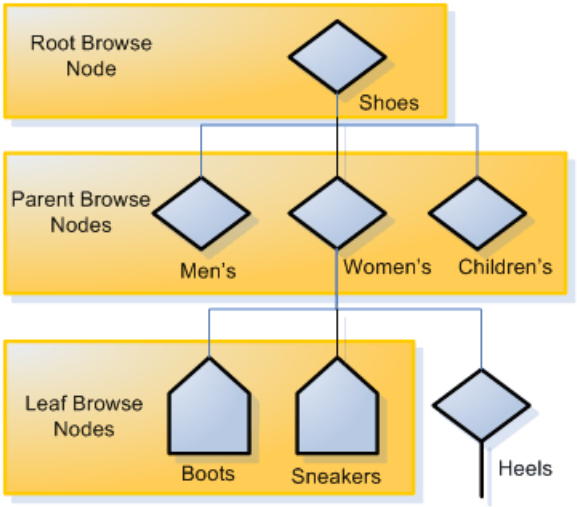
Topics

Browse nodes (p. 32) form an organizational hierarchy in which Amazon places items for sale. As the name suggests, the hierarchy of nodes is designed to ease and speed up the task of browsing through all of the items for sale. Zero or more items for sale are associated with each browse node.

Each browse node has an ID, which is a positive integer, and a name. All items associated with a node have something to do with the name of the node. For example, a node whose name is "Books about Ancient Greece," would be books whose subject is ancient Greece.

This hierarchy of nodes is dynamic as are the items associated with each node., that is, items can be associated and disassociated with browse nodes at any time, and nodes themselves can either be added to, subtracted from, or moved within the node hierarchy.

The value of the hierarchy is that it groups similar items and relates items in an intuitive way. As you progress down the hierarchy, you progress from parent to child nodes where child nodes are subsets of the parent node's product category, as shown in the following figure.



As you can see in this figure, as you go down the hierarchy, you refine then number of items that can be returned. In this very small, representative sample of the browse node hierarchy, the Shoes category has as a subset, Women's Shoes. It has two child nodes, Boots and Sneakers.

ItemSearch (p. 207) returns all of the items associated with a node and all of the nodes below it. For that reason, you can see that a search at the level of Women's Shoes will return all of the items related to Women's Boots and Women's Sneakers. If the customer was really only interested in women's boots, the response would contain a lot of uninteresting results—all of the items that are women's sneakers. A more targeted search would have specified the Boots browse node using `ItemSearch`'s BrowseNode parameter. For example, if the browse node ID of Women's Boots is 123456, a targeted request would be a follows.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Keywords=tall&
SearchIndex=Apparel&
BrowseNode=123456
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Traversing the Browse Node Hierarchy

As we have seen, you traverse down the hierarchy to make your search more targeted. You might also traverse up the tree to find, for example, the root category of a product. For example, if you have an item, such as a carving knife, you might find the node it is associated with and go up the hierarchy to find the root product category of knives and then, in another request, use the TopSellers response group to return the top sellers in the product category. These top sellers might be of interest to your customer.

BrowseNodeLookup (p. 177) operation is the most direct way of traversing the browse node hierarchy. When you supply it a browse node ID, it returns the name of the browse node as well as its direct descendants and a lineage of ancestors. Look at the response to the following request.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[Access Key ID&
Operation=BrowseNodeLookup&
BrowseNodeId=11232&
ResponseGroup=BrowseNodeInfo
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

A snippet of its response follows.

```
<Item>
<ASIN>0131856340</ASIN>
<BrowseNodes>
  <BrowseNode>
  <BrowseNodeId>11232</BrowseNodeId>
  <Name> Social Sciences</Name>
  <Ancestors>
    <BrowseNode>
      <BrowseNodeId>53</BrowseNodeId>
      <Name>Nonfiction</Name>
      <Ancestors>
        <BrowseNode>
          <BrowseNodeId>1000</BrowseNodeId>
          <Name>Subjects</Name>
          <Ancestors>
            <BrowseNode>
              <BrowseNodeId>283155</BrowseNodeId>
              <Name>Books</Name>
            </BrowseNode>
          </Ancestors>
        </BrowseNode>
      </Ancestors>
    </BrowseNode>
  </Ancestors>
  <Children>
    <BrowseNode>
      <BrowseNodeId>11233</BrowseNodeId>
      <Name>Anthropology</Name>
    </BrowseNode>
    <BrowseNode>
      <BrowseNodeId>11242</BrowseNodeId>
      <Name>Archaeology</Name>
    </BrowseNode>
    <BrowseNode>
      <BrowseNodeId>3048861</BrowseNodeId>
      <Name>Children's Studies</Name>
    </BrowseNode>
  </Children>
</BrowseNodes>
```

Traversing down the tree is easy enough: select the browse node ID that is relevant and repeat the `BrowseNodeLookup` operation until you descend to the appropriate browse node.

Traversing up the tree is similar. Although the BrowseNodeLookup (p. 177) operation, by default, returns the complete ancestral lineage of the subject browse node, there is a limitation: the request only returns one parent browse node for each node. That is, if a node has more than one parent, the response will

only show one of the parent nodes. Which one it returns is arbitrary. For that reason, you might want to traverse up the browse node hierarchy one level at a time.

Related Topics

- BrowseNodes Response Group (p. 230)

Finding a Node to Start Your Search

Traversing up or down the browse node hierarchy is easy enough. The question is, where do you start? In the US locale alone, there are over 120,000 nodes and those nodes are changing constantly. Fortunately, there are multiple ways to find the browse node where you can start your search:

- Many high level browse nodes are listed in the Browse Nodes Values section of the Product Advertising API Reference.
- The BrowseNodes response group returns browse nodes. You can use this response group with the following operations: ItemSearch (p. 207), and SimilarityLookup (p. 218).
- On www.amazon.com, search for an item that is similar to the one you want and then copy the browse node out of the URL.

As an example, let's see how you might find listings for a horse bridle. None of the top product categories (search indices) relate directly to horses. So, let's use `ItemSearch` and the BrowseNodes response groups to find a browse node that is associated with horse bridles.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&amp;
AWSAccessKeyId=[AWS Access Key ID]&amp;
Operation=ItemSearch&amp;
Keywords=horse,bridle&amp;
SearchIndex=PetSupplies,SportingGoods
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

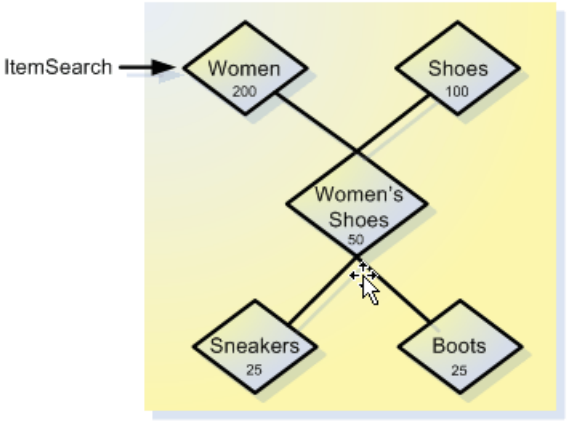
This request looks in the PetSupplies and SportingGoods search indices for anything related to horses or bridles. The following XML is a snippet of the response.

Related Topics

- BrowseNodes Response Group (p. 230)

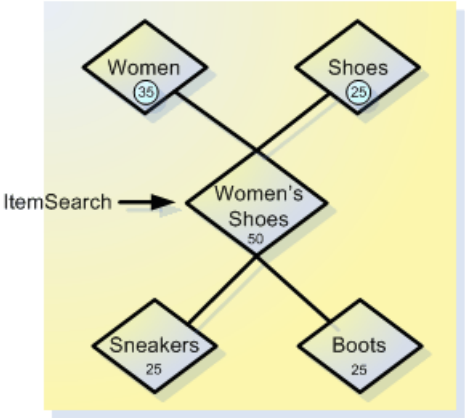
Understanding BrowseNode Results When Drilling Down

BrowseNodes are related in a hierarchy where one BrowseNode can have zero or more ancestor and child BrowseNodes, as shown in the following figure.



This diagram shows five BrowseNodes and their hierarchy. Node A might be, for example, Shoes. Node B might be Women. Node C, a child of BrowseNodes A and B, might be Women's Shoes.

The number inside each node represents the number of items in the BrowseNode when `ItemSearch` is called on Node A. BrowseNode C is a subset of BrowseNode A and a subset of BrowseNode B. Perhaps fifty of the items in BrowseNode C are also in BrowseNode A. When `ItemSearch` was run on BrowseNode A, the operation returned that BrowseNode A has 200 items. When `ItemSearch` is repeated using a BinParameter Name value that was returned in the first `ItemSearch` request, it is the same as running `ItemSearch` on a child of BrowseNode A, which, in this case, is BrowseNode C. By narrowing the search to only those values found in BrowseNode C, `ItemSearch` returns only those items in BrowseNode A that are also in BrowseNode C. As shown in the following figure, the number of common items is 35.



As a result, `ItemSearch` returns that BrowseNode A now only contains 35 items. While the change in the item count in BrowseNode A might, at first, be confusing, understanding that `ItemSearch` is returning only the intersection of BrowseNodes A and C explains why the number of items in BrowseNode A changes. Notice that the item counts in other BrowseNodes can also change.

Because the number of items in surrounding BrowseNodes can change with each new drill down using `ItemSearch`, the BrowseNodes that contain the greatest number of items can change dramatically. In the previous discussion, for example, the number of items in BrowseNode A changed from 200 to 35 on successive `ItemSearch` requests. Because `ItemSearch` returns only the top ten BrowseNodes that contain the most items, the identity of the top ten BrowseNodes can change with each `ItemSearch`.

Motivating Customers to Buy

Topics

Motivating customers to buy products through your web site is vital to the health of your business. Many Product Advertising API operations and response groups provide ways to pique a customer's interest and help drive sales. All of the sections in this chapter explain ways to enhance the selling potential of your web site.

Images

Topics

An image is said to be worth a thousand words. In e-commerce, this is almost always true. The downside of e-commerce is that customers cannot try on clothes or handle items they might like to buy. Providing customers with great images helps overcome that hurdle.

Images Supplied by the Product Advertising API

Amazon catalogs millions of product images. These images reside on a series of servers dedicated to serving images, as you can see from the URL of one image:

```
http://images.amazon.com/images/P/B000BNM5OA.01_SL110_.jpg
```

All image URLs have a common structure.

URL Portion	Description
Endpoint	URL of the image server. For all images it's *http://images.amazon.com/images/P/*.

URL Portion	Description
Alphanumeric token	An ID that uniquely identifies an image, for example, B000BNM5OA.01.SWCH. The 01 is a country code. Country codes are important because the language in the figures varies by locale. The following list shows the country code values. • 01 - US, CA • 02 - UK • 03 - DE • 08 - FR • 09 - JP
Suffix	Specifies the size of the image returned. Possible suffix values are: • Large size : _SL500_.jpg • Medium size: ._SL160_.jpg • Small size: ._SL110_.jpg • Thumbnail size: ._SL110_.jpg • Tiny: ._SL75_.jpg • Very small (swatch) size: _SL30_.jpg

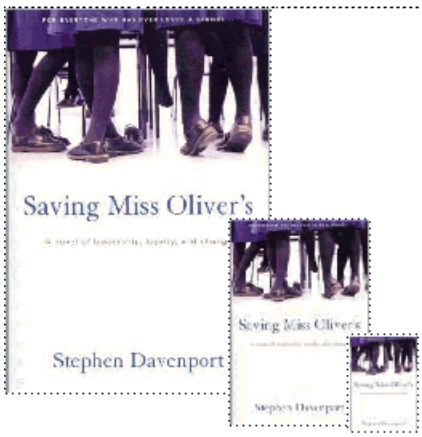
Note

The small and thumbnail sizes are the same.

Looking at the suffixes, you can see that the numerical values refer to the maximum number of pixels on the longest side of the image. Generally, Amazon images will be exactly the number of pixels specified in the filename, that is, a _SL500_.jpg image will be 500 pixels long on its longest side. Images that do not come from Amazon will not necessarily be 500 pixels exactly. The image returned will be the largest available image that is equal to or less than the image size specified in the image's filename. For example, an _SL500_.jpg image might only be 400 pixels on its longest side.

There are 96 pixels per inch.

The following figure shows the relative sizes of Large, Medium, and Small images.



Notice that all of the images are the same, just in different sizes. This corresponds to the image ID being the same for all of the preceding images, just the suffixes, which indicate size, are different. Each image is about half the size of the next larger size.

Hard coding image URLs into applications is not recommended because images come and go. Using the Images or VariationImages response groups to return images guarantees up-to-date image URLs. Once you retrieve the image URLs, you can manipulate just the suffixes of the image name to display the different sizes of the image. This technique saves you from having to send separate requests for each image size.

Related Topics

- VariationImages Response Group (p. 297)
- Images Response Group (p. 242)

Image Details

The Product Advertising API returns images in two ways:

- Under <Item> in the <SmallImage> , <MediumImage> , and <LargeImage> elements.
- Under <ImageSets> where the image sizes are specified by the _SLXXX_ modifier, where XXX is the number of pixels on the longest side of the image. A medium size image, for example, has 160 pixels on its longest side.

The following response snippet shows the first way in which the image is returned.

```
<Item>
  <ASIN>B000Q6780O</ASIN>
    <SmallImage>
     <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL75_.jpg</URL>

      <Height Units="pixels">75</Height>
      <Width Units="pixels">58</Width>
    </SmallImage>
    <MediumImage>
     <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL160_.jpg</URL>

      <Height Units="pixels">160</Height>
      <Width Units="pixels">124</Width>
```

```
    </MediumImage>
    <LargeImage>
      <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L.jpg</URL>
      <Height Units="pixels">500</Height>
      <Width Units="pixels">389</Width>
    </LargeImage>
```

Notice that each image is associated with an item identifier, which, in the preceding example, is an ASIN. The image details include the following:

- Image's size, which is captured in the elements <SmallImage> , <MediumImage> , and <LargeImage>
- URL of the image on Amazon's image server
- Image's height and width measurements in pixels to aid in displaying the images

  The standard resolution is 24 bit color and 96 pixels per inch

The following response snippet shows the second way in which the image is returned.

```
<ImageSets>
  <ImageSet Category="primary">
    <SwatchImage>
     <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL30_.jpg</URL>

      <Height Units="pixels">30</Height>
      <Width Units="pixels">23</Width>
    </SwatchImage>
    <SmallImage>
     <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL75_.jpg</URL>

      <Height Units="pixels">75</Height>
      <Width Units="pixels">58</Width>
    </SmallImage>
    <ThumbnailImage>
     <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL75_.jpg</URL>

      <Height Units="pixels">75</Height>
      <Width Units="pixels">58</Width>
    </ThumbnailImage>
    <TinyImage>
     <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL110_.jpg</URL>

      <Height Units="pixels">110</Height>
      <Width Units="pixels">86</Width>
    </TinyImage>
    <MediumImage>
     <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL160_.jpg</URL>

      <Height Units="pixels">160</Height>
      <Width Units="pixels">124</Width>
    </MediumImage>
    <LargeImage>
      <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L.jpg</URL>
      <Height Units="pixels">500</Height>
      <Width Units="pixels">389</Width>
    </LargeImage>
```

```
 </ImageSet>
</ImageSets>
```

The sizes of the images under <ImageSets> are specified by the _SLXXX_ suffix in the URL, where XXX is the number of pixels on the longest side of the image. A medium size image, for example, has 160 pixels on its longest side so it has the suffix _SL160_. This is the preferred way to reference images.

The ImageSets element attribute, Category, is set to Primary. Primary images are the same images that appear in the <Item> section.

Multiple Images Versus Variation Images

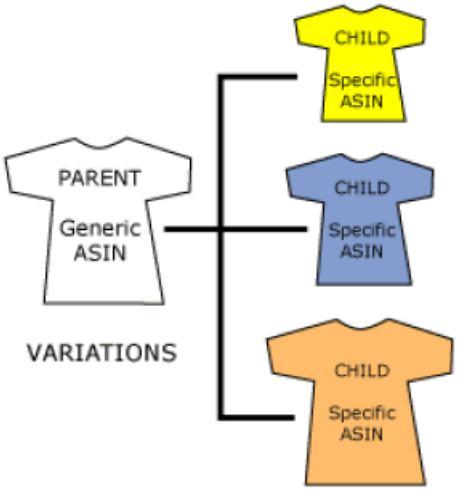
Amazon images are associated in the following ways:

- Size
- View
- Variation

Product Advertising API makes every image available in three or four different sizes. One item is presented in multiple views and each of the views is presented in three or four different sizes. The following images show the same item in two different views. Both images are medium sized.

One item has one or more variations. The images returned show the same item in its multiple variations and each variation is presented in three or four different sizes. The following images show two variations of the same item.

Amazon groups items as variations of one another when the items are identical except for one or two properties. In the preceding figure, the same shirt is shown in two different colors. Each variation shown is called a child item. The parent item is what associates all of the child items and their images, as shown in the following figure.



In this example, the parent item comes in three different colors. Each color is a separate variation, and each variation has its own item identifier, such as a ASIN. Typically, the parent item does not have an image associated with it because it is an abstraction of all of its child items. If a parent ASIN is included in a request, Product Advertising API typically tries to return the image of a child ASIN.

In the XML responses, the image URLs are grouped according to these associations within the ImageSet element.

Image Sets

An image set groups together related images. Those images can be related by size, item, or variation. The following image set shows images grouped according to size.

```
<ImageSets>
 <ImageSet Category="primary">
  <SmallImage>
   ...
  </SmallImage>
  <MediumImage>
   ...
  </MediumImage>
  <LargeImage>
   ...
  </LargeImage>
 </ImageSet>
</ImageSets>
```

If the images were variations of the primary image, the image set would include a fourth image, the swatch size image.

```
<SwatchImage>
 ...
</SwatchImage>
```

If multiple views of the same item were returned, you would have multiple *`ImageSet`* elements within an *`ImageSets`* element, where each *`ImageSet`* represented a different view, as follows.

```
<ImageSets>
  <ImageSet>
    ...
  </ImageSet>
  <ImageSet>
    ...
  </ImageSet>
<ImageSets>
```

In an *ImageSet*, then, you have one view of an image presented in three or four sizes. So, if Amazon has five different images of an item, Product Advertising API would return three or four sizes of each image, totaling fifteen or twenty image URLs.

Typically, XML responses include multiple ImageSets, which are returned within the *ImageSets* element.

ImageSet Categories

In your XML response, you might get as many as thirty image URLs. Which one or ones should you display? Image name suffixes specify the size of the image. Choosing a size narrows the possible images to display. Still, you must choose which image set to use.

Product Advertising API takes the guess work out of such situations by adding an attribute to the *ImageSet* element called Category. The following XML response snippet shows an example of a "primary" Category ImageSet.

```
<ImageSets>
  <ImageSet Category="primary">
```

The following table shows *Category* values and their potential uses.

Category Value	Description
primary	Use the images in this image set for the first image a customer sees of an item.
variant	Use images in this image set to display other views of the same item or to present variations of the primary item.
swatch	Use images in this image set to get four sizes of the swatch image.

The following image shows an example of an image from the primary *ImageSet*.

The following image shows the variant image associated with the primary from the variant *ImageSet*.

This image is the back cover of the same book. Displaying multiple views of the same item helps give customers a better sense of what they are buying. For that reason, providing customers with the opportunity to display "variant" image sets in addition to the "primary" image set helps drive sales.

Retrieving Images

Images are returned by two response groups:

- Images
- VariationImages

These response groups are also included in the Medium and Large response groups.

The Images response group can be used with the following Product Advertising API operations:

- `ItemLookup`
- `ItemSearch`
- `SimilarityLookup`

The *`Images`* response group does not return images of child items; it only returns an image of the parent item, if there is one. To return the images of the child items, use the *`VariationImages`* response group, which can only be used with the `ItemLookup` operation.

Conversely, *`VariationImages`* only returns images of child items. If you use this response group with items that are not child items, `ItemLookup` does not return any images.

Some items do not have images or have images that are missing, for example, the large size of an image might be missing. If an item does not have an associated image, either the image is missing or the item is a parent item. Often parent items do not have associated images because they are abstractions of all of the child items. Product Advertising API sometimes returns the image of a child item for a parent item so that something can be displayed.

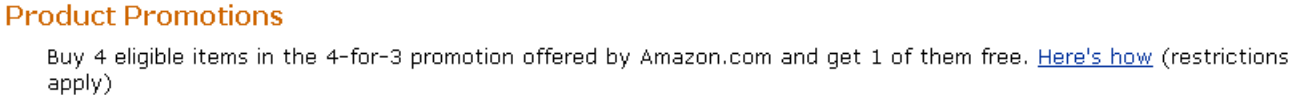
If the item in the request, such as a parent item, does not have images associated with it, the *`ImageSet`* and *`ImageSets`* elements are not included in the response.

Promotions

Topics

Giving customers the opportunity to buy items at discounted prices is a great way to drive sales. You can determine whether or not an item has an associated promotion by including the PromotionSummary response group in your request.

The *PromotionSummary* response group returns most if not all of the information you need to display promotional information on a web site, as shown in the following figure.



Promotions are only available with items that have offers. Items that do not have offers are, for example, Collection and Variation parent ASINs. For that reason, when you use either of the promotion response groups, you must also use, in the same request, a response group that returns an offer, including:

- *Large*
- *OfferFull*
- *Offers*

You can use the following operations to get promotion information because these operations can return offers:

- `ItemLookup`
- `ItemSearch`
- `SimilarityLookup`

Product Advertising API operations do not enable you to create promotions for items. The promotional response groups only enable you to return promotional information.

Related Topics

- PromotionSummary Response Group (p. 273)

Promotion Categories

There are many different kinds of promotions associated with items for sale. Product Advertising API fully supports the following promotion categories.

Category	Description
ForEachQuantityXGetQuantityFreeX	When you purchase a specified number of items you receive some number of the same items for free. For example, for every six dozen eggs you buy you get a dozen eggs free.
BuyAmountXGetSimpleShippingFreeX	The item is shipped free of charge.
BuyAmountXGetAmountOffX	For a specified dollar amount you receive a discount off another item. For example, buy three balls and get 25% off of a baseball glove.

These promotional categories are returned by the response element:

- Category, in *PromotionSummary* responses

Benefit and Eligible Items

Items returned by one of the promotions response groups are related to the promotion in one of the following ways:

- **Benefit—**The item is part of the promotional reward
  For example, in a buy one camera and get a camera case free promotion, the camera case is the benefit item
- **Eligible—**The item is what the customer must purchase to qualify for the promotion
  For example, in a buy one camera and get a camera case free promotion, the camera is the eligible item
- **Both—**It is possible for an item to be both the benefit and eligible item
  For example, in a buy two shirts get the third shirt half off promotion, the shirt is both the benefit and the eligible item

The promotions response groups specify whether an item is the benefit or eligible item by using the elements *`IsInBenefitSet`* and *`IsInEligibilitySet`*.

```
<IsInBenefitSet>true</IsInBenefitSet>
<IsInEligibilityRequirementSet>true</IsInEligibilityRequirementSet>
```

The value type for both of these elements is boolean. When "true," the item is part of that set. For example, if *`IsInBenefitSet`* is "true," the item is one of the benefits of the promotion.

If *`IsInBenefitSet`* is "true," the response also contains the element "BenefitDescription," which describes the benefit item, as shown.

```
<BenefitDescription>Save $25.00 when you spend $125.00 or more on Kitchen &
Housewares or Bed & Bath products offered by Amazon.com. Enter code AUGSAVER
at checkout.</BenefitDescription>
```

If IsInEligibilitySet is "true," the response will contain the element, "EligibilityDescription," which describes the eligible item.

Typical Response Elements Used

The promotion response groups provide a wealth of information about specific items. It is almost always a good idea to include one or both of the promotion response groups in requests that also ask for offers. Among all of the response elements returned by PromotionSummary, the following are typically used in the display of promotion information.

Response Element	Description
BenefitDescription	Describes the benefit, which is the item(s) that the customer receives as a result of the promotion. This element will not be present if the item is not part of the promotional benefits.
EligibilityDescription	Describes the items the customer must purchase to qualify for the promotion. This element will not be present if the item does not qualify the customer to receive the promotional benefit.
TermsAndConditions	Specifies the terms and conditions of the promotion.

The following response snippet shows the values for these elements.

```
<BenefitDescription>Save $25.00 when you spend $125.00 or more on Kitchen &
Housewares or Bed & Bath products offered by Amazon.com. Enter code AUGSAVER
at checkout.</BenefitDescription>
<TermsAndConditions><STRONG>To receive the Best Value discount:</STRONG> <OL>
<LI>Add $125 of qualifying Kitchen & Housewares or Bed & Bath products to your
 Shopping Cart via the <STRONG>Add to Shopping Cart</STRONG> button on each
respective product information page. …
</TermsAndConditions>
```

These three elements are meant to be read by people. For that reason, they are returned in HTML.

Response Element	Description
ComponentType	Specifies what the promotion applies to, for example, Shipping, ItemPrice, Subtotal.
CouponCombinationType	Specifies the kinds of promotional coupons that can be combined, for example, Unrestricted, Preferential, and Exclusive, where: • *Unrestricted*—Can be used with any other promotions • *Preferential*—Can be used with some other promotions • *Exclusive*—Cannot be used with any other promotions
FormattedPrice	The price formatted for display.
IsInBenefitSet	A boolean value. When True, the associated ASIN is what the customer receives as part of the promotion.
IsInEligibilityRequirementSet	A boolean value. When "true," the associated ASIN is what the customer must purchase to qualify for the promotion.

The following response snippet shows the values for these elements.

```
<ComponentType>ItemPrice</ComponentType>
<CouponCombinationType>Unrestricted</CouponCombinationType>
<IsInBenefitSet>true</IsInBenefitSet>
<IsInEligibilityRequirementSet>true</IsInEligibilityRequirementSet>
```

These values show that the promotion is based on discounting the price of the item ("ItemPrice"), that the promotion can be combined with any others ("Unrestricted"), and that the item is both what needs to be bought and what is awarded ("True").

For full responses, see the PromotionSummary response group.

Related Topics

- PromotionSummary Response Group (p. 273)

RSS Feeds for Promotions

Amazon sends RSS feeds that give you the latest promotions offered by Amazon. The RSS feeds are free and come in XML so the data can be customized for display on a web site. Use any RSS reader to receive the feeds.

The following example shows a snipped from an RSS feed.

```
<rss version="2.0">
  <channel>
  <title>Dollar Off Promotion</title>
  <link>http://www.amazon.com</link>
  <description>Dollar Off Promotion</description>
  <pubDate>Thu, 11 Mar 2010 08:01:13 GMT</pubDate>
  <dc:date>2011-09-24T08:01:13Z</dc:date>
  <image>
    <title>Amazon.com—Earth's Biggest Selection</title>
    <url>http://images.amazon.com/images/G/01/rcm/logo2.gif</url>
    <link>http://www.amazon.com</link>
  </image>
  <item>
  <title> Save $20 on Katadyn Water Filters</title>
    <link>
http://www.amazon.com/s/ref=nb_ss_sg/002-1272578-
9248007?url=node%3D3400371&field-keywords=soldbyamazon+katadyn&tag=rssfeeds-
20&campaign_id=212301
    </link>
    <description>
      Save $20 on Katadyn Water Filters *** Starting from Fri Sep 01 00:00:00
PDT 2006 to Sun Oct 01 23:59:59 PDT 2006 ***
    </description>
    <guid>
http://www.amazon.com/s/ref=nb_ss_sg/002-1272578-
9248007?url=node%3D3400371&field-keywords=soldbyamazon+katadyn&tag=rssfeeds-
20&campaign_id=212301
    </guid>
</item>
```

Reviews

Topics

One of the great benefits of shopping on Amazon is the wealth of reviews of both items and sellers.

All of these features, in Product Advertising API, are read-only, that is, it is possible to retrieve reviews but not possible to create new reviews or lists.

Getting Editorial Reviews

The `EditorialReviews` response group provides the merchant's description of an item. This information is less of a review and more of a product description. The content of the `EditorialReviews` response group can be displayed individually as part of an `ItemLookup`, `ItemSearch`, or `SimilarityLookup` request.

Product Description

From the Manufacturer
This award-winning ride-on rocket features retro styling and Radio Flyer dependability. The classic red steel body is safe and durable. Kids love the astronaut and space sounds, vibrating motor action, clicking nose cone, and light up panel and after burner. Comfortable no-slip seat and covered front wheels make this car safe and sturdy.

For books, Editorial Reviews is a separate listing on the DetailPageURL web page. For example,

Editorial Reviews

The Midwest Book Review
SAVING MISS OLIVER'S is an engaging novel and is very highly recommended to all general fiction readers.

Bookwire, April 07, 2006
----Rich characters, unexpecdted plot development,and underlying themes of hope, lofty ideals, educational excellence, and historical pride.

Stephen Waters, Former School Head
---couldn't put it down. It was like a wire was attached to my brain, my memories, my very soul.

The following XML snippet shows part of the equivalent Product Advertising API EditorialReview response.

```
<EditorialReview>
  <Source>Bookwire, April 07, 2006</Source>
  <Content>——Rich characters, unexpected plot development,and underlying themes
 of hope, lofty ideals, educational excellence, and historical pride.</Content>
</EditorialReview>
```

The Editorial Reviews section also contains the author's description of the book.

Getting Customer Reviews

The `Reviews` response group provides an iframe URL that contains customer reviews for a specified item.

Important

The content of the `Reviews` response group must be periodically updated as it expires in 24 hours.

The reviews are returned in the "Customer Reviews" section on the DetailPageURL web page, as shown in the following figure.

The following XML snippet shows part of the equivalent Product Advertising API Customer Reviews response returned by an `ItemLookup` or `ItemSearch` request.

```
<ItemLookupResponse>
 <Items>
  <Item>
   <ASIN>0316067938</ASIN>
   <CustomerReviews>
    <IFrameURL>
    http://www.amazon.com/reviews/iframe?akid=[AWS Access Key
ID]&asin=0316067938&exp=2010-09-
02T17%3A54%3A07Z&linkCode=xm2&summary=0&tag=ws&truncate=256&v=2&sig=[Signature]

    </IFrameURL>
   </CustomerReviews>
  </Item>
 </Items>
</ItemLookupResponse>
```

After you receive the `IFrameURL` in the Customer Reviews response, you can embed it into an HTML page. For example:

```
<iframe src="http://www.amazon.com/reviews/iframe?akid=[AWS Access Key
ID]&asin=0316067938&exp=2010-09-
02T17%3A54%3A07Z&linkCode=xm2&summary=0&tag=ws&truncate=256&v=2&sig=[Signature]"
 />
```

Important

Each iframe URL is valid for 24 hours. If the iframe URL expires, you will receive a `403 Forbidden` error code.

Suggesting Similar Items to Buy

Topics

When a customer purchases or searches for an item, they demonstrate an interest. As a developer, you can capitalize on that fact by suggesting to the customer similar items they might also like to buy. Add-on purchases is a powerful means of quickly increasing sales.

Similarity is based on items that customers bought, that is, customers who bought X also bought Y. This algorithm is different from one based on items viewed, for example. Basing the algorithm on purchases rather than viewing assures you that the similar items are more likely to be of interest to the customer.

Product Advertising API provides the following means of returning a list of similar items:

- Similarities response group
- CartSimilarities response group
- SimilarityLookup operation

Typically, the `Similarities` response group is used to find item IDs that are similar to the items returned in a response. To return extended information about those items, you could use the similar item IDs in `ItemLookup` requests.

Alternately, if you know the ID of the item for which you want similar items, you can use the `SimilarityLookup` operation to return extended information about each similar item.

Finding Similar Item IDs

The `Similarities` response group returns the title and ID of items that are similar to those returned in a response. For example, in the following sample request.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId= B000184IY0&
ResponseGroup=Similarities
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

the following similar item was returned:

```
<SimilarProduct>
  <ASIN>B00004GJVO</ASIN>
  <Title>Minor Move</Title>
</SimilarProduct>
```

You can use the information in this response to display the titles of similar items. If you want to display more information about the similar items, use the ASIN returned in a new `ItemLookup` request and specify an appropriate response group, such as ItemAttributes.

Returning Items Similar to Cart Items

Customers demonstrate their interest in an item by putting it in their shopping cart. Showing customers similar items to what they already have in their cart is a good way to spur add-on sales.

The CartSimilarities response group, which works with most cart operations, returns items that are similar to the ones included in the cart request. Items are returned based on:

- **Similarity—**Items are similar to the one purchased
- **Items viewed—**Customers who viewed the item in the cart also viewed the items returned by this response group
- **Similar items—**Similar items in other product categories

Each of these similarity types have their own element tag: <SimilarProduct>, <SimilarViewedProduct>, and <OtherCategoriesSimilarProduct>, respectively, as shown.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
Operation=CartCreate&
Item.1.ASIN=B000062TU1&
Item.1.Quantity=2&
ResponseGroup=CartSimilarities
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request produces the following response snippet.

```
<SimilarProducts>
  <SimilarProduct>
    <ASIN>B00008DDXC</ASIN>
    <Title>Harry Potter and the Chamber of Secrets (Widescreen Edition)(Harry
Potter 2)</Title>
  </SimilarProduct>
</SimilarProducts>
<SimilarViewedProducts>
  <SimilarViewedProduct>
    <ASIN>B000E6UZZK</ASIN>
    <Title>Harry Potter Years 1-4 (Harry Potter and the Sorcerer's Stone /Chamber
 of Secrets / Prisoner of Azkaban / Goblet of Fire) (Widescreen Edition)
    </Title>
  </SimilarViewedProduct>
</SimilarViewedProducts>
<OtherCategoriesSimilarProducts>
  <OtherCategoriesSimilarProduct>
    <ASIN>0590353403</ASIN>
    <Title>Harry Potter and the Sorcerer's Stone (Book 1)</Title>
  </OtherCategoriesSimilarProduct>
</OtherCategoriesSimilarProducts>
```

This response shows that the *Harry Potter and the Sorcerer's Stone* DVD placed in the shopping cart is similar to another Harry Potter DVD, *Harry Potter and the Chamber of Secrets*. Customers who viewed the DVD in the shopping cart also viewed the DVD, *Harry Potter Years*.. A related item to the DVD in the cart but in a different product category is the book version of the DVD: *Harry Potter and the Sorcerer's Stone*.

Related Topics

- CartSimilarities Response Group (p. 237)

Using SimilarityLookup

When you put an item ID in a `SimilarityLookup` operation requires, the response includes similar items. In previous sections, you saw that you can use the `Similarities` response group to return similar items. So, why use `SimilarityLookup`? This operation gives you the following advantages:

- `SimilarityLookup` provides a number of request parameters, such as *`Condition`*, to help filter the response
- `SimilarityLookup` is able to use many response groups to customize the information in the response
- You can specify more than one item in a request and find items that are similar to all of them, or similar to each item in the request

The following request returns items similar to the specified ASIN.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=SimilarityLookup&
ItemId=[ASIN]
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The following XML is a snippet of the full response.

```
<Item>
  <ASIN>B0009VX8VI</ASIN>
  <ItemAttributes>
  <ProductGroup>Apparel</ProductGroup>
  <Title>Mark VII Men's Short Sleeve Knit Golf Shirt</Title>
</ItemAttributes>
</Item>
```

This response shows that one item, Mark VII Men's Short Sleeve Knit Golf Shirt, was found that is similar to the ASIN specified in the request.

Specifying Multiple Items

The `SimilarityLookup` operation enables you to specify multiple items in a request, for example:

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=SimilarityLookup&
```

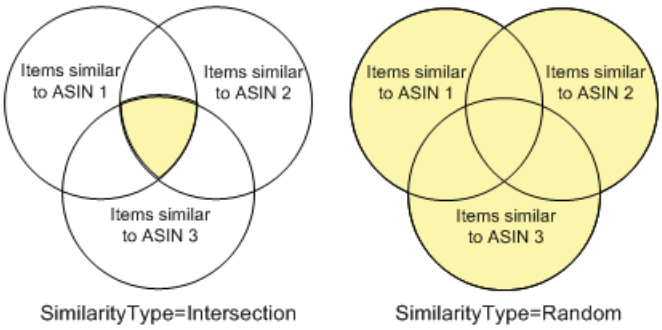
```
ItemId=ASIN1,ASIN2,ASIN3
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The response contains either items that are:

- Similar to all of the items specified in the request (intersection)
- Similar to one or more of the items specified in the request (union)

You specify the intersection or union of similar ASINs by setting the value of the *`SimilarityType`* parameter in a `SimilarityLookup` request. The possible parameter values are:

- Intersection, which means that the items returned are similar to all of the items specified in the `SimilarityLookup` request. This value is the default.
- Random, which means that the items returned are similar to at least one of the items specified in the `SimilarityLookup` request.



The second value is called Random because the similar items returned are a random pick of all similar items found. The operation can only return up to ten similar items. So, it is possible, if there are more than ten similar items, that identical `SimilarityLookup` requests can return different lists of similar items. It is also possible for the operation to return no similar items. This response is returned as an error message, for example:

```
<Error>
  <Code>AWS.ECommerceService.NoSimilarities</Code>
  <Message>There are no similar items for this ASIN: B000B776KY.
  </Message>
</Error>
```

An empty result is even more likely when the *`SimilarityType`* value is "Intersection."

Filtering Similar Items

The `SimilarityLookup` operation enables you to restrict the items returned by MerchantId. You can specify that you want to find similar items sold by only Amazon by setting the optional MerchantId parameter to "Amazon". This parameter is optional in `SimilarityLookup` requests.

Similar Versions of the Same Item

Some items come in a variety of media formats. For example, a book might come in hardback, paperback, audio cassette, audio CD, and DVD. A person buying an item in one format might be interested in the

same item presented in a different format. To return all of the media formats of a single item, Product Advertising API provides the AlternateVersions response group. The catch is, the response group only works with items in the Books or ForeignBooks product categories. That means, for `ItemSearch` and `ItemLookup`, the search index must be Books or, in non-US locales, ForeignBooks, and, for `ItemLookup`, the specified item must be a book or foreign book.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
Operation=ItemSearch&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
SearchIndex=Books&
Keywords=potter&
ResponseGroup=AlternateVersions&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request returns books and media in other formats that contain "potter" in their title or product description, as shown in the following snippet of the response.

```
<AlternateVersions>
  <AlternateVersion>
    <ASIN>030728364X</ASIN>
    <Title>Harry Potter and the Half-Blood Prince (Book 6)
    </Title>
    <Binding>Audio Cassette</Binding>
  </AlternateVersion>
  <AlternateVersion>
    <ASIN>0307283658</ASIN>
    <Title>Harry Potter and the Half-Blood Prince (Book 6)
    </Title>
    <Binding>Audio CD</Binding>
  </AlternateVersion>
</AlternateVersions>
```

This response shows that the book, *Harry Potter and the Half-Blood Prince*, is available on audio cassette and CD.

Related Items

You can return a list of related items using the RelatedItems response group in an ItemLookup request. When you do, you are required to use, in the same request, the RelationshipType parameter, which specifies the relationship between the related items, as shown in the following section. If you have more than ten related items, you use the RelatedItemPage parameter to return a set of ten related items. A value of 2, for example, returns the second set of ten related items.

Relationship Types

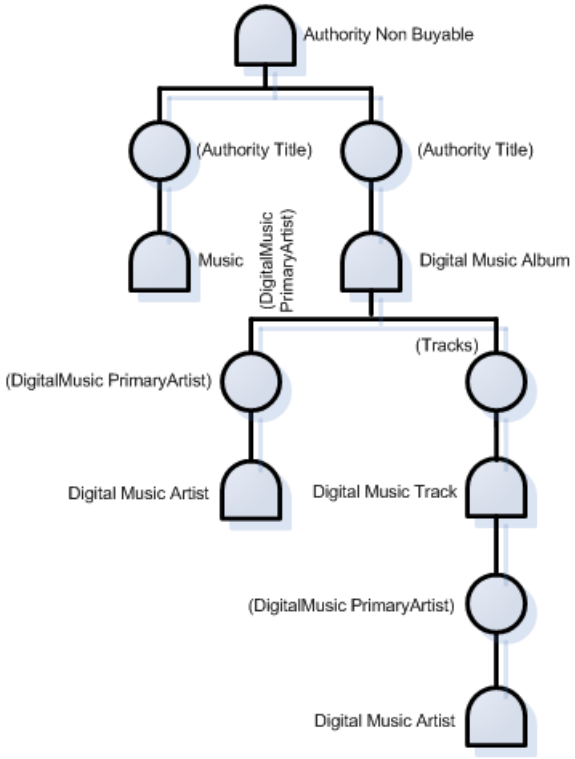
When an `ItemLookup` request uses the *`RelatedItems`* response group, the *`RelationshipType`* parameter is required. This parameter specifies the means by which items are related to the one specified in the `ItemLookup` request (by the *`ItemId`* parameter). The following list specifies all of the valid values for *`RelationshipType`*.

RelationshipType	Description
AuthorityTitle	Links a non-buyable ASIN TitleAuthority parent with its buyable children. A book might have a single TitleAuthority ASIN that relates to a list of children ASINs for different formats of the same book (hardback, paperback, audio book, kindle). MP3 albums have the same AuthorityTitle parent as its physical CD counterpart.
DigitalMusicArranger	Non-buyable child of both MP3 albums and tracks.
DigitalMusicComposer	Non-buyable child of both MP3 albums and tracks.
DigitalMusicConductor	Non-buyable child of both MP3 albums and tracks.
DigitalMusicEnsemble	Non-buyable child of both MP3 albums and tracks.
DigitalMusicLyricist	Non-buyable child of both MP3 albums and tracks.
DigitalMusicPerformer	Non-buyable child of both MP3 albums and tracks.
DigitalMusicPrimaryArtist	Non-buyable child of both MP3 albums and tracks. This is the relation ship that show all MP3 downloads for a single artist on amazon.com.
DigitalMusicProducer	Non-buyable child of both MP3 albums and tracks.
DigitalMusicRemixer	Non-buyable child of both MP3 albums and tracks.
DigitalMusicSongWriter	Non-buyable child of both MP3 albums and tracks.
Episode	Relates an Unbox Season (parent) to Episodes (children) from that season. Note this is the same relationship as Tracks and can be used interchangeably.
Season	Relates an Unbox Series (parent) to its Seasons (children).
Tracks	Relates an MP3 Album (parent) to its Tracks (children). Note this is the same relationship as Episode and can be used interchangeably.

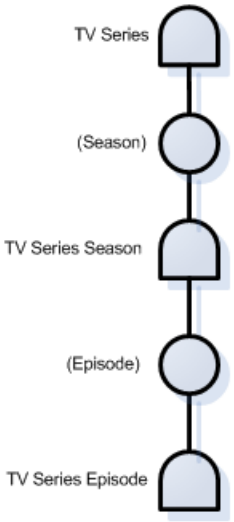
Hierarchy of Relationship Types

The RelationshipTypes values are hierarchically arranged. The following set of images show the hierarchies for MP3Downloads, UnboxVideo, and KindleStore items.In these images, the circles represent relationship types and the bell shaped figure represent items.

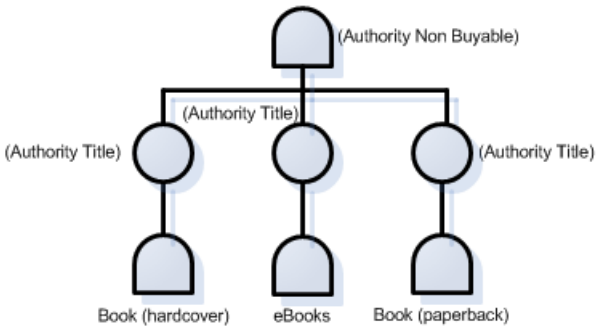
The following illustration shows the MP3Downloads hierarchy.



The following illustration shows the UnboxVideo hierarchy.



The following illustration shows the KindleStore hierarchy.



AuthorityTitle is similar to a variation parent in that it is an organizational tool (a container) rather than a real item that you can purchase. They do not have detail pages.

Recommending Items To Others

The Small response group, and its parent response groups, return a Tell Your Friend link that looks similar to the following:

```
http://www.amazon.com/gp/tf/-/B00H5S3Sbs/ref=tf_tell_friend_item/103-3N9W634-7F2?ie=UTF8&tag=[Associate%20ID]&creative=[Creative%20ID]&linkCode=xm2&camp=2025&creativeASIN=B00H5S
```

This link brings up an email so the customer can easily email a friend about the item.

Top Sellers

Topics

- Generalizing the Top Seller (p. 132)
- Getting Top Sellers From the Shopping Cart (p. 133)

Top selling items, by definition, are enjoyed by many people. Items labeled top sellers often spark a customer's interest.

You can use the `BrowseNodeLookup` operation with the `TopSellers` response group to return the top sellers in a browse node. For example, the following request returns the top sellers in browse node 20.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=20&
ResponseGroup=TopSellers
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The following response snippet includes the name and ASIN of the two top selling items in browse node 20.

```
<TopSeller>
  <ASIN>0446578622</ASIN>
  <Title>The Notebook Girls</Title>
</TopSeller>
```

```
<TopSeller>
  <ASIN>1400062586</ASIN>
  <Title>You're Wearing That? : Understanding Mothers and Daughters in
Conversation</Title>
</TopSeller>
```

You might wonder where you can get the browse node ID of an item so that you can find the top sellers in the browse node. The easiest way to return browse node IDs is to use the `BrowseNodes` response group with `ItemLookup`, `ItemSearch`, or `SimilarityLookup`, as shown.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B00008OE6I&
ResponseGroup=Browsenodes
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The following is a small portion of the full response.

```
<Item>
  <ASIN>B000002ADT</ASIN>n
  <BrowseNodes>
    <BrowseNode>
      <BrowseNodeId>63926</BrowseNodeId>
      <Name>General</Name>
```

This response provides the browse node ID and name with which the ASIN, B000002ADT, is associated.

Alternately, you can look in the Product Advertising API Developer Guide for a list of the top level browse node IDs.

Generalizing the Top Seller

Sometimes you might like to return the best sellers in the root browse node. For example, if a customer is buying a camera, you might assume that the person is interested other electronic items. In this case, you use the `BrowseNodeLookup` operation. Its default response group is `BrowseNodes`. For example, the following request searches for DVD comedies.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=163357
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The response, part of which follows, includes the browse node IDs of the specified browse node's parent and child browse nodes.

```
<BrowseNode>
  <BrowseNodeId>163357</BrowseNodeId>
  <Name>Comedy</Name>
```

```
  <Children>
    <BrowseNode>
      <BrowseNodeId>599826</BrowseNodeId>
      <Name>Boxed Sets</Name>
    </BrowseNode>
  ...
<Item>
  <Ancestors>
  <BrowseNode>
    <BrowseNodeId>549726</BrowseNodeId>
    <Name>Performing Arts</Name>
    <Ancestors>
      <BrowseNode>
        <BrowseNodeId>5</BrowseNodeId>
    <Name>Entertainment</Name>
    <Ancestors>
      <BrowseNode>
        <BrowseNodeId>2000</BrowseNodeId>
        <Name>Subjects</Name>
        <Ancestors>
          <BrowseNode>
            <BrowseNodeId>1000</BrowseNodeId>
            <Name>Books</Name>
```

This response shows one of the browse node's children. More important to this discussion, however, is the browse node ancestry that is returned. The further down the response that you read, the higher up you go in the browse node hierarchy. The eldest ancestor of the browse node ID, 163357, which is in the request is browse node ID 1000, Books.

Now that you have the browse node ID of the root product category, you can use the `BrowseNodeLookup` operation again, this time with the `TopSellers` response group to return the top sellers in the root browse node category.

Note

`BrowseNodeLookup` only returns one ancestor of a browse node even if a node has multiple ancestors. Which ancestor is returned is not predictable. This limitation of the operation typically is of little consequence in this use case. Any root browse node ID returned by the operation is always relevant to the browse node ID in the request.

Getting Top Sellers From the Shopping Cart

It is assumed that any item that a customer adds to their shopping cart is of interest to them. For that reason, Product Advertising API provides the CartTopSellers response group that returns the ASINs and titles of the top five, best sellers in the root category of the item specified in the cart operation. For example, when adding a television to a cart, the five top sellers in the root category, electronics, are returned, for example, the top selling computers, MP3 players, or digital cameras.

The CartTopSellers response group can be used with most cart operations, as shown.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
Operation=CartCreate&
Item.1.ASIN=B000062TU1&
```

```
Item.1.Quantity=2&
ResponseGroup=CartTopSellers
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The following is a snippet of the response:

```
<TopSellers>
  <TopSeller>
    <ASIN>B00005JOFQ</ASIN>
    <Title>Brokeback Mountain (Widescreen Edition)</Title>
  </TopSeller>
  <TopSeller>
    <ASIN>B000E6EK3S</ASIN>
    <Title>Harry Potter and the Goblet of Fire (Widescreen Two-Disc Deluxe
Edition)(Harry Potter 4)</Title>
  </TopSeller>
</TopSellers>
```

This response shows that there are two top sellers associated with the root product category of the item that was added to the shopping cart.

New Releases

Whether it's the release of a new book or a new DVD, new releases are always popular with customers. Product Advertising API enables you to take advantage of that motivator with two response groups, NewReleases and CartNewReleases.

The NewReleases response group returns the ASIN and title of newly released items in a specified browse node. This response group works only in `BrowseNodeLookup` requests, as shown.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=4229&
ResponseGroup=NewReleases
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The response includes the new releases associated with browse node 4229.

```
<NewReleases>
  <NewRelease>
    <ASIN>0446578622</ASIN>
    <Title>The Notebook Girls</Title>
  </NewRelease>
</NewReleases>
```

To find the browse node ID of an item, use the BrowseNodes response group.

Returning New Releases From Shopping Cart Items

A great way to spur add-on sales is by presenting customers with newly released items that are similar to items in their shopping cart. The CartNewReleases response group enables you to do this. This response group works with most cart operations.

The CartNewReleases response group returns the ASINs and titles of the top five new releases in the root category of the item specified in the cart operation. For example, when adding a television to a cart the top five new releases in the root category, electronics, are returned.

In the following request, an item is added twice to a newly-created shopping cart.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
Operation=CartCreate&
Item.1.ASIN=B000062TU1&
Item.1.Quantity=2&
ResponseGroup=CartNewReleases
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The response includes a new release from the root product category.

```
<NewReleases>
  <NewRelease>
    <ASIN>B00005JOFQ</ASIN>
    <Title>Brokeback Mountain (Widescreen Edition)</Title>
  </NewRelease>
</NewReleases>
```

Sorting by Popularity, Price, or Condition

The order in which you present the items returned is important to your customers. Often, they would like to know things, such as, what item is the least expensive, which is the most popular, which items are collectibles? Product Advertising API provides this functionality using the Sort parameter in `ItemSearch` requests.

The valid values for the Sort parameter in `ItemSearch` requests vary by locale and search index. For a complete list of valid values, see ItemSearch Sort Values By Locale. (p. 325)

Some of the most common `ItemSearch` Sort parameter values are described in the following table.

Sort Parameter	Description
pricerank, price, +price	Order items according to price from cheapest to the most expensive. The three versions of this value are equivalent in meaning but are valid in different search indices.
inversepricerank, -price	Order items according to price from the most expensive to the cheapest. The two versions of this value are equivalent in meaning but are valid in different search indices.

Sort Parameter	Description
salesrank	Order items according to how well they've sold, from best to worst sellers.
relevancerank	Order items according to how often the keyword appears in the product description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Order items according to how highly rated the item was reviewed by customers where the highest ranked items are listed first and the lowest ranked items are listed last.

For example, the following request returns a list of toy rockets sorted from the cheapest to the most expensive.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Keywords=Rocket&
SearchIndex=Toys&
Sort=price&
ResponseGroup=Offers&
ItemPage=10
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The following is a snippet of the response.

```
<Item>
  <ASIN>B000BOWQWA</ASIN>
  <OfferSummary>
    <LowestNewPrice>
      <Amount>210</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$2.10</FormattedPrice>
    </LowestNewPrice>
    <TotalNew>3</TotalNew>
    <TotalUsed>0</TotalUsed>
    <TotalCollectible>0</TotalCollectible>
    <TotalRefurbished>0</TotalRefurbished>
  </OfferSummary>
</Item>
<Item>
  <ASIN>B0006N6MGW</ASIN>
  <OfferSummary>
    <LowestNewPrice>
      <Amount>211</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$2.11</FormattedPrice>
    </LowestNewPrice>
  <TotalNew>2</TotalNew>
```

```
  <TotalUsed>0</TotalUsed>
  <TotalCollectible>0</TotalCollectible>
  <TotalRefurbished>0</TotalRefurbished>
  </OfferSummary>
</Item>
```

This response lists two different items (toy rockets), which are ordered by price. The same ASIN can be offered for sale by multiple merchants and sellers. The `Offers` response group provides a summary of the lowest price per condition. For example, the first item, B000BOWQWA, listed in the response is being sold as "New" by three sellers (<TotalNew>3</TotalNew>). The lowest price of this item in New condition is $2.10. No one is selling this item in Used, Collectible, or Refurbished condition.

All other Sort parameter values work in a similar way.

Shipping Options

Shipping options are generally spelled out in child elements of the OfferListing element. One such child element is *`IsEligibleForSuperSaverShipping`*, a boolean value that specifies whether an item is eligible for super saver shipping. You cannot set this value; you can only use the value.

Returning Price and Availability Information

Topics

Previous chapters have shown how to find items and how to suggest additional items that a customer might find interesting. Returning price and availability information is important for the customer.

Returning price and availability information is explained in the following sections.

Returning Prices

Topics

Items for sale on Amazon can be sold by more than one seller or merchant. For example, the same camera might be sold by three different merchants or sellers. The following figure from Amazon's retail web site shows an example in which two merchants are selling the same book.

Notice that the merchants are selling the same book for different prices. In other examples, the same item might be offered in multiple conditions, such as New and Used.

Any item being sold is associated with an offer. An offer is a combination of price, condition, and vendor. For example, one offer might be Amazon selling the new book for $11.67. Therefore, to find an item's price, you return the offers made by the vendors selling the item.

Offer information is made available by the OfferSummary, Offers, and OfferFull response groups, which can be part of `ItemLookup`, `ItemSearch`, and `SimilarityLookup` requests.

Note

Some manufacturers have a minimum advertised price (MAP) that can be displayed on Amazon's retail web site. When performing an ItemSearch or ItemLookup operation in these cases, the string "Too Low to Display" is returned instead of the actual price. The only way to see the actual price is to add the item to a remote shopping cart and follow the `PurchaseURL`.

Offer Summaries

The OfferSummary response group returns summary information about offers, including:

- Total number of offers per condition
- Lowest price per condition

For example, the following response snippet shows the lowest prices for an item in New and Used condition.

```
<OfferSummary>
  <LowestNewPrice>
    <Amount>801</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$8.01</FormattedPrice>
  </LowestNewPrice>
  <LowestUsedPrice>
    <Amount>799</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$7.99</FormattedPrice>
  </LowestUsedPrice>
  <TotalNew>45</TotalNew>
  <TotalUsed>20</TotalUsed>
  <TotalCollectible>0</TotalCollectible>
  <TotalRefurbished>0</TotalRefurbished>
</OfferSummary>
```

Returning All Offers

Amazon returns offers for new and used items. The Small response group, and all of its parent response groups, return a link that displays all new and used offers for the item in the response. The link is similar to the following.

```
http://www.amazon.com/gp/offer-
listing/B000IHL51S%3FSubscriptionId%3D1C36YNE9WHA6T34ADP
EXAMPLECode%3Dxm2%26camp%3D2025%26creative%3D386001%26creativeASIN%3DB000IHL51S
```

Returning More Information About the Offer

Perhaps you want to display more than just the cheapest offer of an item in each condition. In this case, you would use the Offers response group.

The following response snippet shows the data included for a single offer.

```
<Offer>
  <OfferAttributes>
    <Condition>Used</Condition>
  </OfferAttributes>
  <OfferListing>

<OfferListingId>[illegible]

    </OfferListingId>
    <Price>
      <Amount>21995</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$219.95</FormattedPrice>
    </Price>
    <Availability>Usually ships in 1 2 business days
    </Availability>
    <IsEligibleForSuperSaverShipping>0
    </IsEligibleForSuperSaverShipping>
  </OfferListing>
</Offer>
```

If you want more information about the vendor than is included in the Offers response group, use the OfferFull response group. That response group adds to the Offers response the:

- Merchant's name
- Merchant's nickname

Providing Price Ranges for Product Categories

You can use the SearchBins response group to return the price ranges of items for sale in a specified product category.. The following response to an ItemSearch (p. 207) request shows how the SearchBins response group can provide price ranges.

```
<SearchBinSets>
  <SearchBinSet NarrowBy="PriceRange">
  <Bin>
```

```
    <BinName>$25-$49</BinName>
    <BinItemCount>316</BinItemCount>
    <BinParameter>
      <Name>MinimumPrice</Name>
      <Value>2500</Value>
    </BinParameter>
    <BinParameter>
      <Name>MaximumPrice</Name>
      <Value>4999</Value>
    </BinParameter>
  </Bin>
</SearchBinSet>
<Bin>
  <BinName>$0-$24</BinName>
  <BinItemCount>280</BinItemCount>
  <BinParameter>
    <Name>MinimumPrice</Name>
    <Value>0</Value>
  </BinParameter>
  <BinParameter>
    <Name>MaximumPrice</Name>
    <Value>2499</Value>
  </BinParameter>
</Bin>
</SearchBinSet>
```

This response shows that there are 316 items in this product category that cost between $25 and $49 and 280 items that cost between $0 and $24.

Knowing what price ranges are available for a product category enables you to use the *`MaximumPrice`* and *`MinimumPrice`* parameters in a second `ItemSearch` request to retrieve items only in the specified price range. Using one of the offer response groups in the same request enables you to display the prices of the items in a specified price range.

Items That Do Not Have Offers

There are two kinds of items returned by Product Advertising API:

- Regular items
- Parent items

A parent item is an abstraction of a collection of items. For example, shirt is a parent item. A shirt that is a specific size and color would be a child of the parent item, which is called a regular item. Each regular item has an item identifier, such as an ASIN. That means that a red, large shirt would have a different ASIN from the same shirt that is blue and size large. The child items are also called variations.

Because a parent item is an abstraction, it cannot be sold so it does not have an *OfferListingId*.

Determining Parent Items

How do you determine if the item returned in a response is a parent item? If you use either the Variations or VariationSummary response groups in a request, the responses will include a VariationSummary tag if the item is a parent item.

Secondly, if you use one of the offer response groups, parent items do not have offers. So, the summary would be as follows.

```
<OfferSummary>
  <LowestNewPrice>
    <Amount>0</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$0.00</FormattedPrice>
  </LowestNewPrice>
  <TotalNew>0</TotalNew> <TotalUsed>0</TotalUsed>
<TotalCollectible>0</TotalCollectible> <TotalRefurbished>0</TotalRefurbished>
</OfferSummary>
```

Returning Offers From Item Variations

The Book, Music, Video, and DVD search indexes have very few parent items. The Apparel, Jewelry, and SportingGoods search indexes have many parent items and variations.

If a parent item is returned in a response, it will not have an offer. Instead, you must find the offers associated with the variations of the parent item. You do this by adding the Variations response group to the request, as follows.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=1MEXAMPLEZBG2&
Operation=ItemSearch&
Keywords=Ralph%20Lauren&
SearchIndex=Apparel&
Sort=pricerank&
ResponseGroup=Offers,Variations&
Availability=Available&
Condition=All
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The Variations and Offers response groups returns the first ten variations of the parent item and their offers, which is shown in the following response snippet.

```
<Variations>
  <TotalVariations>6</TotalVariations>
  <TotalVariationPages>1</TotalVariationPages>
  <Item>
  <ASIN>B000FG8I6W</ASIN>
  <ItemAttributes>
    <Binding>Apparel</Binding>
    <Brand>Polo Ralph Lauren</Brand>
    <ClothingSize>10 - 13</ClothingSize>
    <Color>Beige Heather</Color>
    <Department>mens</Department>
    ...
  </ItemAttributes>
  <Offers>
    <Offer>
      <Merchant>
      <Name>Polo.com</Name>
      </Merchant>
      <OfferAttributes>
        <Condition>New</Condition>
```

```
    </OfferAttributes>
    <OfferListing>

<OfferListingId>[illegible]</OfferListingId>

    <Price>
      <Amount>1400</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$14.00</FormattedPrice>
    </Price>
    <SalePrice>
      <Amount>560</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$5.60</FormattedPrice>
    </SalePrice>
    <Availability>Usually ships in 1-2 business days</Availability>
  </OfferListing>
</Offer>
</Offers
```

Variation Dimensions

A variation is a child ASIN. The parent ASIN is an abstraction of the children items. For example, Shirt is a parent ASIN. Parent ASINs cannot be sold. A child ASIN of it would be a blue shirt, size 16, sold by MyApparelStore. This child ASIN is one of potentially many variations. The ways in which variations differ are called dimensions.

In the preceding example, size and color are the dimensions. Parent ASINs therefore return two related elements:

- VariationDimensions
- VariationDimension

The following response snippet shows these elements.

```
<VariationDimensions>
  <VariationDimension>ClothingSize</VariationDimension>
  <VariationDimension>Color</VariationDimension>
</VariationDimensions>
```

The values returned by these elements are the dimensions listed in the child ASIN's response, as shown.

```
<Item>
  ...
  <ItemAttributes>
    ...
    <ClothingSize>Large</ClothingSize>
    <Color>Black</Color>
    ...
  </ItemAttributes>
  ...
</Item>
```

Determining Availability

Topics

Only items that are available can be put into the active area of a remote shopping cart. Unavailable items can be put in the SaveForLater area of the cart.

Amazon defines available items as those that are:

- Currently for sale
- Pre-release orders, such as buying a Harry Potter book before it is released
- Special orders
- e-mail me when the items become available

The availability of an item can change without notice. Putting an item in a cart does not reserve it. Items in carts can become unavailable and their prices can change as soon as the item goes into the cart.

Determining if an item is available is determined differently for merchants and sellers.

Availability of Merchant Items

1	Check the Availability element value returned by the Offers or OfferFull response groups.
2	Make sure the item has an offer listing ID.
3	Use the Availability request parameter along with the Condition parameter in an `ItemSearch` request. Items might be available, for example, in one condition but not another.

With Merchants, you use one of the Offer response groups to determining item availability.

Availability Values

The Offer Summary Report lists the availability of an item using the Availability element. The value of the element indicates if the item can be purchased and how soon it will be shipped, as shown.

```
<Availability>Usually ships in 24 hours</Availability>
```

This response, which is returned by the Offers and OfferFull response groups, confirms that an item is available to buy.

The value returned by the Availability element may not match the one on the Amazon retail web site's product detail page because typically there is a short and long version of an availability message:. Product Advertising API returns the short version. The more verbose availability message is used on the retail web site.

For non-Amazon products in JP, FR, DE, and CA; the value returned by the Availability element is “1-2 business days,” or the localized equivalent, regardless of the actual availability.

The following table describes the possible Availability element values.

Message	Description
Usually ships in %X	A dynamic response where %X represents a variable amount of time.
Not yet released	The item is not available for purchase. The item may or may not have a projected release date. If there is a release date, it may show up in the ReleaseDate element of the item attributes.
Not yet published	The item is not available for purchase. The item may or may not have a projected release date. If there is a release date, it may show up in the ReleaseDate element of the item attributes.
This item is not stocked or has been discontinued.	The item is not available for purchase.
Out of Stock	The item is currently not available for purchase, but may be in the future.
Limited Availability	Used for items sold by third-parties if an item is out of stock, but may be available for purchase later.
Out of Print--Limited Availability	Customers can choose to be notified if a copy becomes available.
Special Order	Titles occasionally go out of print or publishers run out of stock. The buyer is notified if the item becomes unavailable."
This item is currently not available by this merchant	The message is sent primarily for Amazon offers that are missing an availability message.

These availability messages apply to Amazon only. Sellers might return a completely different set of availability messages.

Checking for an Offer Listing ID

An offer listing ID is an alphanumeric token that uniquely identifies an item that is sold by any merchant, including Amazon. Whereas an offer is a combination of Condition and Price, an offer listing ID is similar to a price tag, one is associated with each item for sale, as shown in the following figure.

This figure shows that three vendors are selling the same item, a shirt. The vendor, labeled Offer 1, has three shirts in stock and each has an OfferListingId.

If an item is for sale, it has an offer listing ID. This ID is returned by the Offers and OfferFull response groups, as shown in the following response snippet.

```
<OfferListing>
  <OfferListingId>[Offer Listing ID]</OfferListingId>
  <Price>
    <FormattedPrice>Too low to display</FormattedPrice>
  </Price>
  <Availability>Usually ships in 24 hours</Availability>
  <IsEligibleForSuperSaverShipping>0
  </IsEligibleForSuperSaverShipping>
</OfferListing>
```

Offers contain availability and shipping information.

Using the OfferSummary Response Group

The OfferSummary response group returns, in part, the total number of items available in each condition, for example:

```
<Item>
  <ASIN>B000BWFJQ2</ASIN>
  <OfferSummary>
    <LowestNewPrice>
      <Amount>295</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$2.95</FormattedPrice>
    </LowestNewPrice>
    <TotalNew>1</TotalNew> <TotalUsed>0</TotalUsed>
<TotalCollectible>0</TotalCollectible> <TotalRefurbished>0</TotalRefurbished>
  </OfferSummary>
```

This response shows that only one of the specified items, B000BWFJQ2, is available and it is only available in "New" condition.

Using the ItemSearch Availability Parameter

The `ItemSearch` *Availability* parameter enables you to search only for items that are available. The only valid value for Availability is "Available," as shown in the next example.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=All&
Availability=Available&
SearchIndex=Apparel&
Keywords=Shirt
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Items can become unavailable quickly. Even though this parameter filters out unavailable items, checking that an item has an *OfferListingId* is still recommended.

Working With Remote Shopping Carts

Topics

Most e-commerce web sites have adopted the metaphor of the shopping cart, which is a place to save and list the items a customer wants to buy. Just like a shopping cart in a grocery store, the e-commerce shopper has the ability to change the number and kinds of items the cart contains. They might add new items to a cart, increase or decrease the quantity of an item that is already in the cart, even empty the cart of all items. Instead of physically adding and removing items from a real cart, Product Advertising API operations provide similar functionality on a virtual shopping cart, called the Product Advertising API remote shopping cart..

Once the customer has found something they want to buy, they typically click an HTML button or link called, for example, **Add To Cart** . This action adds the item(s) to their existing shopping cart. If this is the first item they have chosen to place in a shopping cart, the shopping cart itself is created.

In Product Advertising API, the shopping cart is called remote because the cart is hosted by Amazon servers. In that sense, the shopping cart is remote to the seller's or Associate's servers where the customer is shopping..

The opposite of a remote shopping cart is a local shopping cart, which is the shopping cart customers use while shopping on www.amazon.com. It is considered local because Amazon hosts the shopping web pages as well as the shopping cart. Product Advertising API operations work solely with remote shopping carts.

The following sections describe remote shopping carts and how to work with them.

Shopping Cart Concepts

Topics

The following sections describe remote shopping carts in detail.

Cart Identifiers

When you create a remote shopping cart using `CartCreate`, Product Advertising API returns the new cart's ID (CartId). The cart ID is also an alphanumeric token that is used to identify a cart. The cart ID and HMAC values must be used in all Product Advertising API requests related to the cart (except `CartCreate`).

Cart Limitations

A cart can contain up to 50 entries. Each entry can represent one or more (up to 999) of the same items. That means that a Product Advertising API shopping cart can contain a maximum of almost 50,000 items.

Many products have a limit on the quantity that can be added to a cart. Sometimes this is inherent to the product, for example, one-of-a-kind jewelry items. Sometimes it is a limit set by the vendor, for example, they might set a limit on big ticket items, such as a motorcycle, of one per customer. If you add an item to the cart with a quantity that exceeds the maximum allowed, `CartAdd` automatically resets the quantity to the maximum value. No error is generated. Therefore, you must verify the quantity of items in the cart with the number ordered to determine if the quantity has been capped. If so, you must notify the customer.

Active and SaveForLater Areas

Shopping carts have two distinct areas:

- **Active—**Contains the items that are ready to be purchased.
- **SavedForLaterItem—**Contains items that a customer has chosen to buy but are currently unavailable, or items that a customer has designated they want to save and buy later.

Available items placed in a shopping cart are automatically added to the Active area unless the customer specifies otherwise. Unavailable items or items in the shopping cart that become unavailable are automatically put into the SaveForLater area. When those items become available Amazon automatically moves the items to the Active area of the shopping cart.

The SaveForLater area can also be used by customers as a holding place for items they are interested in but not ready to purchase. The prices and availability for items in this area are regularly updated by Amazon.

Items That Cannot Be Added to the Active Cart Area

Product Advertising API operations sometimes return items that cannot be added to the Active cart area. The following sections explain those cases:

- Out of stock items
- Items with limited quantities
- Variation parent items
- Collection parent items

Out of Stock Items

Most items for sale are available immediately. There are times, however, when that is not true. Items can go out of stock or very popular items, such as a new Harry Potter book, are pre-sold, that is, the book is sold before the book is even published.

When a customer adds an item to their cart that is not available, it is added to the SavedForLater area. Also, if an item in the cart, for some reason, becomes unavailable, Amazon automatically moves the item in the cart to the SavedForLater items section.

It is also possible for a customer to add an item directly to the SavedForLater items area in their cart so that they can easily purchase the item at a later date.

When items become available, you can use the `CartModify` operation with the Action parameter to move items from the SaveForLater section of the cart to the Active section, which is fully described in Moving Adding Items as Saved For Later (p. 155). Or, if Amazon automatically moved an item in the Active area to the SaveForLater area because the item went out of stock, Amazon will move it back into the Active area automatically when it becomes available.

Items With Limited Quantities

Many products have a limit on the quantity that can be added to a cart. Sometimes this is inherent to the product, for example, one-of-a-kind jewelry items. Sometimes it is a limit set by the vendor, for example, they might set a limit for big ticket items of one per customer. If you add an item to the cart with a quantity that exceeds the maximum, `CartAdd` or `CartCreate` automatically resets the quantity to the maximum. No error is generated. Therefore, you must verify the quantity of items in the cart with the number ordered to determine if the quantity has been capped. If so, you must notify the customer.

Variation Parents

Variation parent items cannot be added to a cart. For example, you cannot add Apparel to a cart because it is not clear what apparel you really want to buy. You could add one of its children, however, such as a red shirt, size large. If you try to add a parent item, you get an error message similar to the following.

```
The item you specified, [ASIN], is not eligible to be added to the cart. Check
 the item’s availability to make sure it is available.
```

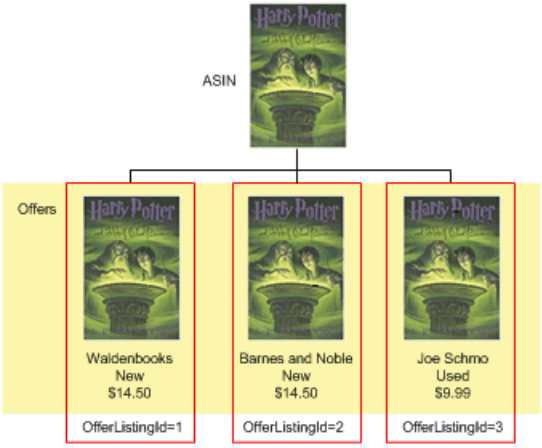
Variation parents do not have corresponding OfferListingId's.

Collection Parents

Collection parent items are abstractions of the items in the collection. The collection parent cannot be added to a cart. The Collection parent is used as a name holder for the collection. Collection parents do not have corresponding OfferListingId's.

Offers and Identifiers

ASINs identify items, such as a Harry Potter book. ASINs do not, however, identify sales information related to those items.



As you can see in the preceding figure, an ASIN identifies an item but not an instance of it that a customer can purchase. Every ASIN can be sold by many merchants. Each merchant lists the condition and price of the item. This combination of data represents an *OfferListingId*. Taken together, all *OfferListingIds* are referred to as offers. If an ASIN does not have an *OfferListingId*, the item cannot be purchased.

For this reason, it is recommended that you add items to a shopping cart by their *OfferListingId* rather than by their *ASIN*.

Remote Shopping Carts are Hosted by Amazon

Although a Product Advertising API application or web site runs on your servers, Product Advertising API remote shopping carts are hosted by Amazon servers. You use Product Advertising API operations to create and modify the cart and its contents. Keeping a local copy of a shopping cart is not recommended because the price and availability of items change often and Amazon automatically updates items in shopping carts. Also, Amazon automatically updates the status of lists, such as Wishlists, when a customer buys an item on a list. Keeping a local copy of a shopping cart runs the risk of cart items getting out of sync with their real price, availability, and status.

Because Product Advertising API shopping carts are hosted by Amazon instead of a local host, the shopping carts are called "remote." The cart used by a customer shopping on the Amazon retail web site, www.amazon.com in the US locale, is considered the "local" shopping cart. Amazon maintains for each customer ID only one local shopping cart. Developers can maintain more than one remote shopping cart for a customer but maintaining one cart per customer ID is recommended.

Cart Lifespan

Shopping carts have a lifespan. A remote shopping cart is created and, in time, it expires. In between, the contents of the cart can be modified and purchased, as shown in the following diagram.

Cart Creation

The first time a customer wants to add an item to their shopping cart, a remote shopping cart must be created, which is accomplished using the `CartCreate` operation. It is not possible to create an empty cart. At least, one item must be added.

Modifying a Cart

If a customer has an existing remote shopping cart, it should be used and modified accordingly. Modification can come in many forms:

- Items can be added to and deleted from the cart
- Items can be moved from the Active area of the cart to the SaveForLater area, or the reverse

The CartId and HMAC are used in the `CartModify`, `CartAdd`, or `CartClear` operations to modify the contents of the cart.

Cart Expiration

It is not possible to delete a remote shopping cart. Instead, it expires automatically after ninety days of disuse if there are items in the cart, or, if the cart is empty, after seven days of disuse. Carts are emptied either by the `CartClear` or `CartModify` operations, or automatically when the customer purchases the items in their cart. The expiration of the cart is reset when the cart is modified. For example, if, on day eighty-nine of disuse, a customer modifies the items in their shopping cart, the life time of the cart is reset to an additional ninety days. In that way, shopping carts can last indefinitely.

Once the items in a cart have been purchased using the *`PurchaseURL`* value, the cart is not deleted immediately but it should no longer be used. From a customer's point of view, once he or she has purchased the items in their cart and there are no items remaining in the SaveForLater section, the old cart is gone. The next time they want to purchase an item, they receive a new cart.

Remote Shopping Cart Tasks

Topics

Product Advertising API operations give you complete control of the items in a remote shopping cart. The tasks you can implement using Product Advertising API cart operations are described in the following sections.

Creating a Remote Shopping Cart

A remote shopping cart must be created the first time a customer decides to add an item to a shopping cart. Once the cart is created, it is reused and modified appropriately until it expires. Only one local cart can be created and only one cart per customer per vendor should be created.

Typically, an HTML button labeled, for example, Add to Cart, implements a `CartCreate` request, which includes:

- At least one item to add to the cart.
  You cannot create an empty cart.
- An *`AssociateTag`*.
  Including the *`AssociateTag`* value gives the Associate credit for the customer's purchase.

Example Creating a Cart

The following request creates a cart that will be merged with the customer's cart when the customer uses *`PurchaseURL`*.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]& AssociateTag=[ID]&
Operation=CartCreate&
Item.1.OfferListingId=B000062TU1&
Item.1.Quantity=2
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

In this example, two of the same item (Quantity=2) whose ASIN is B000062TU1 are added to the newly created (Operation=`CartCreate`) shopping cart. The Associate specified will receive a referral fee if the customer purchases the item.

In the lifespan of a shopping cart, `CartCreate` can be used only once. If you call `CartCreate` a second time, you create a new shopping cart. We recommend that you only create one cart per customer.

Specifying the Items in the CartCreate Request

In one `CartCreate` request, you can add up to ten different items to the cart. The quantity of each item can be between 1 and 50. That means in one `CartCreate` request, you could add up to five-hundred items (10 items * 50 quantity).

Items are specified in parameter pairs, as follows:

```
Item.N.ItemIdType=[Item identifier]
Item.N.Quantity=[Number of Item.N items]
```

N is a positive integer. The *N* value associates the ItemId with the quantity of those items that should be added to the cart, for example:

```
Item.1.OfferListingId=[An OfferListingId]&
Item.1.Quantity=3&
Item.2.OfferListingId=[An OfferListingId]&
Item.2.Quantity=5&
```

This request adds three number 1 items and five number 2 items to the remote shopping cart. You could interchange lines without affecting the end result, for example:

```
Item.1.OfferListingId=[An OfferListingId]&
Item.2.OfferListingId=[An OfferListingId]&
Item.1.Quantity=3&
Item.2.Quantity=5&
```

The values for *N* do not have to be listed in ascending or descending order, nor do the values of *N* need to be sequential. You could, for example, use the values, 1, 4, 7 for *N*. Also, the value of *N* is not retained between requests. For that reason, it is permissible to use Item.1 in a `CartCreate` request and then in a `CartAdd` request, which adds an item to the cart. In these requests, Item.1 can refer to two completely different items.

The value of adding multiple items to the cart in one request is performance. It takes less time to add ten items in one request than it does to add ten items to the cart in ten separate requests.

Using OfferListingId

You can specify items to add to a cart in the following ways.

Identifiers	Description
ASIN	An alphanumeric token that uniquely identifies an item sold by Amazon. Amazon assigns ASINs to items. Using OfferListingId is the preferred method of identifying items to add to the cart.
OfferListingId	An alphanumeric token that uniquely identifies an item sold by a seller or merchant. Amazon assigns OfferListingIds to items. You can find an OfferListingId using the `ItemLookup` and `ItemSearch` operations with Offers and OfferFull response groups. Using OfferListingId is the preferred method of identifying items to add to the cart.

The `CartCreate` example uses OfferListingId to identify the items to add to the cart. An OfferListingId is returned by the Offers and OfferFull response groups. An alternative is using an item's ASIN. An ASIN is assigned to every item offered by Amazon. An OfferListingId is assigned to every item that is available for sale.

The advantage of using OfferListingId in the `CartCreate` request is that items that cannot be purchased, such as Variation parent items, do not have OfferListingId's. By using the OfferListingId, you are assured that an item can be purchased.

AssociateTag Parameter

An *`AssociateTag`* is an alphanumeric token distributed by Amazon that uniquely identifies an Associate. It is the means by which Amazon credits an Associate for a sale. As we will see in a later section, the *`AssociateTag`* becomes part of the *`PurchaseURL`*, which is the URL used to purchase the items in a remote shopping cart. You must include an *`AssociateTag`* in a `CartCreate` request.

If a request does not contain an *`AssociateTag`*, Product Advertising API returns an error.

Be careful when you specify an *AssociateTag* in the `CartCreate` request. Errors are not returned for incorrect values.

Using Values Returned by CartCreate in Other Cart Operations

The following XML is a snippet of the response to the `CartCreate` request.

```
<CartId>002 2197248 2529608</CartId>
<HMAC>/WrekkZAPx782xttLFbZqviNUOA=</HMAC>
<URLEncodedHMAC>%2FWrekkZAPx782xttLFbZqviNUOA%3D</URLEncodedHMAC>
<PurchaseURL>https://www.amazon.com/gp/cart/aws merge.html?cart id=002 2197248
 2529608%26associate
id=ws%26hmac=/WrekkZAPx782xttLFbZqviNUOA=%26AWSAccessKeyId=[AWS Access Key
ID]</PurchaseURL>
<SubTotal>
  <Amount>2998</Amount>
  <CurrencyCode>USD</CurrencyCode>
  <FormattedPrice>$29.98</FormattedPrice>
</SubTotal>
<CartItems>
  <SubTotal>
    <Amount>2998</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$29.98</FormattedPrice>
  </SubTotal>
  <CartItem>
    <CartItemId>UV3W10T4V7PCZ</CartItemId>
    <ASIN>B000062TU1</ASIN>
    <Quantity>2</Quantity>
    <Title>Harry Potter and the Sorcerer's Stone (Full Screen Edition) (Harry
Potter 1)</Title>
  <ProductGroup>DVD</ProductGroup>
  <Price>
    <Amount>1499</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$14.99</FormattedPrice>
  </Price>
  <ItemTotal>
    <Amount>2998</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$29.98</FormattedPrice>
  </ItemTotal>
</CartItem>
</CartItems>
```

Many of the item attributes in the response are used in future cart requests. Those item attributes are explained in the following table.

Attribute	Description
CartId	An alphanumeric token that uniquely identifies a remote shopping cart. This value must accompany every cart operation associated with the newly created cart.
HMAC	Hash Message Authentication Code. This is an encrypted alphanumeric token used to authenticate requests. A URL-encoded version of this value must accompany every cart operation associated with the newly created cart. The alternative is to include, instead, the URLEncodedHMAC value.

Attribute	Description

URLEncodedHMAC	This is the *HMAC* value with all of the characters converted in to a URL-compliant form. Some *HMAC* characters, such as plus (+), are incompatible with URLs. This is a problem because the *HMAC* must be included in every cart operation and thus is part of the URL request. The *URLEncodedHMAC* value, then, is a convenience function that relieves developers of the need to create their own URL-encoded *HMAC* value. This value must accompany every cart operation.
	`<HMAC>Cw1g4IbVzOtzFkJR/zBj1GNnZMA</HMAC>`
	`<URLEncodedHMAC>Cw1g4IbVzOtzFkJR/zBj1GNnZMA`
	`</URLEncodedHMAC>`

PurchaseURL	This is the URL that is submitted, like a request, to enter the Order Pipeline and purchase the items in a cart. *PurchaseURL* includes the Associate's Tag. It is important that this URL is used to make the purchase otherwise the Associate will not get credit for the sale. `<PurchaseURL>https://www.amazon.com/gp/cart/aws-merge.html?cart-id=102-9464231-2184159%26associate-id=ws%26hmac=Cw1g4IbVzOtzFkJR/zBj1GNnZMA=%26AWSAccessKeyId=1VEXAMPLE9C02</PurchaseURL>`
CartItemId	An alphanumeric token that uniquely identifies an item in the cart. Although the items added to a remote shopping cart using `CartCreate` were identified by an *ASIN* or *OfferListingId*, those items lose those associations. Instead, to refer to an item in a cart in future cart operation requests, you must use the item's *CartItemId* value.

Modifying the Items in a Remote Shopping Cart

Product Advertising API operations give you a lot of flexibility to modify the number of items in a cart. For example, use:

- **CartAdd—**To add new items to a cart.
- **CartClear—**To remove all items from a cart.
- **CartModify—**To increase or decrease the number of items that are already in a cart, and to move items between the Active and the SaveForLater cart areas.
  You can use this operation to delete a single item from a cart by setting its quantity to zero.

All of these operations can be used only on an existing remote shopping cart.

Adding Items to a Cart

Often a customer, after creating a shopping cart, wants to keep shopping and add additional items to an existing shopping cart. You can facilitate this activity using the Product Advertising API operations `CartAdd` and `CartModify`.

If the item being added is already in the cart, you have to use the `CartModify` operation to change the quantity of the items already in the cart. You cannot use `CartAdd` to add items that are already in a cart. In the following example, the quantity of the specified item is changed to 10.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&amp;
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate ID]&
Operation=CartModify &
CartId=102-5929035-5792105&
HMAC=[HMAC]&
Item.1.CartItemId=[Cart Item ID]& Item.1.Quantity=10
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Notice that the item in the cart is referred to by its *CartItemId*. The *Quantity* value, 10, is the total number of those items that should be in the cart.

If the item being added to the cart is not already in the cart, you must use the `CartAdd` operation. `CartAdd` cannot increase the quantity of items that are already in the cart. If you try to do that, you get an error.

The following is an example of a request using `CartAdd`.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
CartId=102-5929035-5792105&
HMAC=[HMAC]=&
Operation=CartAdd &
Item.1.OfferListingId=1400042127& Item.1.Quantity=2
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request adds to the cart two (Quantity=2) of the same items, which are identified by the OfferListingId (1400042127).

As you can see, the method of identifying the item and its quantity to add to the cart is the same as it was for `CartCreate`. For more information, see CartCreate. (p. 187)

Adding Items as Saved For Later

To add an item to the SaveForLater area, use "SaveForLater" as the value for the *Action* parameter in a `CartModify` operation, for example:

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate ID]&
Operation=CartModify&
CartId=[Cart ID]&
HMAC=[HMAC]&
Item.1.CartItemId=[Cart Item ID]&
Item.1.Quantity=1
Item.1.Action=SaveForLater
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

In this request, the item identified by Item.1 is moved to the cart's SaveForLater area. The Quantity parameter enables you to move only some of one type of item into the SaveForLater area, for example,

if you had fifteen copies of the book, *Saving Miss Oliver's*, in the cart, you could move eight of them to the SaveForLater area by setting Item.1.Quantity=8 in the preceding request.

Notice that only `CartModify` can move an item between the Active and SaveForLater areas. This means that an item cannot be added directly to the SaveForLater area; it must first be added to the cart using `CartCreate` or `CartAdd` and then moved to SaveForLater. Amazon will automatically move an item to the SaveForLater area if the item is currently unavailable. `CartModify` will not be able to move that item into the Active area until it becomes available. At that time, Amazon will move the item automatically into the Active area.

Note

The value for the *`Action`* parameter is "SaveForLater." Items in the SaveForLater area are tagged in the XML response with the element, SavedForLaterItem, as shown in the following XML snippet from a response.

```
<SavedForLaterItem>
  <CartItemId>ULI7S9IYFJHX0</CartItemId>
  <ASIN>B0009GZV4A</ASIN>
  <Quantity>2</Quantity>
  <Title>Mark VII Men's Short Sleeve Golf Shirts with
   Tri -Colored Stripe Trim</Title>
  <ItemTotal>
    <Amount>1288</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$12.88</FormattedPrice>
  </ItemTotal>
</SavedForLaterItem>
```

To move an item from the SaveForLater area to the Active area, use " MoveToCart" as the value for the *`Action`* parameter in a `CartModify` operation:

```
Item.1.CartItemId=[Cart Item ID]&
Item.1.Quantity=1
Item.1.Action=MoveToCart
```

Removing Items From a Cart

A customer might, at times, decide to remove some or all of the items in their remote shopping cart.

To remove all items from a cart

- Use `CartClear` to remove all items from a cart.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
Operation=CartClear&
CartId=002-2041347-9034467&
HMAC=[HMAC]
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

If the customer, however, wants to remove only some of the items in their cart, use `CartModify` and set the *`Quantity`* and *`CartItemId`* parameters appropriately, for example:

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=CartModify&
CartId=[cart ID]&
Item.1.CartItemId=U1I8M9790QFDO7&
Item.1.Quantity=15&
Item.2.CartItemId=U3K5GRHEXU6FHK&
Item.2.Quantity=18
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request sets the quantity of the item specified by the *`CartItemId`* to 0 thereby removing it from the cart.

Retrieving the Contents of a Cart

You should not maintain a local copy of the remote shopping cart. Instead, use `CartGet` to retrieve the items in a shopping cart.

Retrieving the items in a cart

- Use the cart's *`HMAC`* and *`CartId`* values in a `CartGet` request.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
Operation=CartGet&
CartId=002-2041347-9034467&
HMAC=[HMAC]

&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

By default, `CartGet` uses the Cart response group. This response group provides a lot of information about items, including, for example, their price, quantity, and the seller ID of the seller selling the item, as shown in the following response snippet.

```
<PurchaseURL>https://www.amazon.com/gp/cart/aws-merge.html?cart-id=002-9918938-
1696046%26associate-
id=ws%26hmac=b0ogzvivVYLXjSZ9WwoBRFesFYU=%26AWSAccessKeyId=[AWS Access Key
ID]</PurchaseURL>
<CartItems>
  <SubTotal>
    <Amount>1994</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$19.94</FormattedPrice>
  </SubTotal>
```

```
  <CartItem>
    <CartItemId>U3KYV0C66V3PAA</CartItemId>
    <ASIN>B000062TU1</ASIN>
    <SellerNickname>Amazon.com, LLC</SellerNickname>
    <Quantity>2</Quantity>
    <Title>Harry Potter and the Sorcerer's Stone (Full Screen Edition)(Harry
Potter 1)</Title>
    <ProductGroup>DVD</ProductGroup>
    <Price>
      <Amount>997</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$9.97</FormattedPrice>
   </Price>
   <ItemTotal>
     <Amount>1994</Amount>
     <CurrencyCode>USD</CurrencyCode>
     <FormattedPrice>$19.94</FormattedPrice>
    </ItemTotal>
  </CartItem>
</CartItems>
```

You could, for example, use the *`CartItemId`* in a `CartModify` operation to change the quantity of that item. You could also use the *`PurchaseURL`* to purchase the items in the cart.

Add to Cart Form

The Add to Cart form enables you to add any number of items to a customer's shopping cart and send the customer to the Amazon retail web site.

Note

There is no direct way for a customer to get back to your site after getting sent to the Amazon site. To improve a customer's experience on your site, you may want to note this on your site or provide a way to get back to your site (using a new window, for example).

Although the parameters are optional, but you must specify at least one of the following parameters: ASIN or OfferListingId.

To add more than one item to the customer's cart

- Append each set of parameters with a period, then a unique identifier, which establishes a relationship between the parameters (for example, "ASIN.1=*`[ASIN]`*&Quantity.1=1&ASIN.2=[Another ASIN]&Quantity.2=12").

The Add to Cart form works with all locales. Send the Add to Cart form data to one of the following URLs:

- http://www.amazon.com/gp/aws/cart/add.html
- http://www.amazon.co.jp/gp/aws/cart/add.html
- http://www.amazon.co.uk/gp/aws/cart/add.html
- http://www.amazon.de/gp/aws/cart/add.html
- http://www.amazon.fr/gp/aws/cart/add.html
- http://www.amazon.ca/gp/aws/cart/add.html

The following table describes the input parameters for the Add to Cart form.

Parameter	Description	Required?
ASIN.x	Specifies one or more product ASINs to add, where x is a unique identifier.	Optional
OfferListingId.x	An alternative way to specify one or more product offer listings from third-party sellers, where x is a unique identifier.	Optional
AWSAccessKeyId	Your Access Key ID Here. You may want to specify this as a hidden parameter.	Required
AssociateTag	Your associate tag. You may want to specify this as a hidden parameter.	Required

Example HTML for the Add to Cart Form

```
<form method="GET" action="http://www.amazon.com/gp/aws/cart/add.html">
  <input type="hidden" name="AWSAccessKeyId" value="Access Key ID" /><br/>
  <input type="hidden" name="AssociateTag" value="Associate Tag" /><br/>
  <p>One Product<br/>
  ASIN:<input type="text" name="ASIN.1"/><br/>
  OfferListingId:<input type="text" name="OfferListingId.1"/><br/>
  <p>Another Product<br/>
  ASIN:<input type="text" name="ASIN.2"/><br/>
  OfferListingId:<input type="text" name="OfferListingId.2"/><br/>
  </p>
  <input type="submit" name="add" value="add" />
  </form>
```

Purchasing the Items in a Remote Shopping Cart

Topics

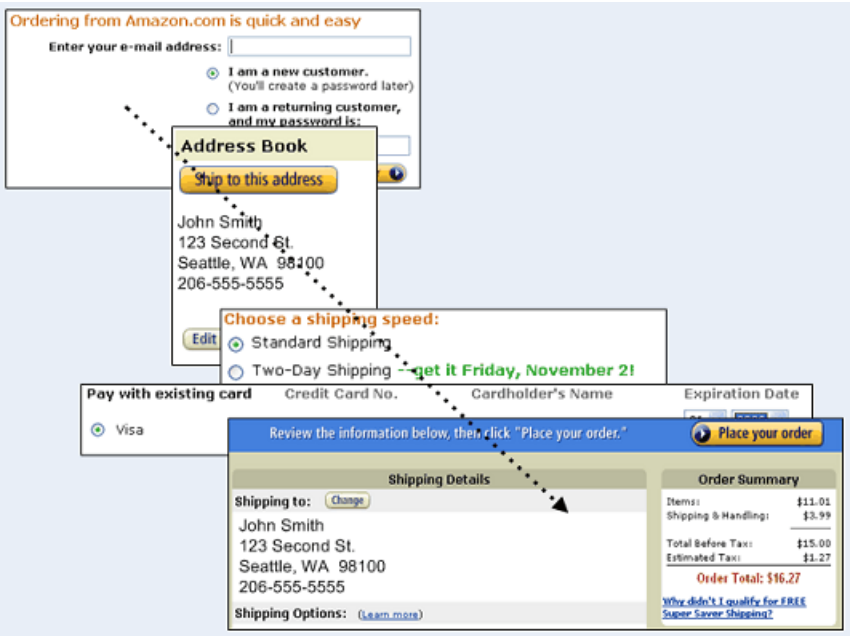
When a customer has finished shopping and wants to purchase the items in their remote shopping cart, they typically click an HTML button on a web site labeled, for example, **Proceed to Checkout**. This button must submit the value of the *`PurchaseURL`*, which is returned by all cart operations except `CartClear`. Once the customer clicks this button, the items in the Active area are emptied from the shopping cart and entered into the Order Pipeline.

This chapter explains how to submit the contents of a remote shopping cart for purchase in the following sections.

Order Pipeline

The Order Pipeline is a series of web pages hosted by Amazon that guides the customer through entering all of the information necessary to purchase the items. That information includes, for example, the

customer's name, shipping address, billing address, payment method, and a purchase confirmation web page, as shown in the following figure.



Implementing the Proceed to Checkout Button

All of the web pages in the Order Pipeline are hosted by Amazon so your application plays no role in purchasing the items or order fulfillment. Your application must, however, use the *`PurchaseURL`* returned by Product Advertising API cart operations to enter the shopping cart items into the Order Pipeline. The *`PurchaseURL`* is often implemented as an HTML form and button labeled, for example, **Proceed to Checkout**, as shown in the following figure.

PurchaseURL

All cart operations except `CartClear` return a value for *`PurchaseURL`*, for example:

```
<PurchaseURL>https://www.amazon.com/gp/cart/aws-merge.html?cart-id=002-9918938-
1696046%26associate-id=[Your
ID]%26hmac=b0ogzvivVYLXjSZ9WwoBRFesFYU=%26AWSAccessKeyId=[Access Key
ID]<PurchaseURL>
```

This value is a URL that should be submitted, like a request, to purchase the items in a remote shopping cart. The value contains the Associate's Tag if it was specified in the `CartCreate` operation. The Associate's Tag value in the *`PurchaseURL`* is what links the customer's purchase to the Associate. If the *`PurchaseURL`* is not used to buy the items in a cart, the Associate will not receive credit for the sale.

The information in bold includes:

- Cart identity (cart-id, hmac)
- Associate identity (associate-id)
- Request submitter (AWSAccessKeyId)

You can manually change any of the values in the *PurchaseURL* but that is not recommended.

The following is an HTML implementation of the *PurchaseURL*.

```
<form name="Proceed to Checkout" method="post"
action=https://www.amazon.com/gp/cart/aws-merge.html?cart-id=002-9918938-
1696046%26associate-id=[Your
ID]%26hmac=b0ogzvivVYLXjSZ9WwoBRFesFYU=%26AWSAccessKeyId=[Access Key ID]>
  <input type="submit" name="Proceed to Purchase"
  value="
          PurchaseURL
         Purchase" >
</form>
```

Buy Now Form

The Buy Now form enables the customer to enter the appropriate handling and shipping fee. So, be sure to explain this somewhere in your application.

Send the Buy Now form data to the following URL:

```
http://s1.amazon.com/exec/varzea/dt/cbop/order-checkout/
```

The following table describes the parameters in the Buy Now form.

Parameter	Description	Required?
purchase-navbar	An image that is added to the top of the purchase pages. This allows you to associate the look and feel of your site with the purchase pages on the Amazon Web site.	Required
purchase-store-name	The name of your site, which is listed under the navbar image.	Required
purchase-return-url	The link URL that is activated when a user clicks on the navbar image.	Required
purchase-exchange-id	The item sold by a third-party seller.	Required
purchase-quantity	The amount of a particular item to purchase.	Required
purchase-storefront-name	Name of the storefront that is selling the product	Required

Example HTML for the Buy Now Form

```
  <form action="http://s1.amazon.com/exec/varzea/dt/cbop/order-checkout/"
method="post">
  <input type="hidden" name="purchase-navbar"
value="https://www.mysite.com/navbar.gif"/>
  <input type="hidden" name="purchase-store-name" value="My Storefront Name"/>

  <input type="hidden" name="purchase-return-url" value="http://www.mysite.com"/>

  <input type="hidden" name="purchase-exchange-id" value="[An exchange ID]"/>
  <input type="text" name="purchase-quantity" value="1" size="2"/>
  <br><input type="hidden" name="purchase-storefront-name"
value="thestorefrontname"/>
  <input type="image"
    src="http://images.amazon.com/images/G/04/buttons/buy-now-from-seller-with-
payments.gif"
    width="170" height="29" border="0"/>
  </form>
```

Handling Errors and Troubleshooting Applications

Topics

Product Advertising API provides specific and descriptive errors to help you troubleshoot problems with your requests. There are two kinds of errors, as explained in the following sections.

Syntax and Parameter Errors

Topics

All responses contain an *IsValid* element, for example:

```
<IsValid>False</IsValid>
```

The *IsValid* element tells you whether or not there is an error in the syntax of any elements of the request and if all required parameters are included.. For example, if you were to omit the *Operation* parameter, which is required in every request, Product Advertising API would set *IsValid* to False and not process your request. Typically, these kinds of errors are a little more subtle. Typical errors are

incorrectly capitalized parameters or values, or the failure to include a required parameter in the request. For example, in the following request, the *SearchIndex* parameter is entered as "Searchindex."

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Keywords=Potter&
Searchindex=Books
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Product Advertising API returns the following error:

```
<Errors>
  <Error>
    <Code>AWS.MissingParameters</Code>
    <Message>Your request is missing required parameters. Required parameters
include SearchIndex.
    </Message>
  </Error>
</Errors>
```

Product Advertising API believes that the request is missing the *SearchIndex* parameter just because the "I" was not capitalized in the parameter name.

In the sample response, notice that the error consists of an error code that identifies the error, and an error message that describes the error. The error messages is in the language of the locale.

For a detailed list of error codes and messages, see Error Codes and Messages (p. 165) section of the API Reference.

Processing Error

Is it possible to submit a valid request and still have an error? Yes. If you were to submit a request and no items in Amazon satisfied the request, you would receive an error. The following request is an example of this problem.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=Refurbished&
ResponseGroup=Images&
SearchIndex=Automotive&
Title=Harry%20Potter
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The following response snippet shows the error.

```
<IsValid>True</IsValid>
...
<Errors>
  <Error>
```

```
    <Code>AWS.ECommerceService.NoExactMatches</Code>
    <Message>We did not find any matches for your request.
    </Message>
  </Error>
</Errors>
```

Looking for a refurbished edition of a Harry Potter book in the Automotive section doesn't make much sense. So, it's not surprising that there were no items that satisfied that request. Similar problems can be much more subtle, however. For example, the default value of *Condition* is "New." In part, this is because Amazon only sells new items. If you submitted a request without specifying a value for *Condition* and received no results, you might assume that there are no items that satisfy the request. By setting *Condition* to "All," however, you might find that items do satisfy the request.

A similar error message is displayed when, for example, you use **`ItemLookup`** to find an item that does not exist in the Amazon catalog.

```
<Errors>
  <Error>
    <Code>AWS.InvalidParameterValue</Code>
    <Message>B0111NOSUCHITEM is not a valid value for ItemId. Please change
this value and retry your request.
    </Message>
  </Error>
</Errors>
```

ItemId was set equal to "B0111NoSuchItem" in the request. The error message shows that Product Advertising API converts *ItemId* values to all caps.

Results and Errors

As long as the request is valid, Product Advertising API will try to process it. There are times when Product Advertising API returns a result and an error. For example, if you had a batch request, one item might be found, the other not. In that case, *`IsValid`* would be true, the response would contain the item attributes of the item found, and the response would also contain an error message that the second item could not be found. Product Advertising API attempts to return as much information as possible even when an error prevents the successful completion of a portion of a request.

Retrieving Errors

Typically, you want your application to check whether or not a request generated an error before spending any time processing results. The easiest way to find out if an error occurred is to look for an *`Error`* node in the response.

XPath syntax provides a simple way to search for the presence of an *`Error`* node, as well as an easy way to retrieve the error code and message. The following code snippet uses Perl and the XML::XPath module to determine if an error occurred during a request. If an error occurred, the code prints the first error code and message in the response.

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ( $xp->find("//Error") )
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Troubleshooting Applications

We recommend the following processes to diagnose and resolve problems with your Product Advertising API-enabled Web sites or applications.

- Verify that Product Advertising API is running correctly.
  To do this, simply open a browser window and submit a REST request. This will confirm that the service is available and responding to requests. Normally Product Advertising API is available 24 hours per day, 7 days per week.
- Verify that your XSL style sheets are valid.
  To do this, run your requests both with and without the XSL style sheet, to determine if the problem is in the request or in your style sheets.
- Check that you are specifying inclusive response groups.
  You may not receive the expected information in your Product Advertising API response if you have specified a response group(s) that does not return the data you want.
- Check the structure of your request.
  Each Product Advertising API operation has a reference page. Double check that you are using parameters correctly. Try the request on one of the other locales. If SOAP is not working, try your request with REST through your browser. That will help you determine if the problem lies with your code or your SOAP client or with Product Advertising API.
- If your application is submitting requests faster than once per second per IP address, you may receive error messages from Product Advertising API until you decrease your rate of requests.
- Look at the sample applications to see if they are doing similar operations to give you hints on what you might be doing wrong.
- Try your request with multiple ASINs or keyword/title search strings.
  The Amazon catalog is extremely large and sometimes individual products have anomalous data results.

Efficiency Guidelines

If your application is trying to submit requests that exceed the maximum hourly request limit for your account, you may receive error messages from Product Advertising API. The hourly request limit for each account is calculated based on revenue performance. Each account used to access the Product Advertising API is allowed an initial usage limit of 2,000 requests per hour. Each account will receive an additional 500 requests per hour (up to a maximum of 25,000 requests per hour) for every $1 of shipped item revenue driven per hour in a trailing 30-day period. You could verify that your sales are being attributed to your calls to the Product Advertising API by checking for the following:

- You are using the only links provided by the API when linking back to Amazon.
- Your Associate Account and Product Advertising API account are created using the same Amazon Account (or email).
- You are passing your Associate tag in all your requests to the API.

Error Codes and Messages

Product Advertising API (Product Advertising API) errors provide you with information about syntactical errors in your requests, as well as errors that occur during the execution of your request (for example, a search for products returns no results). Errors are composed of two elements: *code* and *message*. The error code is a unique string that identifies the error; the error message is a human-readable description of the error that serves as an aid in debugging. These elements will be nested within an Error element. If a request generates more than one error, all Errors will appear in the response.

Errors may appear at different levels in your response. Their location reflects at what stage in the execution of the request the error was generated and what kind of error it is. Errors in syntax that prevent requests from being executed will appear as children of the response's root element. An error associated with a particular item in the response will be a child of the Item element. See the sample requests for examples of each of these situations.

Error Codes

Product Advertising API returns errors in three categories so that you can easily determine how best to handle the problem:

- 2XX errors are caused by mistakes in the request. For example, your request might be missing a required parameter. The error message in the response gives a clear indication what is wrong.
- 4XX errors indicate the request was not authenticated correctly.
- 5XX errors are non-transient errors reflecting an error internal to Amazon. A 503 error means that you are submitting requests too quickly and your requests are being throttled. If this is the case, you need to slow your request rate to one request per second.

Error Messages

Product Advertising API returns error messages in English for the Amazon.com (US), Amazon.co.uk (UK), Amazon.de (DE), Amazon.fr (FR), and Amazon.ca (CA) locales. Error messages are in Japanese for the Amazon.co.jp (JP) locale.

Error Code & Description	Message	Affected Operations
AWS.ExactParameterRequirement You will receive this message when the value of your parameter is longer than permitted by Product Advertising API.	Your request contains too much data for *`[ParameterName]`*. This parameter can have a maximum length of *`[MaximumNumber]`*.	All
AWS.ExceededMaximumParameterValues You will receive this error message when you specify too many values for one or more parameters in your request.	Your request contains too many values for *`[ParameterName]`*. This parameter can have a maximum of *`[MaximumNumber]`* values.	• `ItemLookup` • `SimilarityLookup`
AWS.InsufficientParameterValues You will receive this error message when your request contains an insufficient number of values for a required parameter.	Your request contains too few values for *`[Parameter Name]`*. This parameter must have a minimum of *`[Minimum Value]`* values.	All
AWS.InternalError You will receive this error if Product Advertising API is unable to complete your request due to an internal problem or outage. For SOAP, this will be presented as a SOAP fault rather than an error.	We are unable to process your request at this time. Please retry your request. If you encounter this error repeatedly, please post a message on the AWS discussion board.	All

Error Code & Description	Message	Affected Operations
AWS.InvalidEnumeratedParameter You will receive this error message when your request contains an invalid value for a parameter that has an explicit list of valid values, such as SearchIndex.	The value you specified for *[ParameterName]* is invalid. Valid values include *[EnumeratedValuesList]*.	• `CartAdd` • `CartCreate` • `CartModify` • `ItemLookup` • `ItemSearch` • `SimilarityLookup`
AWS.InvalidISO8601Time You will receive this error when your request contains a date or time value that is not formatted according to the profile of the ISO-8601 date/time standard that is described at http://www.w3.org/TR/NOTE-datetime. For example, this error will be returned if your request contains an invalid value for the Version parameter.	*[ParameterName]*has an invalid value. It must contain a valid ISO 8601 date and time.	All
AWS.InvalidOperationForMarketplace You will receive this error message when you try to execute an operation in a locale where the operation is not supported.	This operation, *[OperationName]*, is not available for this locale.	All
AWS.InvalidOperationParameter You will receive this error message when the operation name you entered is not available from Product Advertising API. For instance, if you tried to use AsinSearch (from AWS 3.0) as an operation name, you would get this error since AsinSearch is not a valid operation name in Product Advertising API 4.0.	The Operation parameter is invalid. Please modify the Operation parameter and retry. Valid values for the Operation parameter include *[ListOfOperationValues]*.	All
AWS.InvalidParameterCombination You will receive this error message when two or more of the request parameters you have entered can not be used in the same request. For example, if you are using the CartAdd operation, you would receive this error if you tried to add items to the cart by both ASIN and OfferListingId.	Your request contains an invalid parameter combination. *[ParameterName]* and *[ParameterName]* cannot appear in the same request.	• `ItemSearch` • `CartCreate` • `CartAdd`
AWS.InvalidParameterValue You will receive this error message when your request contains an invalid value for an ID parameter, such as ItemId.	*[ParameterValue]* is not a valid value for *[ParameterName]*. Please change this value and retry your request.	• `CartAdd` • `CartCreate` • `CartModify` • `ItemLookup` • `ItemSearch` • `SimilarityLookup`

Error Code & Description	Message	Affected Operations
AWS.InvalidResponseGroup You will receive this error message when the response group name you entered in your request is incompatible with the operation you would like to perform.	Your ResponseGroup parameter is invalid. Valid response groups for *[Operation Name]* requests include *[Available Response Group List]*.	All
AWS.InvalidServiceParameter You will receive this error message when the service name you provide in your request is not recognized or supported by Amazon. All Product Advertising API requests should use the service name "AWSECommerceService."	The Service parameter is invalid. Please modify the Service parameter and retry. Valid values for the Service parameter include *[ValidServicesList]*.	All
AWS.InvalidSubscriptionId You will receive this error message when the subscription ID you use in your request is not recognized by AWS.	Your request contains an invalid subscription ID. Please retry your request with a valid subscription ID.	All
AWS.InvalidXSLTAddress You will receive this error if the AWS XSLT service is unable to access the XSLT file you used as the value for the Style parameter in your request	We are unable to access your XSLT file. Please verify that you have specified a valid address to your XSLT file.	All
AWS.MaximumParameterRequirement You receive this error message when your request contains the wrong number of parameters from an exclusive group.	Your request should have at most *[Maximum Number]* of the following parameters: *[Parameter Names]*.	All
AWS.MinimumParameterRequirement You receive this error message when your request contains the wrong number of parameters from an exclusive group.	Your request should have at least *[Minimum Number]* of the following parameters: *[Parameter Names]*.	All
AWS.MissingOperationParameter You will receive this error message when your request does not include the Operation parameter and the name of the operation you would like to perform.	Your request is missing the Operation parameter. Please add the Operation parameter to your request and retry. Valid values for the Operation parameter include *[ValidOperationsList]*.	All
AWS.MissingParameterCombination You will receive this error message when your request does not contain a combination of two or more parameters that must be present together in your request.	Your request is missing a required parameter combination. Required parameter combinations include *[Parameter One]*.	• `ItemLookup`

Error Code & Description	Message	Affected Operations
AWS.MissingParameters You will receive this error when your request does not include all of the parameters required by the operation.	Your request is missing required parameters. Required parameters include *[RequiredParameterList]*.	All
AWS.MissingParameterValueCombination You will receive this error message when your request requires a combination of parameters, one or more of which must have a specific value. For example, when you make an ItemLookup request for a product based on its Universal Product Code (or UPC), you are required to include the IdType and ItemId parameters. The value of the IdType parameter must be UPC.	Your request is missing a required parameter combination. When *[Parameter One]* equals *[Restricted Value]*, *[Parameter Two]* must be present.	• `ItemLookup`
AWS.MissingServiceParameter You will receive this error message when the your request does not contain the required Service parameter.	Your request is missing the Service parameter. Please add the Service parameter to your request and retry. Valid values for the Service parameter include *[ValidServicesList]*.	All
AWS.ParameterOutOfRange You will receive this error message when you submit a parameter value that exceeds or is lower than the range of valid values for the parameter. For example, ItemSearch allows you to fetch search results page using the ItemPage parameter. The range of values for ItemPage is 1 to 10. If you supply a value outside that range (less than 1 or greater than 10), you will receive this error.	The value you specified for *[ParameterName]* is invalid. Valid values must be between *[LowerBound]* and *[UpperBound]*.	• `ItemSearch` • `ItemLookup`
AWS.ParameterRepeatedInRequest You receive this error message when you include the same parameter more than once in your request.	The parameter, *[ParameterName]*, appeared more than once in your request.	All
AWS.RestrictedParameterValueCombination You will receive this error message when your request contains a combination of parameter values that are not permitted in the same request.	Your request contains a restricted parameter combination. When *[Parameter One]* equals *[Restricted Value]*, *[Parameter Two]* cannot be present.	All

Error Code & Description	Message	Affected Operations
AWS.XSLTTransformationError You will receive this error message when the AWS XSLT service is unable to parse or apply the XSLT stylesheet you have provided in the Style parameter in your request. Make sure that your XSLT stylesheet is valid, and try again.	We were unable to apply your XSLT file. Please check your XSLT and retry your request.	All
AWS.ECommerceService.ExceededMaximumCartItems You will receive this error message when you exceed the maximum quantity value allowed for items being added to a shopping cart.	You may not add more than *[Maximum Item Quantity]* items to the cart.	• `CartAdd` • `CartCreate`
AWS.ECommerceService.InvalidCartId You will receive this error message when the CartId you entered into your request is not recognized.	Your request contains an invalid value for CartId. Please check your CartId and retry your request.	• `CartAdd` • `CartClear` • `CartGet` • `CartModify`
AWS.ECommerceService.InvalidHMAC You will receive this error message when the shopping cart HMAC value you use in your request is not recognized by Product Advertising API. The HMAC value is a unique token that is used to associate a cart with an Amazon user and a particular session on the Amazon web site.	Your request contains an invalid value for HMAC. Please check your HMAC and retry your request. Remember that the HMAC must be URL-encoded if you are using REST.	• `CartAdd` • `CartClear` • `CartGet` • `CartModify`
AWS.ECommerceService.InvalidQuantity You will receive this error message when the quantity in your request is not valid for the current item.	You have exceeded the maximum quantity allowed for the following item(s): *[ItemId]*.	• `CartAdd` • `CartCreate` • `CartModify`
AWS.ECommerceService.ItemAlreadyInCart You will receive this error message when you try to add an item to a shopping cart that already contains that item.	The item you specified, *[ItemID]*, is already in your cart.	• `CartAdd` • `CartCreate`
AWS.ECommerceService.ItemNotAccessible Some products cannot be manipulated or viewed using Product Advertising API. You will receive this error message when the product ID you use in your request is not available through Product Advertising API.	This item is not accessible through Product Advertising API.	• `ItemLookup`

Error Code & Description	Message	Affected Operations
AWS.ECommerceService.ItemNotEligibleForCart Some products cannot be manipulated or viewed using Product Advertising API. You will receive this error message when you attempt to add such an item to a remote shopping cart.	The item you specified, *[ItemID]*, is not eligible to be added to the cart. Check the item's availability to make sure it is available.	• `CartAdd` • `CartCreate` • `CartModify`
AWS.ECommerceService.NoExactMatches	We did not find any matches for your request.	• `ItemSearch`
AWS.ECommerceService.NoSimilarities	There are no similar items for this ASIN(s): *[ItemID]*.	• `SimilarityLookup`
AccountLimitExceeded You will receive this error if your application is trying to submit requests that exceed the maximum hourly request limit for your account	Account limit of X requests per hour exceeded (X = the hourly request limit for your account)	All
RequestThrottled You will receive this error if your application is submitting requests faster than once per second per IP address.	You will receive this error if your application is submitting requests faster than once per second per IP address.	

Best Programming Practices

Topics

The following checklist of best practices describes how you can increase the effectiveness of your Product Advertising API (Product Advertising API) 4.0 applications.

Read the Product Advertising API Terms and Conditions

The Product Advertising API Terms and Conditions spell out in detail the limitations that Amazon enforces on all Product Advertising API applications. The thrust of all Product Advertising API applications should be to direct sales to Amazon and thus earn Associate sales commissions. If your application is designed

around another purpose, please reconsider and make sure your proposed Product Advertising API application falls within the guidelines of the Product Advertising API Terms and Conditions. Applications that do not meet the Product Advertising API Terms and Conditions will be blocked from accessing Product Advertising API.

Use the Latest API Version

Product Advertising API makes frequent releases. Each release either adds functionality or increases the accuracy, speed, and stability of Product Advertising API. Most releases do all of these. Go to http://aws.amazon.com/resources frequently for documentation about the latest release.

To use the latest version of the API, you must include the `Version` parameter in your requests. Otherwise, Product Advertising API uses the default API version, which is from 2005-10-05.

Product Advertising API 3.0 has been deprecated and will be shut down at the end of March 2008. Make sure you develop against Product Advertising API 4.0.

Understand Available Operations

Product Advertising API provides many different operations to facilitate product discovery. Using the right operations can dramatically enhance your customer's shopping experience and increase your Associate commissions. See the API Reference for a full description of all Product Advertising API operations. Some enable you to:

Task	Operations
Find products and categories	`ItemSearch`, `ItemLookup`, and `BrowseNodeLookup`
Find similar items	`SimilarityLookup`,
Provide shopping cart functionality for your application or web site	`CartCreate`, `CartAdd`, `CartModify`, and `CartGet`

Use the Right Response Group

One of the great features of Product Advertising API is the control you have over the amount of information returned in responses. A response group is a collection of data returned by Product Advertising API. Product Advertising API has over 55 response groups, each serving a different need. So, you can get exactly the information you need and no more.

We recommend that you specify response groups that return only the information your application needs. Response groups, such as Large, and ItemAttributes return lots of data. Such large data sets sometimes incur performance penalties both in Product Advertising API fulfilling the request and in your application's processing of the response.

Use Your Associate Tag in Product Advertising API Requests

To earn commissions for selling Amazon items, you must register with Amazon as an Associate. Go to http://amazon.com/associates. In return, you receive an Associate tag, which identifies you. By including your Associate tag in each Product Advertising API request, you receive commissions for customer purchases.

Amazon also uses your Associate tag to monitor your use of Product Advertising API, which helps us determine how to improve our web service.

Handling Errors

Make sure your application handles errors gracefully. One way to do that is to check the status of the `IsValid` element. `IsValid` is returned with every request. If its value is "False," there will be an error message with a description of why your request was not valid.

It's generally a good practice to log any unexpected error that is returned by Product Advertising API. Errors are returned with an error code and message. The code is a descriptive string that identifies the error. The error message is a more 'human friendly' message that can be displayed to your customers. Your application should be able to handle expected error messages.

Your application, for example, should display error messages that are meaningful to the customer. For example, when Product Advertising API responds with the error, "AWS.ECommerceService.NoExactMatches," your application should display an explanation, such as, "We did not find any matches for your request."

If your application exceeds the number of allowed requests submitted per second, Product Advertising API returns a 503 error, which means that Product Advertising API is restricting the number of requests it is processing from your application. The Product Advertising API Terms and Conditions outline the number of allowed requests permitted per second.

Use Caches Carefully

Product Advertising API product data changes often. Prices can change hourly, Browsenode values change without notice, and product availability is volatile. For these reasons, you should not cache product data.

It is against the Product Advertising API Terms and Conditions to cache customer information derived from Amazon.

You can enhance the performance of your application by caching identifiers, such as the `CartId`, `HMAC`, and `PurchaseURL`.

Use the Correct AWSAccessKeyId

To become a Product Advertising API developer, you must register with Amazon. Go to http://aws.amazon.com and create a web service account. Your account contains an `AWSAccessKeyId` and a `SecretKey`. Every Product Advertising API request you submit must include the `AWSAccessKeyId`. Because your account is linked to your e-mail address, Amazon can contact you easily

Community Forums

Product Advertising API has an active and passionate developer community. It's a great place to get API questions answered or share ideas with other developers. To join, go to http://developer.amazonwebservices.com/connect/forum.jspa?forumID=9

Locale Considerations

Topics

As you have seen throughout the previous chapters, there are subtle differences in Product Advertising API functionality across locales. The operations are the same for all locales but the valid response groups, search indices, and sort values vary from one locale to another. These differences are captured in the appendices of this manual.

This chapter discusses locale considerations in the following sections.

Locales

Amazon is a world-wide venture. Product Advertising API is as well. As you saw in earlier chapters, Product Advertising API operates in six locales:

- CA
- DE
- FR
- JP
- UK
- US

Each of these locales is serviced by an Amazon web site that uses the local language, local customs, and local formatting. For example, when you look at the DE homepage for Amazon, you see the listings in German. If you purchased an item, you would find the price in Euros, and, if you were to purchase a movie, you would find that the movie rating would conform to the movie rating system used in Germany. Product Advertising API responses contain the same localized information. Product Advertising API determines the correct locale by examining the endpoint in the request. For example, the endpoints for the DE locale are:

```
http://ecs.amazonaws.de/onca/xml
https://ecs.amazonaws.de/onca/xml
```

For a list of the other endpoints, see Anatomy Of a REST Request (p. 48).

Currency, for example is localized, as follows.

```
<SubTotal>
  <Amount>4082</Amount>
  <CurrencyCode>USD</CurrencyCode>
  <FormattedPrice>$40.82</FormattedPrice>
</SubTotal>
```

The values for *CurrencyCode* are similar to the names of Product Advertising API locales. In the preceding example, the currency is formatted according to the conventions in the US locale.

General Differences

Each locale serves a different segment of the world population. As a result, the items sold by Amazon vary to the demands of the customers. In addition to Amazon web sites being localized by language, custom, and formatting conventions, the locales vary in many ways, as described in the following table.

Difference	Description
Items for sale.	The selection of DVDs, for example, sold in the JP locale are different than those sold in the DE locale. The same is true for books. Not only are the languages of the books different, the selection of books varies by locale.
Customer feedback.	Amazon encourages customers to share their feedback on items, sellers and merchants. Customer reviews are restricted to the locale in which they are entered.
Sellers and merchants.	Sellers and merchants can do business in multiple locales but they don't have to. As a result, items from a seller might be available in only one locale. The largest merchants, beside Amazon, are available in the US locale only.
Customer accounts	Customer accounts are restricted to a locale. A customer can create accounts in every locale.
Item identifiers	Item identifiers, such as ASINs, are unique to a locale, that is, the same ASIN value can refer to different items in different locales.
Availability of some response groups	The availability of some response groups, sort parameters, and search indices varies by locale.

Associate IDs

To be an Associate, you must sign up in each locale in which you intend to do business. That means, for example, if you have an Associate ID for the US locale, you will not get credit if you submit a `PurchaseURL` in the DE locale. To get credit, you must get an Associate ID in the DE locale. For a list of web sites where you can register as an Associate, see Becoming an Associate.

Shipping Restrictions

Amazon places shipping restrictions based on item and locale. The sale of items must obey the rules and regulations of the host country. It is possible, for example, that a host country would restrict the sale of some electronic equipment to specific countries. Some locales restrict shipping to specified countries regardless of the item. For example, in the UK locale, shipping is restricted to Ireland, Scotland, and England.

Because Amazon is used to fulfill the orders, Amazon takes charge of restricting shipping. Your application or web site, however, should be aware of the shipping restrictions.

To see the latest details on shipping restrictions for each locale, go to *http://www.amazon.com/gp/help/customer/display.html?nodeId=468634*.

Shipping costs also play an important role in helping your customers purchase items for the lowest possible price. For example, Harry Potter books are available in all locales. A customer in the US could purchase a copy of a Harry Potter book from the UK locale. The shipping charges, however, would be significantly higher than if the same item were purchased through the US locale.

For more information about shipping, go to *http://www.amazon.com/gp/help/customer/display.html*.

API Reference

Topics

The following sections of the guide provide reference material for the Product Advertising API. For more information about any concepts or programming tasks associated with the reference material, refer to the previous chapters in this guide.

Operations

The following operations are available in the Product Advertising API.

BrowseNodeLookup

Description

Given a browse node ID, `BrowseNodeLookup` returns the specified browse node's name, children, and ancestors. The names and browse node IDs of the children and ancestor browse nodes are also returned. `BrowseNodeLookup` enables you to traverse the browse node hierarchy to find a browse node.

As you traverse down the hierarchy, you refine your search and limit the number of items returned. For example, you might traverse the following hierarchy: DVD>Used DVDs>Kids and Family, to select out of all the DVDs offered by Amazon only those that are appropriate for family viewing. Returning the items associated with Kids and Family produces a much more targeted result than a search based at the level of Used DVDs.

Alternatively, by traversing up the browse node tree, you can determine the root category of an item. You might do that, for example, to return the top seller of the root product category using the *`TopSeller`* response group in an ItemSearch (p. 207) request.

You can use `BrowseNodeLookup` iteratively to navigate through the browse node hierarchy to reach the node that most appropriately suits your search. Then you can use the browse node ID in an `ItemSearch` (p. 207) request. This response would be far more targeted than, for example, searching through all of the browse nodes in a search index.

Availability

All locales

Request Parameters

Name	Description	Required
`BrowseNodeId`	A positive integer assigned by Amazon that uniquely identifies a product category. Type: String Default: None Valid Values:A positive integer.	Yes
`ResponseGroup`	Specifies the types of values to return. You can specify multiple response groups in one request by separating them with commas. Default: BrowseNodeInfo (p. 228) Valid Values: MostGifted (p. 259) | NewReleases (p. 262) | MostWishedFor (p. 260) | TopSellers (p. 291)	No

`BrowseNodeLookup` also accepts the parameters that all operations can use. For more information, see, Common Request Parameters (p. 221)

Response

Name	Description
Ancestors	Container object for a parent browse node.
BrowseNode	Container object for all browse node data, including browse node ID, browse node name, browse node children and ancestors.
BrowseNodeId	A positive integer that uniquely identifies a product group, such as Literature & Fiction: (17), Medicine: (13996), and Mystery & Thrillers: (18).
Children	Container for one or more browse nodes, which are the children of the browse node submitted in the request.
Name	Name of the BrowseNode, for example, the name of BrowseNode 17 is Literature & Fiction.

For more information about the parent elements of these tags, see the appropriate response group in Response Groups (p. 224)

Examples

Use `BrowseNodeLookup` iteratively to navigate through the hierarchy of browse nodes. In this way, customers can refine their searches, for example.

```
DVD>Actors & Actresses>Steve Martin
DVD>Used DVDs>Kids & Family
```

The first hierarchy narrows the search down to DVDs in which Steve Martin plays a part. The second hierarchy narrows the list of DVDs down to those that are suitable for Kids and Family. This node, for example, might contain thirty such DVD titles.

The following request returns the name, parent, and children of the browse node for DVD comedies (163357)

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=163357
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The response to this request is shown in Sample Response (p. 179).

Use the NewReleases (p. 262) response group to display the newly released items for a specified browse node:

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=163357&
```

```
ResponseGroup=NewReleases
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response

The following XML is a snippet of the full response to the first sample request.

```
<BrowseNode>
  <BrowseNodeId>163357</BrowseNodeId>
    <Name>Comedy</Name>
    <Children>
    <BrowseNode>
      <BrowseNodeId>599826</BrowseNodeId>
      <Name>Boxed Sets</Name>
    </BrowseNode>
    <BrowseNode>
      <BrowseNodeId>538712</BrowseNodeId>
      <Name>African American Comedy</Name>
    </BrowseNode>
    <BrowseNode>
      <BrowseNodeId>163358</BrowseNodeId>
      <Name>Animation</Name>
    </BrowseNode>
    <BrowseNode>
      <BrowseNodeId>720556</BrowseNodeId>
      <Name>Black Comedy</Name>
    </BrowseNode>
    <BrowseNode>
      <BrowseNodeId>291102</BrowseNodeId>
      <Name>British</Name>
    </BrowseNode>
  </Children>
  <Item>
    <Ancestors>
      <BrowseNode>
        <BrowseNodeId>549726</BrowseNodeId>
        <Name>Performing Arts</Name>
        <Ancestors>
          <BrowseNode>
            <BrowseNodeId>5</BrowseNodeId>
            <Name>Entertainment</Name>
            <Ancestors>
              <BrowseNode>
                <BrowseNodeId>1000</BrowseNodeId>
                <Name>Subjects</Name>
                <Ancestors>
                  <BrowseNode>
                    <BrowseNodeId>1000</BrowseNodeId>
                    <Name>Books</Name>
```

This response shows that the browse node, Comedy, which is specified in the request, has five direct descendants (Boxed Sets, African American Comedy, Animation, Black Comedy, and British) and its ancestors are Books>Subjects>Entertainment>Performing>Comedy.

Related Operations

- ItemSearch (p. 207)
- ItemLookup (p. 200) using the BrowseNodes (p. 230) response group

CartAdd

Description

The `CartAdd` operation enables you to add items to an existing remote shopping cart. `CartAdd` can only be used to place a new item in a shopping cart. It cannot be used to increase the quantity of an item already in the cart. If you would like to increase the quantity of an item that is already in the cart, you must use the `CartModify` operation.

You add an item to a cart by specifying the item's *`OfferListingId`*, or *`ASIN`* and *`ListItemId`*. Once in a cart, an item can only be identified by its *`CartItemId`*. That is, an item in a cart cannot be accessed by its ASIN or *`OfferListingId`*. *`CartItemId`* is returned by CartCreate (p. 187), CartGet (p. 192), and CartAdd (p. 180).

To add items to a cart, you must specify the cart using the *`CartId`* and *`HMAC`* values, which are returned by the `CartCreate` operation.

If the associated CartCreate (p. 187) request specified an *`AssociateTag`*, all `CartAdd` requests must also include a value for Associate Tag otherwise the request will fail.

Note

Some manufacturers have a minimum advertised price (MAP) that can be displayed on Amazon's retail web site. In these cases, when performing a Cart operation, the MAP Is returned instead of the actual price. The only way to see the actual price is to add the item to a remote shopping cart and follow the `PurchaseURL`. The actual price will be the MAP or lower.

Availability

All locales.

Request Parameters

Name	Description	Required
`ASIN`	Specifies ASIN of item to be added to the cart where N is a positive integer between 1 and 10, inclusive. Up to ten items can be added at a time. Using an item's *`OfferListingId`* is preferred instead of the item's ASIN. Type: String Default: None Valid Values: Valid ASIN Constraint: Required if an OfferListingId is not specified.	Conditional

Name	Description	Required
CartId	Alphanumeric token returned by CartCreate (p. 187) that identifies a cart. Type: String Default: None Valid Values: Value returned by `CartCreate`.	Yes
HMAC	The Hash Message Authentication Code is an encrypted alphanumeric token that is used to authenticate requests. Type: String Default: None Valid Values: Value is calculated using request parameters, their values, a cryptographic function, and the Secret Key, which acts as the "key" for the function.	Yes
Item	Container for *ASIN* or *OfferListingId*, and *Quantity*. This is a SOAP only parameter. Valid Values: An *ASIN* or an *OfferListingId* is required.	Yes (SOAP only)
Items	Container for one or more *Item*(s). This is a SOAP only parameter. Valid Values: Item	Yes (SOAP only)
MergeCart	This parameter is deprecated. When a customer purchases items in a remote shopping cart, the items are added to the customer's Amazon retail shopping cart.	Deprecated
OfferListingId	An offer listing ID is a token that uniquely identifies an item that is sold by any merchant, including Amazon. This parameter is preferred to using an ASIN to add an item to a cart. Type: String Default: None Valid Values: Valid offer listing ID Constraint: Required if ASIN is not offered .	Conditional
Quantity	Specifies number of items to be added to the cart where N is a positive integer. The Quantity value is matched to the Item.N.ASIN value by the index number, N. For example, Item.1.ASIN is associated with Item.1.Quantity. Type: String Default: None Valid Values: Positive integer between 1 and 999, inclusive. Conditional Required for REST	Conditional

Name	Description	Required
ResponseGroup	Specifies the types of values to return. You can specify multiple response groups in one request by separating them with commas. Type: String Default: Cart (p. 232) Valid Values: CartSimilarities (p. 237) | CartTopSellers (p. 236) | NewReleases (p. 262)	No

`CartAdd` also accepts the parameters that all operations can use. For more information, see, Common Request Parameters (p. 221)

Response

Name	Description
Amount	Price of the item in terms of the lowest currency denomination, for example, pennies.
ASIN	An alphanumeric token that uniquely identifies an item sold by Amazon.
CartAddRequest	Container for *CartId*, *HMAC*, and *Items*.
CartId	Alphanumeric token returned by `CartCreate` that identifies a cart.
CurrencyCode	Format for the display of the money
FormattedPrice	The price to display on the web site.
HMAC	Hash Message Authentication Code returned by `CartCreate` that identifies a cart. This is an encrypted alphanumeric token that is used to authenticate cart operations.
Item	Container for *ASIN* or *OfferListingId*, and *Quantity*.
Items	Container for one or more item(s).
PurchaseURL	URL that customers should use to purchase the items in the cart. It includes the Associate's ID. It is important that they use this URL otherwise the Associate will not get credit for the purchase.
Quantity	Number of items added to cart.
Subtotal	Container for Amount, CurrencyCode, and FormattedPrice

For more information about the parent elements of these tags, see the appropriate response group in Response Groups (p. 224)

Examples

Add a single item to the cart:

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
```

```
CartId=[CartId]&
HMAC=[HMAC]&
Operation=CartAdd&
Item.1.OfferListingId=[Offer Listing ID]&

Item.1.Quantity=1
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The response to this request is shown in, Response to Sample Request (p. 179).

Add multiple items to the cart.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
CartId=[CartId]&
HMAC=[HMAC]&
Operation=CartAdd&
Item.1.OfferListingId=[Offer Listing ID]&
Item.1.Quantity=1&
Item.2.OfferListingId=[Offer Listing ID]&
Item.2.Quantity=3&
Item.3.OfferListingId=[Offer Listing ID]&
Item.3.Quantity=1
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request adds three different items to a cart. A cart can carry up to fifty different items and each item can have a maximum *`Quantity`* value of 999..

Add to the cart an item from a list.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
CartId=[CartId]&
HMAC=[HMAC]&
Operation=CartAdd&
Item.1.ASIN=[ASIN]&
Item.1.Quantity=3&
Item.1.ListItemId=[List item ID]
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request adds 3 items from a list.

Sample Response

The following XML is a snippet of the full response to the first sample request.

```
<CartAddRequest>
  <CartId>102-6666665-5792105</CartId>
  <HMAC>oRqNBTpiRBugxEKEJ+FyUZkVbEg=</HMAC>
  <Items>
```

```
    <Item>
      <ASIN>1400042127</ASIN>
      <Quantity>1</Quantity>
    </Item>
  </Items>
</CartAddRequest>
</Request>
  <CartId>102-6666665-5792105</CartId>
  <HMAC>oRqNBTpiRBugxEKEJ+FyUZkVbEg=</HMAC>
  <URLEncodedHMAC>oRqNBTpiRBugxEKEJ%2BFyUZkVbEg=</URLEncodedHMAC>
<PurchaseURL>https://www.amazon.com/gp/cart/aws-merge.html?cart-id=102-5929035-
5792105%26associate-
id=ws%26hmac=oRqNBTpiRBugxEKEJ%2BFyUZkVbEg=%26AWSAccessKeyId=[AWS Access Key
ID]</PurchaseURL>
  <SubTotal>
    <Amount>3432</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$34.32</FormattedPrice>
  </SubTotal>
```

Related Operations

- CartCreate (p. 187)
- CartGet (p. 192)
- CartModify (p. 196)
- CartClear (p. 184)

CartClear

Description

The `CartClear` operation enables you to remove all of the items in a remote shopping cart, including SavedForLater items. To remove only some of the items in a cart or to reduce the quantity of one or more items, use `CartModify` (p. 196).

To delete all of the items from a remote shopping cart, you must specify the cart using the *`CartId`* and *`HMAC`* values, which are returned by the `CartCreate` operation. A value similar to the HMAC, *`URLEncodedHMAC`*, is also returned. This value is the URL encoded version of the *`HMAC`*. This encoding is necessary because some characters, such as + and /, cannot be included in a URL. Rather than encoding the *`HMAC`* yourself, use the *`URLEncodedHMAC`* value for the *`HMAC`* parameter.

`CartClear` does not work after the customer has used the *`PurchaseURL`* to either purchase the items or merge them with the items in their Amazon cart.

Carts exist even though they have been emptied. The lifespan of a cart is 7 days since the last time it was acted upon. For example, if a cart created 6 days ago is modified, the cart lifespan is reset to 7 days.

Availability

All locales.

Request Parameters

Name	Description	Required
AssociateTag	Alphanumeric token that uniquely identifies an Associate. This value is required. Valid Values: The same *AssociateTag* that was used in the related `CartCreate` request.	Yes
CartId	Alphanumeric token returned by `CartCreate` that identifies a cart. Type: String Default: None Valid Values: Value returned by `CartCreate`.	Yes
HMAC	The Hash Message Authentication Code is an encrypted alphanumeric token that is used to authenticate requests. Type: String Default: None Valid Values: Value is calculated using request parameters, their values, a cryptographic function, and the Secret Key, which acts as the "key" for the function.	Yes
MergeCart	This parameter is deprecated. When a customer purchases items in a remote shopping cart, the items are added to the customer's Amazon retail shopping cart.	Deprecated
ResponseGroup	Specifies the types of values to return. You can specify multiple response groups in one request by separating them with commas. Type: String Default: Cart (p. 232)	No

`CartClear` also accepts the parameters that all operations can use. For more information, see, Common Request Parameters (p. 221).

Response

Name	Description
Cart	Container for remote shopping cart related response elements.
CartClearRequest	Container for *CartClear* and *HMAC*.
CartId	Alphanumeric token returned by `CartCreate` that uniquely identifies a cart.

Name	Description
HMAC	Hash Message Authentication Code returned by `CartCreate` that identifies a cart. This is an encrypted alphanumeric token that is used to authenticate cart operations.
URLEncodedHMAC	URL encoded version of the *HMAC*. This encoding is necessary because some characters, such as + and /, cannot be included in a URL.

For more information about the parent elements of these tags, see the appropriate response group in Response Groups (p. 224)

Examples

Remove all items from a specified cart:

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate Tag]&
Operation=CartClear&
CartId=[Cart ID]&
HMAC=[URL-encoded HMAC]
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This operation removes all of the items in the specified cart. Because the *HMAC* includes the *AssociateTag*, you must supply it in this request.

The response to this request is shown in, Response to Sample Request (p. 179).

Sample Response

The following XML is a snippet of the full response to the first sample request.

```
<Cart>
  <Request>
    <IsValid>True</IsValid>
    <CartClearRequest>
      <CartId>102-2689399-8023324</CartId>
      <HMAC>iLYO/W0hft0Te4uXAbMiBhs36T8=</HMAC>
    </CartClearRequest>
  </Request>
  <CartId>102-2689399-8023324</CartId>
  <HMAC>iLYO/W0hft0Te4uXAbMiBhs36T8=</HMAC>
  <URLEncodedHMAC>iLYO/W0hft0Te4uXAbMiBhs36T8=</URLEncodedHMAC>
</Cart>
```

Related Operations

- CartAdd (p. 180)
- CartCreate (p. 187)
- CartGet (p. 192)
- CartModify (p. 196)

CartCreate

Description

The `CartCreate` operation enables you to create a remote shopping cart. A shopping cart is the metaphor used by most e-commerce solutions. It is a temporary data storage structure that resides on Amazon servers. The structure contains the items a customer wants to buy. In Product Advertising API, the shopping cart is considered remote because it is hosted by Amazon servers. In this way, the cart is remote to the vendor's web site where the customer views and selects the items they want to purchase.

Once you add an item to a cart by specifying the item's *`ListItemId`* and ASIN, or *`OfferListing`* ID, the item is assigned a *`CartItemId`* and accessible only by that value. That is, in subsequent requests, an item in a cart cannot be accessed by its *`ListItemId`* and ASIN, or *`OfferListingId`*. *`CartItemId`* is returned by CartCreate (p. 187), CartGet (p. 192), and CartAdd (p. 180).

Because the contents of a cart can change for different reasons, such as item availability, you should not keep a copy of a cart locally. Instead, use the other cart operations to modify the cart contents. For example, to retrieve contents of the cart, which are represented by *`CartItemIds`*, use CartGet (p. 192).

Available products are added as cart items. Unavailable items, for example, items out of stock, discontinued, or future releases, are added as SaveForLaterItems. No error is generated. The Amazon database changes regularly. You may find a product with an offer listing ID but by the time the item is added to the cart the product is no longer available. The checkout page in the Order Pipeline clearly lists items that are available and those that are SaveForLaterItems.

It is impossible to create an empty shopping cart. You have to add at least one item to a shopping cart using a single `CartCreate` request. You can add specific quantities (up to 999) of each item.

`CartCreate` can be used only once in the life cycle of a cart. To modify the contents of the cart, use one of the other cart operations.

Carts cannot be deleted. They expire automatically after being unused for 7 days. The lifespan of a cart restarts, however, every time a cart is modified. In this way, a cart can last for more than 7 days. If, for example, on day 6, the customer modifies a cart, the 7 day countdown starts over.

Specifying the Items

In one `CartCreate` request, you can add up to ten items; the quantity of each item is set separately, as follows:

```
Item.1.OfferListingId=[Offer Listing ID]
Item.1.Quantity=3
Item.2.OfferListingId=[Offer Listing ID]
Item.2.Quantity=1
```

This code adds two items identified by their OfferListingIds and the quantity of each item is set to 3 and 1, respectively.

You can use any combination *`ListItemId`*s and *`ASIN`*s, or *`OfferListingId`*s in a single request to specify the items to add to a cart. You can only use one type of item identifier to add an item to a cart. You cannot, for example, use an *`ASIN`* and an *`OfferListingId`* in the same request to add the same item to a cart.

For more information the formatting used to add items to a cart, see CartAdd (p. 180).

Availability

All locales.

Request Parameters

Name	Description	Required
ASIN	An alphanumeric token that uniquely identifies an item. Using OfferListingId is preferred instead of ASIN when adding items to a cart. Valid Values: ASIN Constraint: Required if OfferListingId is not used	Conditional
AssociateTag	An alphanumeric token that uniquely identifies an Associate. If this value is not included in the operation, request will return an error response. The AssociateTag must also be included on all future cart operations associated with this cart. Type: String Default: None Valid Values: An Associate Tag.	Yes
Item	For REST, a prefix for *ASIN* and quantity, both of which are used to specify the item to add to the cart, for example, item.1.*ASIN*=1234abcd, item.1.quantity=2. Valid Values: Does not take a value. Type: String Default: None	Yes
Items	Container for one or more Item objects. This parameter is for SOAP only.	No
ListItemId	An alphanumeric token that uniquely identifies an item on a wishlist. This value is returned by the ListItem response group. The *ListItemId* attaches to the request the name and address of the list owner, which the *ASIN* alone does not. An *ASIN* or OfferListingId is still required in addition to *ListItemId*. Type: String Default: None Valid Values: An alphanumeric token returned by the ListItem Response Group.	No
MergeCart	This parameter is deprecated. When a customer purchases items in a remote shopping cart, the items are added to the customer's Amazon retail shopping cart. Type: String	Deprecated

Name	Description	Required
OfferListingId	An offer listing ID is an alphanumeric token that uniquely identifies an item. Use the OfferListingId instead of an item's ASIN to add the item to the cart. Type: String Default: None Valid Values: An Offer Listing ID Constraint: Required if *ASIN* is not used.	Conditional
Quantity	The suffix used with Item to specify the number of items to be added to the cart, for example, Item.1.*ASIN*=0976925524 Item.1.Quantity=2. See *Item*. Type: String Default: None Valid Values: Positive integer	No
ResponseGroup	Specifies the types of values to return. You can specify multiple response groups in one request by separating them with commas. Type: String Default:Cart (p. 232) Valid Values: CartSimilarities (p. 237) | CartTopSellers (p. 236) | CartNewReleases (p. 235)	No

`CartCreate` also accepts the parameters that all operations can use. For more information, see, Common Request Parameters (p. 221).

Response

Name	Description
Amount	Price of the item in terms of the lowest currency denomination, for example, pennies.
ASIN	An alphanumeric token that uniquely identifies an item.
Cart	Container for all of the elements in the cart.
CartCreateRequest	Container for all of the items requested.
CartId	An alphanumeric token that uniquely identifies a cart. This value must be included in all future cart operations related to this cart.
CartItems	Container for purchase related information about each item in the cart
CurrencyCode	Locale of the sale that specifies the formatting of the price.
FormattedPrice	The price of the item as it should appear to the customer.
HMAC	An encoded value used to authenticate a request. This value must be included in all future cart operations related to this cart.

Name	Description
ItemTotal	Amount due for any number one kind of item.
PurchaseURL	The URL the customer should use to purchase the item. The URL includes the *AssociateTag* so that the Associate gets credit for the purchase.
Quantity	Number of individual items ordered.
SavedForLaterItems	Container object for items added to the cart that are currently unavailable.
SubTotal	Pre-tax and shipping subtotal of all items in the cart.
URLEncodedHMAC	A URL encoded version of the HMAC. The HMAC must be put in a URL and so it must be URL compliant. Use this value instead of the HMAC in all future requests related to this cart.

For more information about the parent elements of these tags, see the appropriate response group in Response Groups (p. 224).

Examples

Create a remote shopping cart and add multiple items of varying quantities to it.

```
http:// ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate Tag]&
Operation=CartCreate&
Item.1.ASIN=[ASIN]&
Item.1.Quantity=2&
Item.2.ASIN=[ASIN]&
Item.2.Quantity=7&
Item.3.ASIN=[ASIN]&
Item.3.Quantity=5
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request creates a remote shopping cart and adds three different items, 2 of the first item, 7 of the second item, and 5 of the third item. The response to this request is shown in, Response to Sample Request (p. 179).

Create a shopping cart by adding items not necessarily sold by Amazon.

```
http:// ecs.amazonaws.com/onca/xml
?Service=AWSECommerceService&
AWSAccessKeyId=[Access Key ID]&
AssociateTag=[Associate Tag]&
Operation=CartCreate&
Item.1.OfferListingId=[Offer Listing ID]&
Item.1.Quantity=2&
Item.2.OfferListingId=[Offer Listing ID]&
Item.2.Quantity=7&
Item.3.OfferListingId=[Offer Listing ID]&
```

```
Item.3.Quantity=5
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request creates a remote shopping cart and adds three different items that are not sold by Amazon: 2 of the first item, 7 of the second item, and 5 of the third item. OfferListingId returns items for sale by all sellers and merchants, including Amazon.

Sample Response

The following XML is a snippet of the full response to the first sample request.

```
<Cart>
  <Request>
    <IsValid>True</IsValid>
    <CartCreateRequest>
      <Items>
        <Item>
          <ASIN>B000062TU1</ASIN>
          <Quantity>2</Quantity>
        </Item>
      </Items>
    </CartCreateRequest>
  </Request>
  <CartId>102-5014548-4857758</CartId>
  <HMAC>O2p9hhZwJShnp6ZDWvZDO6FhpAI=</HMAC>
  <URLEncodedHMAC>O2p9hhZwJShnp6ZDWvZDO6FhpAI=</URLEncodedHMAC>

<PurchaseURL>https://www.amazon.com/gp/cart/aws-merge.html?cart-id=102-5014548-
4857758%26associate-
id=ws%26hmac=O2p9hhZwJShnp6ZDWvZDO6FhpAI=%26AWSAccessKeyId=1VMEXAMPLEW9C02</PurchaseURL>

  <SubTotal>
    <Amount>1994</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$19.94</FormattedPrice>
  </SubTotal>
  <CartItems>
      <SubTotal>
        <Amount>1994</Amount>
        <CurrencyCode>USD</CurrencyCode>
        <FormattedPrice>$19.94</FormattedPrice>
      </SubTotal>
    <CartItem>
      <CartItemId>U31XY1DHZEGCTB</CartItemId>
      <ASIN>B000062TU1</ASIN>
      <SellerNickname>Amazon.com, LLC</SellerNickname>
      <Quantity>2</Quantity>
     <Title>Harry Potter and the Sorcerer's Stone (Full Screen Edition) (Harry
 Potter 1)</Title>
      <ProductGroup>DVD</ProductGroup>
      <Price>
        <Amount>997</Amount>
        <CurrencyCode>USD</CurrencyCode>
        <FormattedPrice>$9.97</FormattedPrice>
      </Price>
      <ItemTotal>
```

```
        <Amount>1994</Amount>
        <CurrencyCode>USD</CurrencyCode>
        <FormattedPrice>$19.94</FormattedPrice>
      </ItemTotal>
    </CartItem>
  </CartItems>
</Cart>
```

Sample SavedForLaterItem XML Snippet

The following snippet from the XML response shows the addition of an item to a cart that is currently unavailable. On the retail web site, these items are displayed as "Saved For Later."

```
<SavedForLaterItems>
  <SubTotal>
    <Amount>1288</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$12.88</FormattedPrice>
  </SubTotal>
  <SavedForLaterItem>
    <CartItemId>ULI7S9IYFJHX0</CartItemId>
    <ASIN>B0009GZV4A</ASIN>
    <Quantity>2</Quantity>
    <Title>Mark VII Men's Short Sleeve Golf Shirts with Tri -Colored Stripe
Trim</Title>
    <ProductGroup>Apparel</ProductGroup>
    <Price>
      <Amount>644</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$6.44</FormattedPrice>
    </Price>
    <ItemTotal>
      <Amount>1288</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$12.88</FormattedPrice>
    </ItemTotal>
  </SavedForLaterItem>
</SavedForLaterItems>
```

Related Operations

- CartAdd (p. 180)
- CartClear (p. 184)
- CartGet (p. 192)
- CartModify (p. 196)

CartGet

Description

The `CartGet` operation enables you to retrieve the IDs, quantities, and prices of all of the items, including SavedForLater items in a remote shopping cart.

Because the contents of a cart can change for different reasons, such as availability, you should not keep a copy of a cart locally. Instead, use `CartGet` to retrieve the items in a remote shopping cart.

To retrieve the items in a cart, you must specify the cart using the *`CartId`* and *`HMAC`* values, which are returned in the `CartCreate` operation. A value similar to HMAC, *`URLEncodedHMAC`*, is also returned. This value is the URL encoded version of the *`HMAC`*. This encoding is necessary because some characters, such as + and /, cannot be included in a URL. Rather than encoding the *`HMAC`* yourself, use the *`URLEncodedHMAC`* value for the *`HMAC`* parameter.

`CartGet` does not work after the customer has used the *`PurchaseURL`* to either purchase the items or merge them with the items in their Amazon cart.

All `CartGet` requests must also include a value for *`AssociateTag`*; otherwise, the request will fail.

Availability

All locales.

Request Parameters

Name	Description	Required
`AssociateTag`	Alphanumeric token that uniquely identifies an Associate. This value is required. Type: String Default: None Valid Values: A valid *`AssociateTag`* that was used in the related `CartCreate` request. Constraint: You create a cart by using your AssociateTag, and you must include the same AssociateTag in all future requests related to that shopping cart.	Yes
`CartId`	Alphanumeric token returned by `CartCreate` that identifies a cart. Type: String Default: None Valid Values: Value returned by `CartCreate`	Yes
`CartItemId`	Alphanumeric token that uniquely identifies an item in a cart. Once an item, specified by an ASIN or OfferListingId, has been added to a cart, you must use the *`CartItemId`* to refer to it. The other identifiers will not work. Type: String Default: None Valid Values: Value returned by `CartCreate`	Yes

Name	Description	Required
HMAC	The Hash Message Authentication Code is an encrypted alphanumeric token that is used to authenticate requests. Type: String Default: None Valid Values: Value is calculated using request parameters, their values, a cryptographic function, and the Secret Key, which acts as the "key" for the function.	Yes
MergeCart	This parameter is deprecated. When a customer purchases items in a remote shopping cart, the items are added to the customer's Amazon retail shopping cart.	Deprecated
ResponseGroup	Specifies the types of values to return. You can specify multiple response groups in one request by separating them with commas. Type: String Default: Cart (p. 232) Valid Values: CartSimilarities (p. 237) | CartTopSellers (p. 236) | CartNewReleases (p. 235)	No

`CartGet` also accepts the parameters that all operations can use. For more information, see, Common Request Parameters (p. 221)

Response

Name	Description
Amount	Price of the item in terms of the lowest currency denomination, for example, pennies.
CartId	Alphanumeric token returned by `CartCreate` that identifies a cart.
CartItem	Container for CartItemId, ASIN, SellerNickname, Quantity, Title, ProductGroup, Price, ItemTotal
CartItems	Container for Subtotal and CartItem
CurrencyCode	Format for the display of the money
FormattedPrice	The price to display on the web site.
HMAC	Hash Message Authentication Code returned by `CartCreate` that identifies a cart. This is an encrypted alphanumeric token that is used to authenticate cart operations.
ItemTotal	Container for Amount, CurrencyCode, FormattedPrice. Cost of the item multiplied by the quantity ordered.
Price	Container for Amount, CurrencyCode, FormattedPrice. Price of a single item.

Name	Description
PurchaseURL	URL that customers should use to purchase the items in their cart. The URL includes the Associate's ID, called the Associate Tag. It is important to use the PurchaseURL otherwise the Associate will not get credit for the customer's purchase.
Subtotal	Container for Amount, CurrencyCode, and FormattedPrice. Subtotal is the total price of all of the items in a car butt not including tax or shipping.
URLEncodedHMAC	A URL encoded version of the *HMAC*. This encoding is necessary because some characters, such as + and /, cannot be included in a URL. The value of this parameter is a convenience function. Otherwise, you would have to URL encode the HMAC yourself.

For more information about the parent elements of these tags, see the appropriate response group in Response Groups (p. 224)

Examples

Retrieve all of the items in a specified cart:

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate Tag]&
Operation=CartGet&
CartId=[Cart ID]&
HMAC=[URL-encoded HMAC]
```

This operation retrieves all of the items in the specified remote shopping cart. Because the *HMAC* includes the *AssociateTag*, you must supply it in this request.

The response to this request is shown in, Response to Sample Request (p. 179).

Sample Response

The following XML is a snippet of the full response to the first sample request.

```
<CartId>002-9918938-1696046</CartId>
  <HMAC>b0ogzvivVYLXjSZ9WwoBRFesFYU=</HMAC>
<URLEncodedHMAC>b0ogzvivVYLXjSZ9WwoBRFesFYU=</URLEncodedHMAC>
<PurchaseURL>https://www.amazon.com/gp/cart/aws-merge.html?cart-id=002-9918938-
1696046%26associate-
id=ws%26hmac=b0ogzvivVYLXjSZ9WwoBRFesFYU=%26AWSAccessKeyId=1V293857EXAMPLEFW9C02</PurchaseURL>

  <SubTotal>
    <Amount>1994</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$19.94</FormattedPrice>
  </SubTotal>
  <CartItems>
    <SubTotal>
      <Amount>1994</Amount>
```

```
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$19.94</FormattedPrice>
    </SubTotal>
    <CartItem>
      <CartItemId>U3KYV0C66V3PAA</CartItemId>
      <ASIN>B000062TU1</ASIN>
      <SellerNickname>Amazon.com, LLC</SellerNickname>
      <Quantity>2</Quantity>
     <Title>Harry Potter and the Sorcerer's Stone (Full Screen Edition) (Harry
 Potter 1)</Title>
      <ProductGroup>DVD</ProductGroup>
      <Price>
        <Amount>997</Amount>
        <CurrencyCode>USD</CurrencyCode>
        <FormattedPrice>$9.97</FormattedPrice>
      </Price>
      <ItemTotal>
        <Amount>1994</Amount>
        <CurrencyCode>USD</CurrencyCode>
        <FormattedPrice>$19.94</FormattedPrice>
      </ItemTotal>
    </CartItem>
  </CartItems>
</Cart>
```

This response shows all of the items in the specified cart.

Related Operations

- CartAdd (p. 180)
- CartClear (p. 184)
- CartCreate (p. 187)
- CartModify (p. 196)

CartModify

Description

The `CartModify` operation enables you to change the quantity of items that are already in a remote shopping cart and move items from the active area of a cart to the SaveForLater area or the reverse.

To modify the number of items in a cart, you must specify the cart using the *`CartId`* and *`HMAC`* values that are returned in the CartCreate (p. 187) operation. A value similar to *`HMAC`*, *`URLEncodedHMAC`*, is also returned. This value is the URL encoded version of the *`HMAC`*. This encoding is necessary because some characters, such as + and /, cannot be included in a URL. Rather than encoding the *`HMAC`* yourself, use the *`URLEncodedHMAC`* value for the *`HMAC`* parameter.

You can use `CartModify` to modify the number of items in a remote shopping cart by setting the value of the *`Quantity`* parameter appropriately. You can eliminate an item from a cart by setting the value of the *`Quantity`* parameter to zero. Or, you can double the number of a particular item in the cart by doubling its *`Quantity`* . You cannot, however, use `CartModify` to add new items to a cart.

All `CartModify` requests must also include the value for *`AssociateTag`* that was used in the associated `CartCreate` (p. 187) request; otherwise, the request will fail.

Availability

All locales.

Request Parameters

Name	Description	Required
Action	The Action parameter is a child element of the Item parameter in both SOAP requests. Use the Action parameter to change cart items to move items to the Saved-For-Later area, or change Saved-For-Later (SaveForLater) items to the active cart area (MoveToCart). Type: String Default: None Valid Values: MoveToCart | SaveForLater	No
CartId	Alphanumeric token returned by CartCreate (p. 187) that identifies a cart. Type: String Default: None Valid Values: Value returned by CartCreate (p. 187).	Yes
CartItemId	Specifies an item to be modified in the cart where N is a positive integer between 1 and 10, inclusive. Up to ten items can be modified at a time.*CartItemId* is neither an ASIN nor an OfferListingId. It is, instead, an alphanumeric token returned by CartCreate (p. 187) and `CartAdd`. This parameter is used in conjunction with *Item.N.Quantity* to modify the number of items in a cart. See Item, that follows, for more information. Type: String Default: None Valid Values: Value returned by CartCreate (p. 187) or `CartAdd`.	Yes
HMAC	The Hash Message Authentication Code is an encrypted alphanumeric token that is used to authenticate requests. Type: String Default: None Valid Values: Value is calculated using request parameters, their values, a cryptographic function, and the Secret Key, which acts as the "key" for the function. Value returned by CartCreate (p. 187)	Yes
Item	Specifies the item to modify. Typical construction is Item.1.CartItemId=[ID]; Item.1.Quantity=[number].	Yes

Name	Description	Required
Items	Container for one or more Item objects. This is a SOAP only parameter. Valid Value: Item Constraint: Required and valid only for SOAP requests.	Conditional
MergeCart	This parameter is deprecated. When a customer purchases items in a remote shopping cart, the items are added to the customer's Amazon retail shopping cart.	Deprecated
Quantity	Specifies the revised number of items that the customer would like in the cart where N is a positive integer. This parameter is used in conjunction with *Item.N.CartItemId* to modify the number of items in a cart. The Quantity value is matched to the Item.N.CartItemId value by the index number, N. For example, Item.1.CartItemId is associated with Item.1.Quantity. Type: Positive integer Type: String Default: None Constraint: N is a positive integer between 0 and 999, inclusive. Required with REST queries.	Conditional
ResponseGroup	Specifies the types of values to return. You can specify multiple response groups in one request by separating them with commas. Type: String Default: Cart (p. 232) Valid Values:CartSimilarities (p. 237) | CartTopSellers (p. 236) | CartNewReleases (p. 235)	No

`CartModify` also accepts the parameters that all operations can use. For more information, see, Common Request Parameters (p. 221)

Response

Name	Description
CartId	Alphanumeric token returned by CartCreate (p. 187) that identifies a cart.
CartItemId	Specifies an item in a cart. *CartItemId* is neither an ASIN nor an OfferListingId. It is, instead, an alphanumeric token returned by CartCreate (p. 187) and `CartAdd`. Child of Item.

Name	Description
HMAC	Hash Message Authentication Code returned by CartCreate (p. 187) that identifies a cart. This is an encrypted alphanumeric token that is used to authenticate cart operations.
Item	For REST, a prefix for CartItemId and Quantity, both of which are used to specify the item to be modified, for example, item.1.CartItemId=1234abcd, item.1.Quantity=2 changes the quantity of item 1234abcd to 2.
Items	Container for Item when specifying more than one Item.
Quantity	The suffix used with Item to specify the number of items to be added to the cart. See Item.

For more information about the parent elements of these tags, see the appropriate response group in Response Groups (p. 224)

Examples

Increase the quantity of an item (Item.1) already in the cart and remove an item (Item.2).

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&amp;
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate ID]&
Operation=CartModify&
CartId=[Cart ID]&
HMAC=[HMAC]&amp;
Item.1.CartItemId=[Cart Item ID]&
Item.1.Quantity=10&
Item.2.CartItemId=[Cart Item ID]&
Item.2.Quantity=0
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Notice that items in the cart are referred to by *CartItemId*, not their *ASIN* or *OfferListingId* *CartItemId* is returned by CartCreate (p. 187), CartGet (p. 192), and CartAdd (p. 180). If you want to use CartModify to modify the items in a cart, you must refer to the items using their *CartItemId*. You use CartGet (p. 192) to retrieve the *CartItemId*s.

The response to this request is shown in, Response to Sample Request (p. 179).

Move an item (Item.1) to the SaveForLater area in the cart and reduce then number of items (Item.2) to 1.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&amp;
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate ID]&
Operation=CartModify&
CartId=[Cart ID]&
HMAC=[HMAC]&
Item.1.CartItemId=[Cart Item ID]&
Item.1.Action=SaveForLater
```

```
Item.2.CartItemId=[Cart Item ID]&
Item.2.Quantity=1
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response

The following XML is a snippet of the full response to the first sample request.

```
<CartModifyRequest>
  <CartId>103-8104506-5223005</CartId>
  <HMAC>uiFluRr5yj+GGL3SvWD1mFDyvP0=</HMAC>
  <Items>
    <Item>
      <CartItemId>UQIKL90WK14UD</CartItemId>
      <Quantity>10</Quantity>
    </Item>
  </Items>
</CartModifyRequest>
```

Notice that the quantity of the item was changed to 10 (from 5). The rest of the `CartModify` request is identical to the CartCreate (p. 187) response.

Related Operations

- CartAdd (p. 180)
- CartClear (p. 184)
- CartCreate (p. 187)
- CartGet (p. 192)

ItemLookup

Description

Given an Item identifier, the `ItemLookup` operation returns some or all of the item attributes, depending on the response group specified in the request. By default, `ItemLookup` returns an item's *`ASIN`*, *`Manufacturer`*, *`ProductGroup`*, and *`Title`* of the item.

`ItemLookup` supports many response groups, so you can retrieve many different kinds of product information, called item attributes, including product reviews, variations, similar products, pricing, availability, images of products, accessories, and other information.

To look up more than one item at a time, separate the item identifiers by commas.

Availability

All locales, however, the parameter support varies by locale.

Request Parameters

Name	Description	Required
Condition	Specifies an item's condition. If Condition is set to “All,” a separate set of responses is returned for each valid value of Condition. The default value is "New" (not "All"). So, if your request does not return results, consider setting the value to "All." When the value is "New," the ItemSearch Availability parameter cannot be set to "Available." Amazon only sells items that are "New." Type: String Default: New Valid Values: Used | Collectible | Refurbished, All	No
IdType	Type of item identifier used to look up an item. All *IdTypes* except *ASINx* require a *SearchIndex* to be specified. Type: String Default: ASIN Valid Values: SKU | UPC | EAN | ISBN (US only, when search index is Books). UPC is not valid in the CA locale.	No
IncludeReviewsSummary	When set to `true`, returns the reviews summary within the Reviews iframe. Type: Boolean Default: True Valid Values: `True` | `False`	No
ItemId	One or more (up to ten) positive integers that uniquely identify an item. The meaning of the number is specified by *IdType*. That is, if *IdType* is ASIN, the *ItemId* value is an ASIN. If *ItemId*is an ASIN, a search index cannot be specified in the request. Type: String Default: None Constraints: Must be a valid item ID. For more than one ID, use a comma-separated list of up to ten IDs.	Yes
MerchantId	An optional parameter you can use to filter search results and offer listings to only include items sold by Amazon. By default, the API will return items sold by various merchants including Amazon. Enter `Amazon` if you only want to see items sold by Amazon in the response. Type: String Valid Values: `Amazon`	No

Name	Description	Required
RelatedItemPage	This optional parameter is only valid when the *RelatedItems* response group is used. Each ItemLookup request can return, at most, ten related items. The RelatedItemPage value specifies the set of ten related items to return. A value of 2, for example, returns the second set of ten related items	No
RelationshipType	This parameter is required when the *RelatedItems* response group is used. The type of related item returned is specified by the *RelationshipType* parameter. Sample values include Episode, Season, and Tracks. For a complete list of types, go to Relationship Types (p. 128). Required when RelatedItems response group is used.	Yes
ReviewPage	Page of reviews returned by ItemLookup. There are 5 reviews per page. To examine reviews 6 through 10, for example, set *ReviewPage* to 2. Type: String Default: 1 Valid Values; Integer between 1 and 20, inclusive	No
ReviewSort	Specifies the order in which Reviews are sorted in the return. For more information, see Sorting Reviews Type: String Default: -SubmissionDate Valid Values: -HelpfulVotes | HelpfulVotes | -OverallRating | OverallRating | SubmissionDate	No
SearchIndex	The product category to search. Type: String Default: None Valid Values: A search index, for example, Apparel, Beauty, Blended, Books, and so forth. For a complete of search indices, see Search Indices (p. 443). Constraint:If *ItemId*is an ASIN, a search index cannot be specified in the request. Required for non-ASIN ItemIds.	Conditional
TagPage	Specifies the page of results to return. There are ten results on a page. Type: Integer Type: String Default: None Constraint: The maximum page number is 10.	No

Name	Description	Required
TagsPerPage	The number of tags to return that are associated with a specified item. Type: Integer Type: String Default: None	No
TagSort	Specifies the sorting order for the results. Type: String Default: - Usages Valid Values: • FirstUsed-Sort by the date the item was first used. - FirstUsed • LastUsed-Sort by the date the item was last used. - LastUsed • Name-Alphabetical, A-Z. - Name • Usages-Items least used to most used To sort items in descending order, prefix the values with a negative sign (-).	No
TruncateReviewsAt	By default, reviews are truncated to 1000 characters within the Reviews iframe. To specify a different length, enter the value. To return complete reviews, specify `0`. Type: Integer Default: 1000 Constraints: Must be a positive integer or 0 (returns entire review)	No
VariationPage	Page number of variations returned by `ItemLookup`. By default, `ItemLookup` returns all variations. Use *`VariationPage`* to return a subsection of the response. There are 10 variations per page. To examine offers 11 trough 20, for example, set *`VariationPage`* to 2. Type: String Default: All Valid Values: Integer between 1 and 150, inclusive	No

Name	Description	Required
ResponseGroup	Specifies the types of values to return. You can specify multiple response groups in one request by separating them with commas. Valid Values: Accessories (p. 224) | BrowseNodes (p. 230) | EditorialReview (p. 241) | Images (p. 242) | ItemAttributes (p. 245) | ItemIds (p. 248) | Large (p. 250) | Medium (p. 255) | OfferFull (p. 263) | Offers (p. 267) | PromotionSummary (p. 273) | OfferSummary (p. 271)| RelatedItems (p. 276) | Reviews (p. 280) | SalesRank (p. 282) | Similarities (p. 288) | Tracks (p. 292) | VariationImages (p. 297) | Variations (p. 294) (US only) | VariationSummary (p. 304)	No

`ItemLookup` also accepts the parameters that all operations can use. For more information, see, Common Request Parameters (p. 221)

Response

Name	Description
ASIN	Amazon Standard Identification Number, which is an alphanumeric token assigned by Amazon to an item that uniquely identifies it.
Item	Container for information about the item, including *ASIN*, *Title*, *ProductGroup*. and *Manufacturer*.
ItemAttributes	Container for information about an item, including *Title*, *ProductGroup*. and *Manufacturer*.
Items	Container for one or more Item(s).
Manufacturer	Name of the company that manufactured the item.
ProductGroup	Category of the item, for example, “Book,” “DVD.” You can find the complete list of product groups by going to www.amazon.com and clicking the tab that says "See All 32 Product Categories."
Title	Title of the item.

For more information about the parent elements of these tags, see the appropriate response group in Response Groups (p. 224)

Examples

The following request returns the information associated with *ItemId* B00008OE6I.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B00008OE6I
```

```
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The response to this request is shown in, Response to Sample Request (p. 179).

The following request returns an offer for a refurbished item that is not sold by Amazon.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B00008OE6I&
IdType=ASIN&
ResponseGroup=OfferFull&
Condition=All&
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

In the following request, the *ItemId* is an SKU, which requires that you also specify the *IdType*.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=[SKU]&
IdType=SKU
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

In the following request, the *ItemId* is a UPC, which requires that you also specify the *SearchIndex* and *ItemType*.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=[UPC]&
SearchIndex=Books&
&IdType=UPC
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

In the following request, the *ItemId* is an EAN, which requires that you also specify the *SearchIndex* and *ItemType*.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=[EAN]&
SearchIndex=Electronics&
IdType=EAN
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Use the BrowseNodes (p. 230) response group to find the browse node of an item.

Use the Tracks (p. 292) response group to find the track, title, and number for each track on each CD in the response.

Use the Similarities (p. 288) response group to find the ASIN and Title for similar products returned in the response.

Use the Reviews (p. 280) response group to find reviews written by customers about an item, and the total number of reviews for each item in the response.

Use the OfferSummary (p. 271) response group to find the number of offer listings and the lowest price for each of the offer listing condition classes, including New, Used, Collectible, and Refurbished.

Use the Accessories (p. 224) response group to find the a list of accessory product ASINs and Titles for each product in the response that has accessories.

The following requests an iframe that contains customer reviews for the specified item.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=0316067938&
ResponseGroup=Reviews&
TruncateReviewsAt="256"&
IncludeReviewsSummary="False"&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response

The following code snippet is a response to the first request. It shows all of the item attributes that are returned by default.

```
<Items>
  <Request>
  <IsValid>True</IsValid>
  <ItemLookupRequest>
    <ItemId>B00008OE6I</ItemId>
  </ItemLookupRequest>
  </Request>
  <Item>
    <ASIN>B00008OE6I</ASIN>
    <ItemAttributes>
    <Manufacturer>Canon</Manufacturer>
    <ProductGroup>Photography</ProductGroup>
    <Title>Canon PowerShot S400 4MP Digital Camera w/ 3x Optical Zoom</Title>
   </ItemAttributes>
  </Item>
</Items>
```

The following code snippet is part of a response for an iframe that contains customer reviews.

```
<ItemLookupResponse>
 <Items>
  <Item>
   <ASIN>0316067938</ASIN>
   <CustomerReviews>
    <IFrameURL>

http://www.amazon.com/reviews/iframe?akid=1C36YNE9WHA6T34ADPR2&asin=0316067938&exp=2010-
09-
02T17%3A54%3A07Z&linkCode=xm2&summary=0&tag=ws&truncate=256&v=2&sig=rhBGqRiaQ3L78ycgE5y30u8bF6zXsK3HL%2F2KpNMJns0%3D

    </IFrameURL>
   </CustomerReviews>
  </Item>
 </Items>
</ItemLookupResponse>
```

Related Operations

- ItemSearch (p. 207)

ItemSearch

Description

The `ItemSearch` operation returns items that satisfy the search criteria, including one or more search indices.

`ItemSearch` returns up to ten search results at a time. When *`condition`* equals "All," `ItemSearch` returns up to three offers per condition (if they exist), for example, three new, three used, three refurbished, and three collectible items. Or, for example, if there are no collectible or refurbished offers, `ItemSearch` returns three new and three used offers.

Because there are thousands of items in each search index, `ItemSearch` requires that you specify the value for at least one parameter in addition to a search index. The additional parameter value must reference items within the specified search index. For example, you might specify a browse node (BrowseNode is an `ItemSearch` parameter), Harry Potter Books, within the Books product category. You would not get results, for example, if you specified the search index to be Automotive and the browse node to be Harry Potter Books. In this case, the parameter value is not associated with the search index value.

The *`ItemPage`* parameter enables you to return a specified page of results. The maximum *`ItemPage`* number that can be returned is 10. An error is returned if you try to access higher numbered pages. If you do not include *`ItemPage`* in your request, the first page will be returned by default. There can be up to ten items per page.

`ItemSearch` is the operation that is used most often in requests. In general, when trying to find an item for sale, you use this operation.

Availability

All locales.

Request Parameters

`ItemSearch` has a lot of parameters. Not all of them pertain, however, to all search indices. For example, when the search index is apparel, it would be inappropriate to use the *`Actor`* parameter. As a result, each search index can use only a subset of all of the parameters. For a complete list of the `ItemSearch` parameters that can be used with a specific search index in a specific locale, refer to Search Index and ItemSearch Parameter Combinations (p. 392).

The parameters that apply to the largest number of search indices are shown in the following table.

Parameter	Valid Search Indices
`BrowseNode`	All but All, Blended
`Condition`	All but All, Blended
`Keywords`	All
`MaximumPrice`	All but All, Blended
`MinimumPrice`	All but All, Blended
`Title`	All but All, Blended

`ItemSearch` requires that you specify a search index and at least one of the following parameters:

• Actor • Artist • AudienceRating • Author • Brand • BrowseNode • City	• Composer • Conductor • Director • Keywords • Manufacturer • MusicLabel • Neighborhood	• Orchestra • Power • Publisher • Title

Name	Description	Required
`Actor`	Name of an actor associated with the item. You can enter all or part of the name. Type: String Default: None	No
`Artist`	Name of an artist associated with the item. You can enter all or part of the name. Type: String Default: None	No

Name	Description	Required
AudienceRating	Movie ratings based on MPAA ratings or age, depending upon the locale. You may specify one or more values in a comma-separated list in a REST request or by using multiple elements in a SOAP request. Type: String. Type: String Default: None Valid Values: See Movie Ratings by Locale (p. 215), which follows this table.	No
Author	Name of an author associated with the item. You can enter all or part of the name. Type: String Default: None	No
Availability	Enables `ItemSearch` to return only those items that are available. This parameter must be used in combination with a merchant ID and *`Condition`*. For more information, see Availability Parameter (p. 90), which follows this table.When Availability is set to "Available," the Condition parameter cannot be set to "New." Type: String Default: None Valid Values: Available	Yes
Brand	Name of a brand associated with the item. You can enter all or part of the name. Type: String, for example, Timex, Seiko, Rolex. Type: String Default: None	No
BrowseNode	Browse nodes are positive integers that identify product categories, for example, Literature & Fiction: (17), Medicine: (13996), Mystery & Thrillers: (18), Nonfiction: (53), Outdoors & Nature: (290060). Type: String Default: None Valid Values: Positive integer.	No

Name	Description	Required
City	Name of a city associated with the item. You can enter all or part of the name. This parameter only works in the US locale. Type: String Default: None Valid Values: Chicago | New York | San Francisco | Seattle | Washington, D.C.	No
Composer	Name of an composer associated with the item. You can enter all or part of the name. Type: String Default: None	No
Condition	Use the *Condition* parameter to filter the offers returned in the product list by condition type. By default, *Condition* equals "New". If you do not get results, consider changing the value to "All. When the Availability parameter is set to "Available," the Condition parameter cannot be set to "New." ItemSearch returns up to ten search results at a time. When *condition* equals "All," ItemSearch returns up to three offers per condition (if they exist), for example, three new, three used, three refurbished, and three collectible items. Or, for example, if there are no collectible or refurbished offers, ItemSearch returns three new and three used offers. Type: String Default: New Valid Values: Used | Collectible | Refurbished | All	No
Conductor	Name of a conductor associated with the item. You can enter all or part of the name. Type: String Default: None	No
Director	Name of a director associated with the item. You can enter all or part of the name. Type: String Default: None	No

Name	Description	Required
IncludeReviewsSummary	When set to `true`, returns the reviews summary within the Reviews iframe. Type: Boolean Default: True Valid Values: `True` | `False`	No
ItemPage	Retrieves a specific page of items from all of the items in a response. Up to ten items are returned on a page unless *Condition* equals "All." In that case, `ItemSearch` returns up to three results per *Condition*, for example, three new, three used, three refurbished, and three collectible items. Or, for example, if there are no collectible or refurbished items being offered, `ItemSearch` returns three new and three used items. If you do not include *ItemPage* in your request, the first page is returned. The total number of pages of items found is returned in the *TotalPages* response tag. Valid Values: 1 to 10 (1 to 5 when the search index = "All") Type: String Default: None	No
Keywords	A word or phrase associated with an item. The word or phrase can be in various product fields, including product title, author, artist, description, manufacturer, and so forth. When, for example, the search index equals "MusicTracks," the *Keywords* parameter enables you to search by song title. If you enter a phrase, the spaces must be URL-encoded as %20. Type: String Default: None	No
Manufacturer	Name of a manufacturer associated with the item. You can enter all or part of the name. Type: String Default: None	No
MaximumPrice	Specifies the maximum price of the items in the response. Prices are in terms of the lowest currency denomination, for example, pennies. For example, 3241 represents $32.41. Type: String Default: None Valid Values: Positive integer	No

Name	Description	Required
MerchantId	An optional parameter you can use to filter search results and offer listings to only include items sold by Amazon. By default, Product Advertising API returns items sold by various merchants including Amazon. Use the `Amazon` to limit the response to only items sold by Amazon. Type: String Valid Values: `Amazon`	No
MinimumPrice	Specifies the minimum price of the items to return. Prices are in terms of the lowest currency denomination, for example, pennies, for example, 3241 represents $32.41. Type: String Default: None Valid Values: Positive integer	No
Neighborhood	Name of a neighborhood You can enter all or part of the name. The neighborhoods are located in one of the valid values for *City*. Type: String, for example, Capitol Hill, Arlington, and North Beach. Type: String Default: None	No
Orchestra	Name of an orchestra associated with the item. You can enter all or part of the name. Type: String Default: None	No
Power	Performs a book search using a complex query string. Only works when the search index is set equal to "Books." Valid Values: See, Power Searches (p. 96) following this table. Type: String Default: None	No
Publisher	Name of a publisher associated with the item. You can enter all or part of the name. Type: String Default: None	No

Name	Description	Required
RelatedItemPage	This optional parameter is only valid when the *RelatedItems* response group is used. Each ItemLookup request can return, at most, ten related items. The RelatedItemPage value specifies the set of ten related items to return. A value of 2, for example, returns the second set of ten related items.	No
RelationshipType	This parameter is required when the *RelatedItems* response group is used. The type of related item returned is specified by the *RelationshipType* parameter. Sample values include Episode, Season, and Tracks. A complete list of values follows this table. Constraint: Required when RelatedItems response group is used	Conditional
ReviewSort	Sorts reviews based on the value of the parameter. Type: String Default: None Valid Values: -HelpfulVotes, HelpfulVotes, -OverallRating, OverallRating, Rank, -Rank, -SubmissionDate, SubmissionDate	No
SearchIndex	The product category to search. Many ItemSearch parameters are valid with only specific values of *SearchIndex*. Type: String Default: None Valid Values: A search index, for example, Apparel, Beauty, Blended, Books, and so forth. For Blended searches, go to Blended Searches (p. 95). For a complete of search indices, see Search Indices by Locale (p. 392).	No
Sort	Means by which the items in the response are ordered. Type: String Default: None Valid Values: Valid values vary significantly by search index. For a list of valid values, see ItemSearch Sort Values by Locale (p. 325).	No
TagPage	Specifies the page of results to return. There are ten results on a page. The maximum page number is 10. Type: Integer Type: String Default: None	No

Name	Description	Required
TagsPerPage	The number of tags to return that are associated with a specified item. Type: Integer Type: String Default: None	No
TagSort	Specifies the sorting order for the results. Type: String Default: - Usages Valid Values: • FirstUsed-Sort by the date the item was first used. - FirstUsed • LastUsed-Sort by the date the item was last used. - LastUsed • Name-Alphabetical, A-Z. - Name • Usages-Items least used to most used To sort items in descending order, prefix the values with a negative sign (-).	No
Title	The title associated with the item. You can enter all or part of the title. *Title* searches are a subset of *Keyword* searches. If a *Title* search yields insufficient results, consider using a *Keywords* search. Type: String Default: None	No
TruncateReviewsAt	By default, reviews are truncated to 1000 characters within the Reviews iframe. To specify a different length, enter the value. To return complete reviews, specify `0`. Type: Integer Default: 1000 Constraints: Must be a positive integer or 0 (returns entire review)	No

Name	Description	Required
VariationPage	Retrieves a specific page of variations returned by `ItemSearch`. By default, `ItemSearch` returns all variations. Use *VariationPage* to return a subsection of the response. There are 10 variations per page. To examine offers 11 trough 20, for example, set *VariationPage* to 2. The total number of pages is returned in the *TotalPages* element. Type: String Default: None Valid Values: Positive integer	No
ResponseGroup	Specifies the types of values to return. You can specify multiple response groups in one request by separating them with commas. Type: String Default: Small (p. 289) Valid Values: Accessories (p. 224) | BrowseNodes (p. 230) | EditorialReview (p. 241) | ItemAttributes (p. 245) | ItemIds (p. 248) | Large (p. 250) | Medium (p. 255) | OfferFull (p. 263) | Offers (p. 267) | OfferSummary (p. 271) | Reviews (p. 280) | RelatedItems (p. 276) | SearchBins (p. 283) | Similarities (p. 288) | Tracks (p. 292) | Variations (p. 294) | VariationSummary (p. 304) |	No

`ItemSearch` also accepts the parameters that all operations can use. For more information, see, Common Request Parameters (p. 221)

Movie Ratings Vary by Locale

Movie rating values captured in the *AudienceRating* parameter, vary by locale. The following table shows the valid values of *AudienceRating*.

Locale	AudienceRating Values
CA	G, PG, PG-13, R, NC-17, NR, Unrated, Family Viewing
DE	6, 12, 16
FR	PG, 12, 16, 18
US	G, PG, PG-13, R, NC-17, NR, Unrated

Response

Name	Description
ASIN	Amazon Standard Identification Number, which is an alphanumeric token assigned by Amazon to an item that uniquely identifies it.

Name	Description
Item	Container for item information, including ASIN and ItemAttributes.
ItemAttributes	Container for information about an item, including Manufacturer, ProductGroup, and Title.
Manufacturer	Item's manufacturer.
MoreSearchResultsURL	The URL where the complete search results are displayed. The URLs provided in the search results are the exact ones that you should use when you link back to Amazon.com. They are tagged with your Associate tag and contain other tracking information to increase your hourly request limit as the sales that you generate increase.
ProductGroup	Product category; similar to search index.
Title	Item's title.
TotalPages	Total number of pages in response. There are ten items per page.
TotalResults	Total number of items found.

For more information about the parent elements of these tags, see the appropriate response group in Response Groups (p. 224).

Examples

Use the search index, Toys, and the parameter, *Keywords*, to return information about all toy rockets sold in by Amazon.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Keywords=Rocket&
SearchIndex=Toys
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

The response to this request is shown in, Response to Sample Request (p. 179).

Use a blended search to look through multiple search indices for items that have "Mustang" in their name or description. A blended search looks through multiple search indices at the same time. For more information, see Blended Searches. (p. 95)

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Keywords=Mustang&
SearchIndex=Blended
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Use the *Availability* parameter to only return shirts that are available:

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=All&
Availability=Available&
SearchIndex=Apparel&
Keywords=Shirt
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Set the search index to MusicTracks and *Keywords* to the title of a song to find a song title.

Use the BrowseNodes (p. 230) response group to find the browse node of an item.

Use the Variations (p. 40) response group and the *BrowseNode* parameter to find all of the variations of a parent browse node.

Sample Response

The following XML is a snippet of the full response to the first sample request.

```
<TotalResults>372</TotalResults>
<TotalPages>38</TotalPages>
<Item>
  <ASIN>B00021HBN6</ASIN>
  <ItemAttributes>
    <Manufacturer>Radio Flyer</Manufacturer>
    <ProductGroup>Toy</ProductGroup>
    <Title>Radio Flyer Retro Rocket</Title>
  </ItemAttributes>
  </Item>
  <Item>
  <ASIN>B0007MZV3C</ASIN>
  <ItemAttributes>
  <Manufacturer>Razor USA LLC</Manufacturer>
  <ProductGroup>Toy</ProductGroup>
  <Title>Razor Dirt Rocket MX350 Bike</Title>
  </ItemAttributes>
</Item>
```

The *TotalResults* and *TotalPages* tags indicate the number of items found and the number of pages those items are on. Use *TotalPages* with any of the page parameters, such as *ReviewsPage*, to select the page of results to view. Typically, there are ten results on a page.

Related Operations

- ItemLookup (p. 200)

SimilarityLookup

Description

The `SimilarityLookup` operation returns up to ten products per page that are similar to one or more items specified in the request. This operation is typically used to pique a customer's interest in buying something similar to what they've already ordered.

If you specify more than one item, SimilarityLookup returns the intersection of similar items each item would return separately. Alternatively, you can use the *`SimilarityType`* parameter to return the union of items that are similar to any of the specified items. A maximum of ten similar items are returned; the operation does not return additional pages of similar items. If there are more than ten similar items, running the same request can result in different answers because the ten that are included in the response are picked randomly. The results are picked randomly only when you specify multiple items and the results include more than ten similar items.

When you specify multiple items, it is possible for there to be no intersection of similar items. In this case, the operation returns the following error:

```
<Error>
  <Code>AWS.ECommerceService.NoSimilarities</Code>
  <Message>There are no similar items for this ASIN: B00006WREH.</Message>
</Error>
```

This result is very often the case if the items belong to different search indices. The error can occur, however, even when the items share the same search index.

Similarity is a measurement of similar items purchased, that is, customers who bought X also bought Y and Z. It is not a measure, for example, of items viewed, that is, customers who viewed X also viewed Y and Z.

Items returned can be filtered by:

- Condition—Describes the status of an item. Valid values are All, New (default), Used, Refurbished or Collectible. When the Availability parameter is set to "Available," the Condition parameter cannot be set to "New."

Examples

Return items that are similar to a list of items.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=SimilarityLookup&ItemId=ASIN1,ASIN2,ASIN3
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request returns the intersection of the similarities for each ASIN. The response to this request is shown in Response to Sample Request (p. 179).

Return up to ten items that are similar to any of the ASINs specified.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=SimilarityLookup&ItemId=ASIN1,ASIN2,ASIN3&
SimilarityType=Random
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

This request returns the union of items that are similar to all of the ASINs specified. Only ten items can be returned and those are picked randomly from all of the similar items. Repeating the operation could produce different results.

Request Parameters

Name	Description	Required
Condition	Specifies an item's condition. If Condition is set to “All,” a separate set of responses is returned for each valid value of Condition. Type: String Default: New Valid Values: All | Collectible | Refurbished | Used	No
ItemId	Specifies the item you want to look up. An ItemId is an alphanumeric identifier assigned to an item. You can specify up to ten ItemIds separated by commas. Type: String Default: None Valid Values: ASIN	Yes
MerchantId	An optional parameter that can be used to filter search results and offer listings to only include items sold by Amazon. By default, Product Advertising API returns items sold by various merchants including Amazon. Enter "Amazon" to return only items sold by Amazon. Type: String Valid Values: `Amazon`	No

Name	Description	Required
SimilarityType	"Intersection" returns the intersection of items that are similar to all of the ASINs specified. "Random" returns the union of items that are similar to all of the ASINs specified. Only ten items are returned. So, if there are more than ten similar items found, a random selection from the group is returned. For this reason, running the same request multiple times can yield different results. Type: String Default: Intersection Valid Value:Random	No
ResponseGroup	Specifies the types of values to return. You can specify multiple response groups in one request by separating them with commas. Type: String Default: Request (p. 278) | Small (p. 289) Valid Values: Accessories (p. 224) | BrowseNodes (p. 230) | EditorialReview (p. 241) | Images (p. 242) | Large (p. 250) | ItemAttributes (p. 245) | ItemIds (p. 248) | Medium (p. 255) | Offers (p. 267) | OfferSummary (p. 271) | PromotionSummary (p. 273) | Reviews (p. 280) | SalesRank (p. 282) | Similarities (p. 288) | Tracks (p. 292) | Variations (p. 294) | VariationSummary (p. 304) |	No

`SimilarityLookup` also accepts the parameters that all operations can use. For more information, see, Common Request Parameters (p. 221).

Sample Response

The following XML is a snippet of the full response to the first sample request.

```
<Item>
  <ASIN>B0009VX8XQ</ASIN>
  <ItemAttributes>
    <ProductGroup>Apparel</ProductGroup>
    <Title>Mark VII Men's Short Sleeve Herringbone Knit Golf Shirt</Title>
  </ItemAttributes>
</Item>
<Item>
  <ASIN>B0009VX8VI</ASIN>
  <ItemAttributes>
    <ProductGroup>Apparel</ProductGroup>
    <Title>Mark VII Men's Short Sleeve Knit Golf Shirt</Title>
  </ItemAttributes>
</Item>
```

This response shows that two items were similar to the ASINs submitted. The information included with each item is sufficient to display the item for sale.

Response

Name	Description
ASIN	An alphanumeric token that uniquely identifies an item for sale.
Item	Container for information about an item, including, ItemAttributes, and ASIN.
ItemAttributes	Container for information describing an item, including, ProductGroup and Title.
ProductGroup	Category to which an item belongs. Similar to Search Index.
Title	Name of the item for sale.

For more information about the parent elements of these tags, see the appropriate response group in Response Groups (p. 224).

Related Operations

ItemSearch (p. 207) using the Similarities (p. 288) response group

Common Request Parameters

The following table describes the request parameters that all Product Advertising API operations can use.

Parameter	Definition	Req'd?
AssociateTag	An alphanumeric token that uniquely identifies an Associate. This token is the means by which Amazon identifies the Associate to credit for a sale. If a request is made without identifying an Associate, Associate fees are not paid by Amazon. If the AssociateTag is included in the `CartCreate` request, the value for AssociateTag is automatically included in the PurchaseURL, which is returned by `CartCreate`. To obtain an Associate Tag, go to https://affiliate-program.amazon.com. For more information, see Associate Tag (p. 223). Valid Values: An alphanumeric token distributed by Amazon that uniquely identifies an Associate. Use this value in all requests to receive credit for the customer's purchase.	Yes
AWSAccessKeyId	An alphanumeric token that uniquely identifies a seller. To get an *AWSAccessKeyId*, go to http://aws.amazon.com. Valid Value: Access Key ID distributed by Amazon.	Yes

Parameter	Definition	Req'd?
ContentType	Specifies the format of the content in the response. Generally, *ContentType* should only be changed for REST requests when the Style parameter is set to an XSLT stylesheet. For example, to transform your Product Advertising API response into HTML, set *ContentType* to text/html. See *Style*. Default: text/xml Valid Value: text/xml, text html	No
MarketplaceDomain	Specifies the Marketplace Domain within which to make the request. Do not use this parameter unless making a request against one of the domains specified in the list of valid values. Valid Values: www.javari.jp | www.javari.co.uk | www.endless.com	No
MerchantId	An optional parameter that can be used to filter search results and offer listings to only include items sold by Amazon. By default, the API will return items sold by various merchants including Amazon. Valid values: • Amazon Only items sold by Amazon are included in the response.	No
Operation	Specifies the Product Advertising API operation to execute. Valid Value: An Product Advertising API operation, for example, `ItemLookup`	Yes
Service	Specifies the Product Advertising API service. There is only one value for all Product Advertising API operations. Valid Value: AWSECommerceService	Yes
Style	Controls the format of the data returned in Product Advertising API responses. *Style* only pertains to REST requests. Set this parameter to "XML," the default, to generate a pure XML response. Set this parameter to the URL of an XSLT stylesheet to have Product Advertising API transform the XML response. See *ContentType* Default: XML Valid Values: URL of an XSLT stylesheet .	No

Parameter	Definition	Req'd?
Validate	Prevents an operation from executing. Set the *Validate* parameter to True to test your request without actually executing it. When present, *Validate* must equal True; the default value is False. If a request is not actually executed (Validate=True), only a subset of the errors for a request may be returned because some errors (for example, no_exact_matches) are only generated during the execution of a request. Default: False Valid Values: True, False	No
Version	The version of the Product Advertising API software and WSDL to use. By default, the 2005-10-05 version is used. Alternately, specify a software version, such as 2011-08-01. For a list of valid version numbers, refer to the Product Advertising API Release Notes. Note that the latest version of Product Advertising API is not used by default. Default: 2005-10-05 Valid Values: Valid WSDL version date, for example, 2011-08-01.	No
XMLEscaping	Specifies whether responses are XML-encoded in a single pass or a double pass. By default, *XMLEscaping* is Single, and Product Advertising API responses are encoded only once in XML. For example, if the response data includes an ampersand character (&), the character is returned in its regular XML encoding (&). If *XMLEscaping* is Double, the same ampersand character is XML-encoded twice (&). The Double value for *XMLEscaping* is useful in some clients, such as PHP, that do not decode text within XML elements. Default: Single Valid Values: Single, Double	No

AssociateTag

The *AssociateTag* parameter is a required parameter in Product Advertising API requests. Once a cart is associated with an Associate Tag, you must use it in every other cart operation related to that shopping cart. Otherwise, you will receive an error.

If you do not include an Associate Tag in the `CartCreate` request, your request will fail.

AssociateTag is locale-specific, that is, for each locale in which you want to earn Associate revenue, you must get an Associate Tag. If you try to use a US-locale Associate Tag, for example, in the JP locale, you will not earn Associate revenue.

Be careful when specifying an AssociateTag because errors are not returned for incorrect values.

Response Groups

Response groups help target the information returned. Each operation can only use a subset of all of the available response groups. The following sections explain, for each response group, the:

- Description of the response group.
- Descriptions of the XML elements returned by the response group.
- The ancestry of the element presented using X-path notation.
- Child response groups, which are the response groups that are included in a response group. For example, the Large response group includes the results returned by the ItemAttributes response group.
- Parent response groups, which are response groups that include in the response group. For example, the ItemAttributes response group is included in the Large response group, so, the Large response group is a parent of the ItemAttributes response group.

Product Advertising API includes the following response groups:

Accessories Response Group

The Accessories response group returns up to five ASINs and titles of accessories associated with items in the response. For example, if you performed an ItemLookup (p. 200) of a specific ASIN, say, a camera, and included in the request the Accessories response group, the response would include, by default, the titles and ASINs of up to the first ten accessories associated with the ASIN. The accessories for the camera might include a camera case, lens, filters, and so forth.

Relevant Operations

Operations that can use this response group include:

- SimilarityLookup (p. 218)
- ItemLookup (p. 200)

- ItemSearch (p. 207)

Response Elements

The following table describes the elements returned by Accessories.

- ASIN (p. 308)
- Title (p. 322)
- TotalPages (p. 323)
- TotalResults (p. 323)

Parent Response Group

The following response groups are parent response groups of Accessories (p. 224).

- Large (p. 250)

Child Response Group

The following response groups are child response groups of Accessories (p. 224).

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate ID]&
Operation=ItemLookup&
IdType=ASIN&
ItemId=B00008OE6I&
ResponseGroup=Accessories&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Accessories.

```
<Item>
  <ASIN>B00008OE6I</ASIN>
  <Accessories>
    <Accessory>
      <ASIN>B00003G1RG</ASIN>
      <Title>Viking 128 MB CompactFlash Card (CF128M)</Title>
    </Accessory>
    <Accessory>
```

```
      <ASIN>B00004WCCT</ASIN>
      <Title>Canon Soft Leather Case for Canon Digital ELPH Cameras
      (Black)</Title>
    </Accessory>
    <Accessory>
      <ASIN>B000051408</ASIN>
      <Title>SimpleTech STI-CF/128 128MB CompactFlash Card</Title>
    </Accessory>
  </Accessories>
```

AlternateVersions Response Group

The AlternateVersions response group returns all of the available media formats for a book title. Sample formats include Paperback, Audio CD, Audio Cassette, and Hardcover. This response group works only with the Books and ForeignBooks product categories, that is, for ItemSearch and ItemLookup, the search index must be Books or, in non-US locales, ForeignBooks, and, for ItemLookup, the specified item must be a book or foreign book.

The following list shows the complete set of formats that AlternateVersions can return.

• 3.5" disk • Accessory • Audio CD • Audio Download • Audio Reel Tape • Bargain Book • Bath Book • Board book • Calendar • Card Book • Cards • Cassette audio • CD-ROM • Comic • Diary • Digital • Diskette • DVD Audio • DVD-ROM • Film	• Foam Book • Game • Gift • Hardcover • Hardcover Comic • Journal • Laser Disc • Leather Bound • Library Binding • Loose Leaf • Map • Mass Market Paperback • Microfiche • Mini-Disc • Misc. • Misc. Supplies • MP3 CD • Paperback • Perfect • Personal Computers	• Plastic Comb • Pop-Up • Poster • Rag Book • Ring-bound • Roughcut • School & Library Binding • Sheet music • Slide • Software • Spiral-bound • Stationery • Textbook Binding • Turtleback • Unbound • Unknown Binding • Wall Chart • Workbook

Relevant Operations

Operations that can use this response group include:

- ItemSearch, when the search index is Books or, in non-US locales, ForeignBooks
- ItemLookup when the specified item is a book or foreign book and, if a search index is specified, the value must be Books or, in non-US locales, ForeignBooks

Response Elements

The following table describes and shows the parentage of the elements returned by AlternateVersions.

- AlternateVersion (p. 307)
- ASIN (p. 308)
- Binding (p. 309)
- Title (p. 322)

AlternateVersions also returns the elements that all response groups return, which described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of AlternateVersions.

- None

Child Response Group

The following response groups are children response groups of AlternateVersions.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by AlternateVersions.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
Operation=ItemSearch&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
SearchIndex=Books&
Keywords=potter&
ResponseGroup=AlternateVersions&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by AlternateVersions.

```
<AlternateVersions>
  <AlternateVersion>
    <ASIN>030728364X</ASIN>
    <Title>Harry Potter and the Half-Blood Prince (Book 6)</Title>
    <Binding>Audio Cassette</Binding>
  </AlternateVersion>
  <AlternateVersion>
    <ASIN>0307283658</ASIN>
```

```
    <Title>Harry Potter and the Half-Blood Prince (Book 6)</Title>
    <Binding>Audio CD</Binding>
  </AlternateVersion>
  <AlternateVersion>
    <ASIN>0439785960</ASIN>
    <Title>
   Harry Potter And The Half-Blood Prince (Harry Potter, Book 6) (Harry Potter)

    </Title>
    <Binding>Paperback</Binding>
  </AlternateVersion>
  <AlternateVersion>
    <ASIN>0439786770</ASIN>
    <Title>
    Harry Potter and the Half-Blood Prince (Book 6) [LIBRARY EDITION]
    </Title>
    <Binding>Library Binding</Binding>
  </AlternateVersion>
```

BrowseNodeInfo Response Group

For a given browse node ID, the BrowseNodeInfo response group returns the browse node name and ID of the child and parent browse nodes.

One application of this information is to use the child and parent browse nodes to traverse the browse node hierarchy to either refine a search (child nodes) or to generalize a search (parent nodes). You do that using BrowseNodeLookup (p. 177) one or more times. Refining a search enables you to return more highly targeted results. Generalizing a response enables you to find what product category a browse node belongs to.

This response group is similar to the BrowseNodes (p. 230) response group. The difference is that the BrowseNodes response group is used with operations, including ItemSearch (p. 207), ItemLookup (p. 200), and SimilarityLookup (p. 218), that are based on item attributes, search indices, and lists, and potentially return multiple items. BrowseNodeInfo can only be used with BrowseNodeLookup (p. 177) and the search is always keyed on a browse node ID.

Relevant Operations

Operations that can use this response group include:

- BrowseNodeLookup (p. 177)

Response Elements

The following table describes the elements returned by BrowseNodeInfo.

- BrowseNodeId (p. 309)
- Name (p. 317)

As you can see from this table, BrowseNodeInfo returns the identity of child and parent browse nodes. That information is typically used with BrowseNodeLookup (p. 177) to traverse the browse node hierarchy.

Parent Response Group

The following response groups are parent response groups of BrowseNodeInfo.

- None

Child Response Group

The following response groups are child response groups of BrowseNodeInfo.

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=11232&
ResponseGroup=BrowseNodeInfo&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by BrowseNodeInfo.

```
<Item>
<ASIN>0131856340</ASIN>
<BrowseNodes>
  <BrowseNode>
  <BrowseNodeId>11232</BrowseNodeId>
  <Name> Social Sciences</Name>
  <Ancestors>
    <BrowseNode>
      <BrowseNodeId>53</BrowseNodeId>
      <Name>Nonfiction</Name>
      <Ancestors>
        <BrowseNode>
          <BrowseNodeId>1000</BrowseNodeId>
          <Name>Subjects</Name>
          <Ancestors>
            <BrowseNode>
              <BrowseNodeId>283155</BrowseNodeId>
              <Name>Books</Name>
            </BrowseNode>
          </Ancestors>
        </BrowseNode>
      </Ancestors>
    </BrowseNode>
  </Ancestors>
```

```
  <Children>
    <BrowseNode>
      <BrowseNodeId>11233</BrowseNodeId>
      <Name>Anthropology</Name>
    </BrowseNode>
    <BrowseNode>
      <BrowseNodeId>11242</BrowseNodeId>
      <Name>Archaeology</Name>
    </BrowseNode>
    <BrowseNode>
      <BrowseNodeId>3048861</BrowseNodeId>
      <Name>Children's Studies</Name>
    </BrowseNode>
  </Children>
</BrowseNodes>
```

Notice in this response snippet that the child nodes are all at the same level in the hierarchical browse node tree, which is the level directly beneath the browse node in the request, 11232, Social Science. The parent nodes, however, show the entire lineage of browse nodes from Social Science all the way up the browse node tree to the root browse node, Books, 283155. Only one lineage, however, is returned. If a single node has multiple parents, only one of those parents is returned.

BrowseNodes Response Group

The BrowseNodes response group returns the browse node names and IDs associated with the items returned in the response. The response group also returns the names and IDs of the child and parent browse nodes of the items returned in the response.

It is possible for one item to belong to multiple browse nodes. So, it is common to see multiple hierarchies of browse nodes for a single item.

Some products, such as parent ASINs, do not return information in the BrowsesNodes response group.

This response group is similar to the BrowseNodeInfo (p. 228) response group. The difference is that the BrowseNodes response group is used with operations that are based on item attributes, search indices, and lists. These operations typically return multiple items. BrowseNodeInfo can only be used with BrowseNodeLookup (p. 177) and the search is always keyed on a browse node ID.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by BrowseNodes.

- BrowseNodeId (p. 309)
- IsCategoryRoot (p. 314)
- Name (p. 317)
- TotalPages (p. 323)

- TotalResults (p. 323)

Parent Response Group

The following response groups are parent response groups of BrowseNodes.

- None

Child Response Group

The following response groups are child response groups of BrowseNodes.

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
SearchIndex=Books&
Keywords=Potter&
ResponseGroup=BrowseNodes&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by BrowseNodes.

```
<Item>
  <ASIN>B000002ADT</ASIN>
  <BrowseNodes>
    <BrowseNode>
    <BrowseNodeId>63926</BrowseNodeId>
    <Name>General</Name>
    <Ancestors>
      <BrowseNode>
        <BrowseNodeId>34</BrowseNodeId>
        <Name>Jazz</Name>
        <Ancestors>
          <BrowseNode>
            <BrowseNodeId>301668</BrowseNodeId>
            <Name>Styles</Name>
          </BrowseNode>
        </Ancestors>
      </BrowseNode>
 </Ancestors>
</BrowseNode>
<BrowseNode>
```

```
  <BrowseNodeId>598176</BrowseNodeId>
  <Name>Hard Bop</Name>
  <Ancestors>
    <BrowseNode>
      <BrowseNodeId>598174</BrowseNodeId>
      <Name>Bebop</Name>
      <Ancestors>
        <BrowseNode>
          <BrowseNodeId>34</BrowseNodeId>
          <Name>Jazz</Name>
            <Ancestors>
              <BrowseNode>
                <BrowseNodeId>301668</BrowseNodeId>
                <Name>Styles</Name>
              </BrowseNode>
            </Ancestors>
          </BrowseNode>
        </Ancestors>
      </BrowseNode>
    </Ancestors>
  </BrowseNode>
</BrowseNodes>
</Item>
</Items>
```

Cart Response Group

The Cart response group provides information about a specified remote shopping cart and the items in it. The cart information includes:

- CartId
- HMAC
- PurchaseURL

For each item in the cart, including SavedForLaterItems, the response group returns:

- CartItemId
- ProductName
- ASIN
- Quantity
- ListPrice
- OurPrice

Relevant Operations

Operations that can use this response group include:

- CartAdd (p. 180)
- CartCreate (p. 187)
- CartModify (p. 196)
- CartGet (p. 192)
- CartClear (p. 184)

Response Elements

The following table describes the elements returned by Cart.

Parent Response Group

The following response groups are parent response groups of Cart.

- None

Child Response Group

The following response groups are child response groups of Cart.

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
Operation=CartCreate&
Item.1.ASIN=B000062TU1&
Item.1.Quantity=2&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Cart.

```
<Cart>
  <Request>
  <IsValid>True</IsValid>
    <CartCreateRequest>
    <Items>
      <Item>
        <ASIN>B000062TU1</ASIN>
        <Quantity>2</Quantity>
      </Item>
    </Items>
    </CartCreateRequest>
  </Request>
  <CartId>002-5281165-2803250</CartId>
  <HMAC>5i1uO0G/PHqkvxZqC8oRkzmCano=</HMAC>
  <URLEncodedHMAC>5i1uO0G%2FPHqkvxZqC8oRkzmCano%3D</URLEncodedHMAC>
  <PurchaseURL>https://www.amazon.com/gp/cart/aws-merge.html?cart-id=002-5281165-
2803250%26associate-
id=ws%26hmac=5i1uO0G/PHqkvxZqC8oRkzmCano=%26SubscriptionId=1VMXF86PGNDAX3FW9C02</PurchaseURL>

  <SubTotal>
    <Amount>1994</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$19.94</FormattedPrice>
  </SubTotal>
  <CartItems>
    <SubTotal>
      <Amount>1994</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$19.94</FormattedPrice>
    </SubTotal>
  <CartItem>
  <CartItemId>U2ABORWEFJ0WZP</CartItemId>
  <ASIN>B000062TU1</ASIN>
  <SellerNickname>Amazon.com, LLC</SellerNickname>
  <Quantity>2</Quantity>
  <Title>Harry Potter and the Sorcerer's Stone (Full Screen Edition) (Harry
Potter 1)</Title>
  <ProductGroup>DVD</ProductGroup>
      <Price>
      <Amount>997</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$9.97</FormattedPrice>
    </Price>
    <ItemTotal>
      <Amount>1994</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$19.94</FormattedPrice>
    </ItemTotal>
  </CartItem>
</CartItems>
</Cart>
```

CartNewReleases Response Group

The CartNewReleases response group returns the ASINs and titles of the top five new releases in the root category of the item specified in the cart operation. For example, when adding a television to a cart the top five new releases in the root category, electronics, are returned.

One use of this response group is to suggest to customers additional items to buy.

Availability

This response group is available in the US locale only.

Relevant Operations

Operations that can use this response group include:

- CartAdd (p. 180)
- CartCreate (p. 187)
- CartModify (p. 196)
- CartGet (p. 192)

Response Elements

The following table describes the elements returned by CartNewReleases.

- ASIN (p. 308)
- Title (p. 322)

Parent Response Group

The following response groups are parent response groups of CartNewReleases.

- None

Child Response Group

The following response groups are child response groups of CartNewReleases.

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
Operation=CartCreate&
Item.1.ASIN=B000062TU1&
Item.1.Quantity=2&
```

```
ResponseGroup=CartNewReleases&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by CartNewReleases.

```
<NewReleases>
  <NewRelease>
    <ASIN>B00005JOFQ</ASIN>
    <Title>Brokeback Mountain (Widescreen Edition)</Title>
  </NewRelease>
  <NewRelease>
    <ASIN>B000EHRVMY</ASIN>
    <Title>Memoirs of a Geisha (Widescreen 2-Disc Special Edition)</Title>
  </NewRelease>
</NewReleases>
```

CartTopSellers Response Group

The CartTopSellers response group returns the ASINS and titles of the top five, best sellers in the root category of the item specified in the cart operation. For example, when adding a television to a cart, the five top sellers in the root category, electronics, are returned, for example, the top selling computers, MP3 players, or digital cameras.

Use this response group to specify additional, related items to buy.

Availability

This response group is available in the US locale only.

Relevant Operations

Operations that can use this response group include:

- CartAdd (p. 180)
- CartCreate (p. 187)
- CartModify (p. 196)
- CartGet (p. 192)

Response Elements

The following table describes the elements returned by CartTopSellers.

- ASIN (p. 308)
- Title (p. 322)

CartTopSellers also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of CartTopSellers.

- None

Child Response Group

The following response groups are child response groups of CartTopSellers.

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
Operation=CartCreate&
Item.1.ASIN=B000062TU1&
Item.1.Quantity=2&
ResponseGroup=CartTopSellers&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by CartTopSellers.

```
<TopSellers>
<TopSeller>
  <ASIN>B00005JOFQ</ASIN>
  <Title>Brokeback Mountain (Widescreen Edition)</Title>
  </TopSeller>
<TopSeller>
  <ASIN>B000E6EK3S</ASIN>
  <Title>Harry Potter and the Goblet of Fire (Widescreen Two-Disc Deluxe Edition)
 (Harry Potter 4)</Title>
  </TopSeller>
  </TopSellers>
```

CartSimilarities Response Group

The CartSimilarities (p. 237) response group returns the title and ASINs of items that:

- Are similar to the item specified in the request. These results, tagged in the XML response with <SimilarProducts> and <SimilarProduct>, are displayed on the retail web site under the heading, "Customers who bought this [item] also bought."

- Have been viewed by customers who also viewed the the item specified in the request. These results, tagged in the XML response with <SimilarViewedProducts> and <SimilarViewedProduct>, are displayed on the retail web site under the heading, "Customers who viewed this [item] also viewed"
- Can be found in other categories that are similar to the item specified in the request. These results, tagged in the XML response with <OtherCategoriesSimilarProducts> and <OtherCategoriesSimilarProduct>, are displayed on the retail web site under the heading, "Explore Similar Items," which falls under the heading, "Customers who bought this [item] also bought."

This response group returns a number of items similar to the one the customer added to the shopping cart. The response group is used typically to spur the customer's interest in purchasing additional items.

Relevant Operations

Operations that can use this response group include:

- CartAdd (p. 180)
- CartCreate (p. 187)
- CartModify (p. 196)
- CartGet (p. 192)

Response Elements

The following table describes the elements returned by CartSimilarities.

- ASIN (p. 308)
- OtherCategoriesSimilarProducts (p. 318)
- SimilarProducts (p. 321)
- SimilarViewedProducts (p. 321)
- Title (p. 322)

CartSimilarities also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of CartSimilarities.

- None

Child Response Group

The following response groups are child response groups of CartSimilarities.

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=ws&
Operation=CartCreate&
Item.1.ASIN=B000062TU1&
Item.1.Quantity=2&
ResponseGroup=CartSimilarities&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by CartSimilarities.

```
<SimilarProducts>
  <SimilarProduct>
    <ASIN>B00008DDXC</ASIN>
    <Title>Harry Potter and the Chamber of Secrets (Widescreen Edition) (Harry
 Potter 2)</Title>
  </SimilarProduct>
</SimilarProducts>
<SimilarViewedProducts>
  <SimilarViewedProduct>
    <ASIN>B000E6UZZK</ASIN>
    <Title>Harry Potter Years 1-4 (Harry Potter and the Sorcerer's Stone /
Chamber of Secrets / Prisoner of Azkaban / Goblet of Fire) (Widescreen
Edition)</Title>
  </SimilarViewedProduct>
</SimilarViewedProducts>
<OtherCategoriesSimilarProducts>
  <OtherCategoriesSimilarProduct>
    <ASIN>0590353403</ASIN>
    <Title>Harry Potter and the Sorcerer's Stone (Book 1)</Title>
  </OtherCategoriesSimilarProduct>
</OtherCategoriesSimilarProducts>
```

Collections Response Group

For every item returned in a response, the items associated with it are also returned if the Collections response group is used in the request. Items in collections are related thematically. For example, all of the linens that go into a bedroom might be associated in a bedding collection. The Collections response group returns the ASINs and titles of the items in a collection.

For more information, see Displaying Collections (p. 44).

If an item is part of a collection and that item is returned in a response, all of the other items in the collection are also returned if the request contains the Collections response group.

Relevant Operations

Operations that can use this response group include:

- ItemLookup

- ItemSearch

Response Elements

The following table describes and shows the parentage of the elements returned by Collections.

- ASIN (p. 308)
- Collection (p. 310)
- CollectionItem (p. 310)
- CollectionParent (p. 310)
- Collections (p. 310)
- Title (p. 322)

Parent Response Group

The following response groups are parent response groups of Collections.

- None

Child Response Group

The following response groups are children response groups of Collections.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by Collections.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKey=[Access Key ID]&
Operation=ItemLookup&
ItemId=B000ALMQ9C&
ResponseGroup=ItemIds,Collections&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Collections.

```
<Collections>
  <Collection>
    <CollectionParent>
      <ASIN>B0006PLAOE</ASIN>
      <Title>Fieldcrest® Classic Bedding Collection GarnetIvory</Title>
    </CollectionParent>
    <CollectionItem>
      <ASIN>B00067IV8U</ASIN>
```

```
      <Title>Fieldcrest® Classic Solid Sheets Garnet</Title>
    </CollectionItem>
    <CollectionItem>
      <ASIN>B000673NE2</ASIN>
      <Title>Fieldcrest® Classic Solid Pillowcases Set of 2 Garnet</Title>
    </CollectionItem>
    <CollectionItem>
      <ASIN>B00065WTJY</ASIN>
      <Title>Fieldcrest® Classic Dobby Pillowcases Set of 2 Garnet</Title>
    </CollectionItem>
  </Collection>
</Collections>
```

EditorialReview Response Group

For each item in the response, the EditorialReview response group returns Amazon's review of the item, which, on the Detail page, is labeled the Product Description.

Note

Copyrighted editorial reviews are not returned. For this reason, the reviews returned may be different than those returned by *www.amazon.com*.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by EditorialReview.

- Content
- EditorialReviewIsLinkSuppressed (p. 312)
- Source (p. 321)

EditorialReview also returns the elements that all response groups return, which is described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of EditorialReview.

- None

Child Response Group

The following response groups are child response groups of EditorialReview.

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.jp/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&Keywords=Fable&
SearchIndex=Blended&
ResponseGroup=EditorialReview&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by EditorialReview.

```
<EditorialReviews>
  <EditorialReview>
    <Source>From Amazon.com</Source>
    <Content>Considered by many to be the last great musical comedy, then the
rest of the review continues here.
    </Content>
  </EditorialReview>
</EditorialReviews>
```

Images Response Group

The Images response group returns the URLs to all available images of an item in three sizes: small, medium, and large. For example, if a blender is for sale and there are four images of it, this response group returns the URLs of twelve images: four images, each in three sizes.

In addition to returning the URLs of the images, the response groups also returns the height and width dimensions of each image. Use these values to help you display the images correctly.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by the Images response group.

- Height (p. 314)
- LargeImage (p. 316)

- MediumImage (p. 317)
- SmallImage (p. 321)
- SwatchImage (p. 322)
- ThumbnailImage (p. 322)
- TinyImage (p. 322)
- TotalPages (p. 323)
- TotalResults (p. 323)
- URL (p. 324)
- Width (p. 325)

Images also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of the Images response group.

- None

Child Response Group

The following response groups are child response groups of the Images response group.

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
ItemId=B000Q678OO&
ResponseGroup=Images&
SearchIndex=Blended&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by the Images response group.

```
<Item>
  <ASIN>B000Q678OO</ASIN>
  <SmallImage>
  <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL75_.jpg</URL>
  <Height Units="pixels">75</Height>
  <Width Units="pixels">58</Width>
  </SmallImage>
```

```
  <MediumImage>
  <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL160_.jpg</URL>
  <Height Units="pixels">160</Height>
  <Width Units="pixels">124</Width>
  </MediumImage>
  <LargeImage>
  <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L.jpg</URL>
  <Height Units="pixels">500</Height>
  <Width Units="pixels">389</Width>
  </LargeImage>

<ImageSets>
  <ImageSet Category="primary">
  <SwatchImage>
  <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL30_.jpg</URL>
  <Height Units="pixels">30</Height>
  <Width Units="pixels">23</Width>
  </SwatchImage>
  <SmallImage>
  <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL75_.jpg</URL>
  <Height Units="pixels">75</Height>
  <Width Units="pixels">58</Width>
  </SmallImage>
  <ThumbnailImage>
  <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL75_.jpg</URL>
  <Height Units="pixels">75</Height>
  <Width Units="pixels">58</Width>
  </ThumbnailImage>
  <TinyImage>
  <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL110_.jpg</URL>
  <Height Units="pixels">110</Height>
  <Width Units="pixels">86</Width>
  </TinyImage>
  <MediumImage>
  <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L._SL160_.jpg</URL>
  <Height Units="pixels">160</Height>
  <Width Units="pixels">124</Width>
  </MediumImage>
  <LargeImage>
  <URL>http://ecx.images-amazon.com/images/I/51YL4rlI%2B9L.jpg</URL>
  <Height Units="pixels">500</Height>
  <Width Units="pixels">389</Width>
  </LargeImage>
 </ImageSet>
</ImageSets>
```

The images are returned in two ways: under <Item> and under <ImageSets>. The images under under <Item> are specified by the <SmallImage> , <MediumImage> , and <LargeImage> elements.

The sizes of the images under <ImageSets> are specified by the _SLXXX_ suffix in the URL, where XXX is the number of pixels on the longest side of the image. A medium size image, for example, has 160 pixels on its longest side so it has the suffix _SL160_. This is the preferred way to reference images.

The ImageSets element attribute, Category, is set to Primary. Primary images are the same images that appear in the <Item> section.

ItemAttributes Response Group

The ItemAttributes response group returns a potentially large number of attributes that describe an item. For example, an item in the Camera and Photo search index might return the attributes, height, width, weight, title, UPC, price, manufacture, zoom ratio, number of megapixels, and carrying case.

All search indices can return all item attributes. The number of item attributes returned, however, varies by ASIN. Typically, ASINs within the same search index return the same item attributes. For example, you would expect the item attributes returned for an item in the "Books" search index to be different from those returned for an item in the "Camera and Photo" search index. But all items within a single search index do not necessarily return the same attributes. For that reason, it is impossible to predict exactly which item attributes will be returned in a response.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by ItemAttributes.

- Actor (p. 307)
- Artist (p. 308)
- AspectRatio (p. 308)
- AudienceRating (p. 308)
- AudioFormat (p. 308)
- Author (p. 308)
- Binding (p. 309)
- Brand (p. 309)
- CatalogNumberList
  - CatalogNumberListElement
- Category (p. 310)
- CEROAgeRating (p. 310)
- ClothingSize (p. 310)
- Color (p. 310)
- Creator (p. 311)
  - Role (p. 320)
- Department (p. 311)
- Director (p. 312)
- EAN (p. 312)
- EANList (p. 312)
  - EANListElement (p. 312)
- Edition (p. 312)
- EISBN (p. 312)
- EpisodeSequence (p. 312)

ItemAttributes also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of ItemAttributes.

- None

Child Response Group

The following response groups are child response groups of ItemAttributes.

- None

Sample REST Use Case

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
```

```
Operation=ItemSearch&
Condition=All&
ResponseGroup=ItemAttributes&
SearchIndex=Blended&
Keywords=GodSmack&
Merchant=All&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by ItemAttributes.

```
<Item>
  <ASIN>B000A2XB9U</ASIN>
  <ItemAttributes>
    <AudienceRating>NR (Not Rated)</AudienceRating>
    <Director>Lawrence Jordan (II)</Director>
    <Director>Daniel E. Catullo</Director>
    <EAN>0014381273229</EAN>
    <Format>Color</Format>
    <Format>Compilation</Format>
    <Format>NTSC</Format>
    <Languages>
      <Language>
        <Name>English</Name>
        <Type>Original Language</Type>
      </Language>
    </Languages>
    <ListPrice>
      <Amount>1999</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$19.99</FormattedPrice>
    </ListPrice>
    <NumberOfItems>1</NumberOfItems>
    <ProductGroup>DVD</ProductGroup>
    <ReleaseDate>2005-09-06</ReleaseDate>
    <RunningTime Units="minutes">131</RunningTime>
    <Studio>Image Entertainment</Studio>
    <TheatricalReleaseDate>2005-06-25</TheatricalReleaseDate>
    <Title>Rockin' the Corps</Title>
    <UPC>014381273229</UPC>
  </ItemAttributes>
</Item>
```

ItemIds Response Group

The ItemIds response group returns the ASINs for all items returned in a response.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)

- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by ItemIds.

- ASIN (p. 308)
- CorrectedQuery (p. 311)
- Keywords (p. 315)
- Message (p. 317)
- TotalPages (p. 323)
- TotalResults (p. 323)

ItemIds also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of ItemIds.

- None

Child Response Group

The following response groups are child response groups of ItemIds.

- None

Sample REST Use Case

All locales.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B000A3UB2O&
ResponseGroup=ItemIds&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by ItemIds.

```
  <TotalResults>1</TotalResults>
  <TotalPages>1</TotalPages>
  <Item>
    <ASIN>0976925524</ASIN>
```

```
  </Item>
</Items>
```

Large Response Group

The `Large` response group returns a great deal of information about items in the response. Large is a parent response group that returns the results of the following response groups:

- Accessories (p. 224)
- BrowseNodes (p. 230)
- Medium (p. 255)
- Offers (p. 267)
- Reviews (p. 280)
- Similarities (p. 288)
- Tracks (p. 292)

For more information about what is returned by each of these response groups, refer to their sections.

The Large (p. 250) response group is ideally suited for building product detail pages similar to those found on Amazon's retail web site.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by Large.

- Actor (p. 307)
- Amount (p. 308)
- Artist (p. 308)
- ASIN (p. 308)
- AspectRatio (p. 308)
- AudienceRating (p. 308)
- AudioFormat (p. 308)
- Author (p. 308)
- BackFinding
- BandMaterialType
- Binding (p. 309)
- Brand (p. 309)
- BrowseNodeId (p. 309)
- ChainType
- ClaspType
- ClothingSize (p. 310)

- ReleaseDate (p. 320)
- Role (p. 320)
- RunningTime (p. 320)
- SalesRank (p. 320)
- Size (p. 321)
- SizePerPearl
- SKU (p. 321)
- Source (p. 321)
- Studio (p. 321)
- SubscriptionLength (p. 322)
- Title (p. 322)
- TotalCollectible (p. 322)
- TotalDiamondWeight
- TotalGemWeight
- TotalUsed (p. 323)
- TotalNew (p. 322)
- TotalPages (p. 323)
- TotalRefurbished (p. 323)
- TotalResults (p. 323)
- Type (p. 324)
- UPC (p. 324)
- URL (p. 324)
- Warranty (p. 325)
- Weight (p. 325)
- Width (p. 325)

Large also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of Large.

- None

Child Response Group

The following response groups are child response groups of Large.

- Accessories (p. 224)
- BrowseNodes (p. 230)
- Medium (p. 255)
- Offers (p. 267)
- Reviews (p. 280)
- Similarities (p. 288)
- Tracks (p. 292)

Sample REST Use Case

The following request shows the results of the Large response group.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B000ESHHXG&
ResponseGroup=Large&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Large.

```
<Item>
  <ASIN>B000A3UB2O</ASIN>
  <SalesRank>47589</SalesRank>
  <SmallImage>
    <URL>http://ec1.images-amazon.com/images/P/B000A3UB2O.01-
A3QXF272WQ86AH._SCTHUMBZZZ_.jpg</URL>
    <Height Units="pixels">75</Height>
    <Width Units="pixels">75</Width>
  </SmallImage>
  <MediumImage>
    <URL>http://ec1.images-amazon.com/images/P/B000A3UB2O.01-
A3QXF272WQ86AH._SCMZZZZZZZ_.jpg</URL>
    <Height Units="pixels">160</Height>
    <Width Units="pixels">160</Width>
 </MediumImage>
  <LargeImage>
    <URL>http://ec1.images-amazon.com/images/P/B000A3UB2O.01-
A3QXF272WQ86AH._SCLZZZZZZZ_.jpg</URL>
    <Height Units="pixels">450</Height>
    <Width Units="pixels">450</Width>
  </LargeImage>
  <ImageSets>
    <ImageSet Category="primary">
      <SmallImage>
        <URL>http://ec1.images-amazon.com/images/P/B000A3UB2O.01-A3QXF272WQ86A
 H._SCTHUMBZZZ_.jpg</URL>
        <Height Units="pixels">75</Height>
        <Width Units="pixels">75</Width>
      </SmallImage>
...
<ItemAttributes>
  <Brand>Nixon</Brand>
  <ProductGroup>Apparel</ProductGroup>
  <Title>Nixon Rotolog Wood</Title>
</ItemAttributes>
<OfferSummary>
  <LowestNewPrice>
    <Amount>19999</Amount>
```

```
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$199.99</FormattedPrice>
  </LowestNewPrice>
  <TotalNew>1</TotalNew>
  <TotalUsed>0</TotalUsed>
  <TotalCollectible>0</TotalCollectible>
  <TotalRefurbished>0</TotalRefurbished>
</OfferSummary>
<Offers>
  <TotalOffers>0</TotalOffers>
  <TotalOfferPages>0</TotalOfferPages>
</Offers>
<EditorialReviews>
  <EditorialReview>
    <Source>Product Description</Source>
    <Content>Custom right read direct time. Japanese quartz with led. Custom
30  Meter stainless steel with hardened mineral crystal, white inlay and double
 gasket crown. Custom solid stainless steel with white inlay and butterfly
closure.</Content>
  </EditorialReview>
</EditorialReviews>
<BrowseNodes>
  <BrowseNode>
    <BrowseNodeId>1045534</BrowseNodeId>
    <Name>Jewelry</Name>
    <Ancestors>
      <BrowseNode>
        <BrowseNodeId>1044486</BrowseNodeId>
        <Name>Girls</Name>
        <Ancestors>
           <BrowseNode>
             <BrowseNodeId>1044484</BrowseNodeId>
             <Name>Children's Accessories</Name>
             <Ancestors>
               <BrowseNode>
                 <BrowseNodeId>1036700</BrowseNodeId>
                 <Name>Accessories</Name>
                 <Ancestors>
                   <BrowseNode>
                     <BrowseNodeId>1036682</BrowseNodeId>
                   <Name>Departments</Name>
                 <Ancestors>
               <BrowseNode>

                       <BrowseNodeId>1036592</BrowseNodeId>
                     <Name>Apparel</Name>
                   </BrowseNode>
                  </Ancestors>
                 </BrowseNode>
               </Ancestors>
              </BrowseNode>
             </Ancestors>
           </BrowseNode>
        </Ancestors>
      </BrowseNode>
    </Ancestors>
  </BrowseNode>
<BrowseNode>
```

Medium Response Group

The Medium response group returns a great deal of information about the items in a response. The response group is ideally suited for creating lightweight, product detail pages.

Medium is a parent response group that returns the results of the following response groups:

- EditorialReview (p. 241)
- Images (p. 242)
- ItemAttributes (p. 245)
- OfferSummary (p. 271)
- Request (p. 278)
- SalesRank (p. 282)
- Small (p. 289)

For more information about what is returned by each of these response groups, refer to their sections.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by Medium.

- Actor (p. 307)
- Amount (p. 308)
- Artist (p. 308)
- ASIN (p. 308)
- AspectRatio (p. 308)
- AudienceRating (p. 308)
- AudioFormat (p. 308)
- Author (p. 308)
- BackFinding
- BandMaterialType
- Binding (p. 309)
- Brand (p. 309)
- ChainType
- ClaspType
- ClothingSize (p. 310)
- Color (p. 310)
- Content
- Creator (p. 311)
- CurrencyCode (p. 311)
- Department (p. 311)

- SizePerPearl
- SKU (p. 321)
- Source (p. 321)
- Studio (p. 321)
- SubscriptionLength (p. 322)
- Title (p. 322)
- TotalCollectible (p. 322)
- TotalDiamondWeight
- TotalGemWeight
- TotalUsed (p. 323)
- TotalNew (p. 322)
- TotalPages (p. 323)
- TotalRefurbished (p. 323)
- TotalResults (p. 323)
- Type (p. 324)
- UPC (p. 324)
- URL (p. 324)
- Warranty (p. 325)
- Weight (p. 325)
- Width (p. 325)

Medium also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of Medium.

- None

Child Response Group

The following response groups are child response groups of Medium.

- EditorialReview (p. 241)
- Images (p. 242)
- ItemAttributes (p. 245)
- OfferSummary (p. 271)
- Request (p. 278)
- SalesRank (p. 282)
- Small (p. 289)

Sample REST Use Case

The following request illustrates the XML response elements returned by Medium.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=1890966533&
ResponseGroup=Medium&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Medium.

```
<Item>
      <ASIN>1890966533</ASIN>
      <SalesRank>1738674</SalesRank>
      <SmallImage>
         <URL>http://ec1.images-
amazon.com/images/P/1890966533.01._SCTHUMBZZZ_.jpg          </URL>
         <Height Units="pixels">60</Height>
         <Width Units="pixels">40</Width>
      </SmallImage>
      <MediumImage>
         <URL>http://ec1.images-
amazon.com/images/P/1890966533.01._SCMZZZZZZZ_.jpg         </URL>
         <Height Units="pixels">140</Height>
         <Width Units="pixels">94</Width>
     </MediumImage>
     <ItemAttributes>
      <Author>Wayne G. Pardy</Author>
      <Binding>Paperback</Binding>
      <EAN>9781890966539</EAN>
      <Edition>Reprint</Edition>
      <ISBN>1890966533</ISBN>
      <ListPrice>
        <Amount>5995</Amount>
        <CurrencyCode>USD</CurrencyCode>
        <FormattedPrice>$59.95</FormattedPrice>
      </ListPrice>
      <NumberOfPages>185</NumberOfPages>
      <PackageDimensions>
        <Height Units="hundredths-inches">900</Height>
        <Length Units="hundredths-inches">75</Length>
        <Weight Units="hundredths-pounds">130</Weight>
        <Width Units="hundredths-inches">600</Width>
      </PackageDimensions>
      <ProductGroup>Book</ProductGroup>
      <PublicationDate>1999-05-19</PublicationDate>
      <Publisher>Safetycertified.Com Inc</Publisher>
      <Title>Safety Incentives: The Pros and Cons of Award and Recognition
Programs (Osha Compliance)</Title>
    </ItemAttributes>
    <OfferSummary>
      <TotalNew>0</TotalNew>
      <TotalUsed>0</TotalUsed>
      <TotalCollectible>0</TotalCollectible>
```

```
      <TotalRefurbished>0</TotalRefurbished>
    </OfferSummary>
    <EditorialReviews>
      <EditorialReview>
        <Source>Book Description</Source>
        <Content>Whether you are a small business or a safety manager trying
to decide whether or not awards and incentives are right for your operation,
the Safety Incentives Answer Book is must reading.
        </Content>
      </EditorialReview>
    </EditorialReviews>
  </Item>
```

MostGifted Response Group

The MostGifted response group returns the ASINs and titles of the ten items given as gifts most within a specified browse node.

Availability

This response group is available in all locales.

Relevant Operations

Operations that can use this response group include:

- BrowseNodeLookup (p. 177)

Response Elements

The following table describes the elements returned by MostGifted. In the Ancestry column, the elements on the left side of a slash mark are the parents of the elements on the right side of the slash mark.

- Actors (p. 307)
- Artist (p. 308)
- ASIN (p. 308)
- Authors (p. 308)
- ProductGroup (p. 319)
- Title (p. 322)

MostGifted also returns the elements that all response groups return, which is described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of MostGifted.

- None

Child Response Group

The following response groups are child response groups of MostGifted.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by MostGifted.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=20&
ResponseGroup=MostGifted&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by MostGifted.

```
<TopItemSet>
   <Type>MostGifted</Type>
   <TopItem>
            <ASIN>0553576399</ASIN>
            <Title>Distraction</Title>
            <ProductGroup>Book</ProductGroup>
            <Author>Bruce Sterling</Author>
       </TopItem>
       <TopItem>
        ....
       .....
</TopItemSet>
```

MostWishedFor Response Group

The MostWishedFor response group returns the ASINs and titles of the ten items given as the items listed on the greatest number of wishlists within a specified browse node.

Availability

This response group is available in all locales.

Relevant Operations

Operations that can use this response group include:

- BrowseNodeLookup (p. 177)

Response Elements

The following table describes the elements returned by MostWishedFor.

- Actors (p. 307)
- Artist (p. 308)
- ASIN (p. 308)
- Authors (p. 308)
- ProductGroup (p. 319)
- Title (p. 322)

MostWishedFor also returns the elements that all response groups return, which is described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of MostWishedFor.

- None

Child Response Group

The following response groups are child response groups of MostWishedFor.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by MostWishedFor.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=20&
ResponseGroup=MostWishedFor&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by MostWishedFor.

```
<TopItemSet>
   <Type>MostWishedFor</Type>
   <TopItem>
            <ASIN>0553576399</ASIN>
            <Title>Distraction</Title>
            <ProductGroup>Book</ProductGroup>
            <Author>Bruce Sterling</Author>
       </TopItem>
```

```
        <TopItem>
         ....
        .....
</TopItemSet>
```

NewReleases Response Group

The NewReleases response group returns the ASIN and title of newly released items in a specified browse node.

Availability

This response group is available in the US locale only.

Relevant Operations

Operations that can use this response group include:

- BrowseNodeLookup (p. 177)

Response Elements

The following table describes the elements returned by NewReleases.

- Actors (p. 307) for the Video search index
- Artist (p. 308) for the Music search index
- ASIN (p. 308)
- Authors (p. 308) for the Books search index
- ProductGroup (p. 319)
- Title (p. 322)
- TopItemSet (p. 322)

NewReleases also returns the elements that all response groups return, which is described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of NewReleases.

- None

Child Response Group

The following response groups are child response groups of NewReleases.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by NewReleases.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=4229&
ResponseGroup=NewReleases&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by NewReleases.

```
<TopItemSet>
   <Type>NewReleases</Type>
   <TopItem>
            <ASIN>0553576399</ASIN>
            <Title>Distraction</Title>
            <ProductGroup>Book</ProductGroup>
            <Author>Bruce Sterling</Author>
       </TopItem>
       <TopItem>
        ....
       .....
</TopItemSet>
```

OfferFull Response Group

The OfferFull response group returns comprehensive information about an offer. OfferFull is a parent response group that returns the results of the Offers (p. 267) response group.

Note

Not all merchants will have this data.

This response group is not returned for Amazon Kindle digital books. An Amazon Kindle ASIN can be verified through the *Binding*, *Format*, and *ProductTypeName* response elements.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by OfferFull.

- Amount (p. 308)
- Availability (p. 308)

- Condition (p. 310)
- CurrencyCode (p. 311)
- FormattedPrice (p. 313)
- IsEligibleForSuperSaverShipping (p. 314)
- MoreOffersURL
- Name (p. 317)
- OfferListingId (p. 318)
- TotalCollectible (p. 322)
- TotalNew (p. 322)
- TotalOfferPages (p. 323)
- TotalOffers (p. 323)
- TotalRefurbished (p. 323)
- TotalUsed (p. 323)

OfferFull also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of OfferFull.

- None

Child Response Group

The following response groups are child response groups of OfferFull.

- Offers (p. 267)

Sample REST Use Case

The following request illustrates the XML response elements returned by OfferFull.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=All&
SearchIndex=Books&
Title=Harry%20Potter&
ResponseGroup=OfferFull&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by OfferFull.

```
<Item>
  <ASIN>0439682584</ASIN>
  <OfferSummary>
    <LowestNewPrice>
      <Amount>2580</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$25.80</FormattedPrice>
    </LowestNewPrice>
    <LowestUsedPrice>
      <Amount>1599</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$15.99</FormattedPrice>
    </LowestUsedPrice>
    <LowestCollectiblePrice>
      <Amount>2580</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$25.80</FormattedPrice>
    </LowestCollectiblePrice>
    <TotalNew>40</TotalNew>
    <TotalUsed>16</TotalUsed>
    <TotalCollectible>3</TotalCollectible>
    <TotalRefurbished>0</TotalRefurbished>
  </OfferSummary>
  <Offers>
    <TotalOffers>0</TotalOffers>
    <TotalOffers>1</TotalOffers>
    <TotalOfferPages>0</TotalOfferPages>
    <TotalOfferPages>1</TotalOfferPages>
    <Offer>
      <Merchant>
        <Name>Amazon.com</Name>
      </Merchant>
    <OfferAttributes>
      <Condition>New</Condition>
    </OfferAttributes>
    <OfferListing>

<OfferListingId>cMVuIFx8kiYSgRIJXiCzKZZyylwztVSAYV8vCo2OxHS8L9SB7lwho8fK6CxYkmdDPy8thFzm30Y%3D</OfferListingId>

      <Price>
        <Amount>2580</Amount>
        <CurrencyCode>USD</CurrencyCode>
        <FormattedPrice>$25.80</FormattedPrice>
      </Price>
      <Availability>Usually ships in 24 hours</Availability>
      <IsEligibleForSuperSaverShipping>1</IsEligibleForSuperSaverShipping>
    </OfferListing>
  </Offer>
 </Offers>
</Item>
```

OfferListings

The `OfferListings` response group returns the `OfferListings` for items returned in the response. The values returned are similar to those returned by the `Offers` response group minus the values returned by the `OfferSummary` response group. OfferListings returns shipping options, including

`IsEligibleForSuperSavingShipping` which specifies whether the item qualifies for super saver shipping.

Note

This response group is not returned for Amazon Kindle digital books. An Amazon Kindle ASIN can be verified through the `Binding`, `Format`, and `ProductTypeName` response elements.

Relevant Operations

Operations that can use this response group include:

- ItemLookup
- ItemSearch
- SimilarityLookup

Response Elements

The following table describes and shows the parentage of the elements returned by OfferListings. In the Ancestry column, the elements on the left side of a slash mark are the parents of the elements on the right side of the slash mark.

OfferListings also returns the elements that all response groups return, which described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of OfferListings.

- None

Child Response Group

The following response groups are children response groups of OfferListings.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by OfferListings.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
AssociateTag=[Associate ID]&
Operation=ItemSearch&
Keywords=sports&
ResponseGroup=OfferListings&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by OfferListings.

```
<Item>
   <ASIN>B000AYGDIO</ASIN>
   <Offers>
      <TotalOffers>18</TotalOffers>
      <TotalOfferPages>2</TotalOfferPages>
      <Offer>
         <OfferAttributes>
            <Condition>New</Condition>
         </OfferAttributes>
         <OfferListing>
            <OfferListingId>
              pBmLD7%2F4J7zqIMjOLpQI5pkO774zf%2BsvrzbG4JR92xLmC%
            </OfferListingId>
            <Price>
               <Amount>24700</Amount>
               <CurrencyCode>USD</CurrencyCode>
               <FormattedPrice>$247.00</FormattedPrice>
            </Price>
            <Availability>Usually ships in 1-2 business days</Availability>
          <IsEligibleForSuperSaverShipping>0</IsEligibleForSuperSaverShipping>


         </OfferListing>
      </Offer>
      <!-- 9 ADDITIONAL OFFERS -->
   </Offers>
</Item>
```

Offers Response Group

The Offers response group is a parent response group that returns the contents of the OfferSummary Response Group (p. 271) response group plus, by default, offer listing information. The Offers response group can take *Condition* and *MerchantId* as an optional input parameter. By default, the offer for Buy Box winner is returned, which is an item listed in the box on an item's detail page that enables you to add an item to a shopping cart. The only valid values for MerchantId are “Amazon”, to get the Amazon offer for an item, if it exists and “All” which is the default value.

The following table shows the expected behavior of the Offers response group for various input parameters:

Optional Input Parameters		Offer(s) Data Returned by the API
Condition	**MerchantId**	
(Unspecified)	(Unspecified)	Buy Box Winner if it exists [OR] Lowest priced new offer.
(Unspecified)	Amazon	Amazon offer
(Unspecified)	Featured, Featured Buy Box Winner, or All	Buy Box Winner if it exists [OR] Lowest priced new offer. The specified MerchantId parameter will not have any effect on the response.
(Unspecified)	Any other value	Error: invalid value
Used, New, Refurbished, or Collectible	(Unspecified) or All	Lowest priced offer for given condition
Used, New, Refurbished, or Collectible	Amazon	Amazon offer
Used, New, Refurbished, or Collectible	Featured, Featured Buy Box Winner, or All	Lowest priced offer for given condition. The specified MerchantId parameter will not have any effect on the response.
Used, New, Refurbished, or Collectible	Any other value	Error: invalid value
All	(Unspecified)	Lowest priced offer for each condition
All	Amazon	Amazon offer
All	Featured, Featured Buy Box Winner, or All	Lowest priced offer for each condition. The specified MerchantId parameter will not have any effect on the response.
(Unspecified)	Any other value	Error: invalid value

This response group is not returned for Amazon Kindle digital books. An Amazon Kindle ASIN can be verified through the `Binding`, `Format`, and `ProductTypeName` response elements.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by Offers.

- Amount (p. 308)
- Availability (p. 308)
- Condition (p. 310)
- CurrencyCode (p. 311)
- FormattedPrice (p. 313)
- IsEligibleForSuperSaverShipping (p. 314)
- LoyaltyPoints (p. 316)
- ???
- Name (p. 317)
- OfferListingId (p. 318)
- TotalCollectible (p. 322)
- TotalNew (p. 322)
- TotalOfferPages (p. 323)
- TotalOffers (p. 323)
- TotalRefurbished (p. 323)
- TotalUsed (p. 323)

Offers also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Loyalty Points

In the JP locale only, loyalty points are returned. Loyalty points are used to encourage patronage and stimulate sales.

Parent Response Group

The following response groups are parent response groups of Offers.

- None

Child Response Group

The following response groups are child response groups of Offers.

- OfferSummary (p. 271)

Sample REST Use Case

The following request illustrates the XML response elements returned by Offers.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=1VEXAMPLEDS6W82&
Operation=ItemSearch&
Condition=All&
SearchIndex=Books&
Title=Harry%20Potter&
ResponseGroup=Offers&
Version=2011-08-01
```

```
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Offers.

```
<Item>
  <ASIN>047061529X</ASIN>
  <OfferSummary>
    <LowestNewPrice>
      <Amount>1025</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$10.25</FormattedPrice>
    </LowestNewPrice>
    <LowestUsedPrice>
      <Amount>1110</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$11.10</FormattedPrice>
    </LowestUsedPrice>
    <TotalNew>37</TotalNew>
    <TotalUsed>12</TotalUsed>
    <TotalCollectible>0</TotalCollectible>
    <TotalRefurbished>0</TotalRefurbished>
  </OfferSummary>
  <Offers>
    <TotalOffers>2</TotalOffers>
    <TotalOfferPages>1</TotalOfferPages>
    <MoreOffersUrl>http://www.amazon.com/gp/offer-
listing/047061529X/?SubscriptionId=0TMGJB9ZR3ERMA342002&ie=UTF8&tag=adrpik-
20&creative=386001&camp=2025&linkCode=xm2 </MoreOffersUrl>
    <Offer>
      <OfferAttributes>
        <Condition>New</Condition>
      </OfferAttributes>
      <OfferListing>

<OfferListingId>[illegible]
 </OfferListingId>
        <Price>
          <Amount>1025</Amount>
          <CurrencyCode>USD</CurrencyCode>
          <FormattedPrice>$10.25</FormattedPrice>
        </Price>
        <AmountSaved>
          <Amount>974</Amount>
          <CurrencyCode>USD</CurrencyCode>
          <FormattedPrice>$9.74</FormattedPrice>
        </AmountSaved>
        <PercentageSaved>49</PercentageSaved>
        <Availability>Usually ships in 1-2 business days</Availability>
        <AvailabilityAttributes>
          <AvailabilityType>now</AvailabilityType>
          <MinimumHours>24</MinimumHours>
          <MaximumHours>48</MaximumHours>
        </AvailabilityAttributes>
        <IsEligibleForSuperSaverShipping>0</IsEligibleForSuperSaverShipping>
```

```
      </OfferListing>
    </Offer>

    <Offer>
      <OfferAttributes>
        <Condition>Used</Condition>
      </OfferAttributes>
      <OfferListing>

<OfferListingId>Xle7H3gZ7B1Vcs5HjsPDGBDCrzXPZEDRfwjXbHZMB4vPNZlxC1GC1ZYiuPTKsGHVZKA
 </OfferListingId>
        <Price>
          <Amount>1110</Amount>
          <CurrencyCode>USD</CurrencyCode>
          <FormattedPrice>$11.10</FormattedPrice>
        </Price>
        <AmountSaved>
          <Amount>889</Amount>
          <CurrencyCode>USD</CurrencyCode>
          <FormattedPrice>$8.89</FormattedPrice>
        </AmountSaved>
        <PercentageSaved>44</PercentageSaved>
        <Availability>Usually ships in 1-2 business days</Availability>
        <AvailabilityAttributes>
          <AvailabilityType>now</AvailabilityType>
          <MinimumHours>24</MinimumHours>
          <MaximumHours>48</MaximumHours>
        </AvailabilityAttributes>
        <IsEligibleForSuperSaverShipping>0</IsEligibleForSuperSaverShipping>
      </OfferListing>
    </Offer>
  </Offers>
</Item>
```

The Request that Generated the Response (REST)

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=<replaceable>
<replaceable>[AWS Access Key ID]</replaceable>
</replaceable>&
AssociateTag=<replaceable>
<replaceable>[Associate ID]</replaceable>
</replaceable>&
Operation=ItemLookup&
ItemId=B000AYGDIO&
ResponseGroup=Offers
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

OfferSummary Response Group

The OfferSummary (p. 271) response group returns, for each item in the response, the number of offer listings and the lowest price for each condition type. Condition types are New, Used, Collectible, and Refurbished. So, for example, this response group returns the lowest price for each Condition:

- New item
- Used item
- Collectible item
- Refurbished item

Individual offer listings are not returned. The OfferSummary is dependent only on the ASIN parameter and is not affected by the MerchantId (p. 317) or Condition (p. 310) parameters (i.e. the OfferSummary will always be the same for a given ASIN independent of other parameters).

Note

This response group is not returned for Amazon Kindle digital books. An Amazon Kindle ASIN can be verified through the `Binding`, `Format`, and `ProductTypeName` response elements.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by OfferSummary.

- Amount (p. 308)
- CurrencyCode (p. 311)
- FormattedPrice (p. 313)
- TotalCollectible (p. 322)
- TotalNew (p. 322)
- TotalRefurbished (p. 323)
- TotalUsed (p. 323)

OfferSummary also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of OfferSummary.

- None

Child Response Group

The following response groups are child response groups of OfferSummary.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by OfferSummary.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B000A3UB2O&
ResponseGroup=OfferSummary&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by OfferSummary.

```
<OfferSummary>
  <LowestNewPrice>
    <Amount>801</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$8.01</FormattedPrice>
  </LowestNewPrice>
  <LowestUsedPrice>
    <Amount>799</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <FormattedPrice>$7.99</FormattedPrice>
  </LowestUsedPrice>
  <TotalNew>45</TotalNew>
  <TotalUsed>20</TotalUsed>
  <TotalCollectible>0</TotalCollectible>
  <TotalRefurbished>0</TotalRefurbished>
</OfferSummary>
```

PromotionSummary Response Group

The PromotionSummary response group returns summary information about a promotion, including the type of promotion, beginning and ending dates of the promotion, the promotion ID, eligibility requirements, and text that describes the specifics of the promotion.

PromotionSummary must be used along with one of the following response groups:

- Large
- OfferFull
- Offers

An error is returned if ProductDetails is not accompanied by one of these response groups.

Promotion Types

Promotion types are returned by the Category element and include:

- **ForEachQuantityXGetQuantityFreeX —**When you purchase a specified number of items you receive some number of the same items for free. For example, for every six dozen eggs you buy you get a dozen eggs free.
- **BuyAmountXGetSimpleShippingFreeX—**The item is shipped free of charge.
- **BuyAmountXGetAmountOffX—**For a specified dollar amount you receive a discount off another item. For example, buy three balls and get 25% off of a baseball glove.

Relevant Operations

Operations that can use this response group include:

- ItemLookup
- ItemSearch
- SimilarityLookup

Response Elements

The following table describes and shows the parentage of the elements returned by PromotionSummary.

With the exception of Promotions, Promotion, and Summary, the ancestry of all elements in the table are Offers/Offer/Promotions/Promotion/Summary.

PromotionSummary also returns the elements that all response groups return, which described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of PromotionSummary.

- None

Child Response Group

The following response groups are children response groups of PromotionSummary.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by PromotionSummary.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
Operation=ItemLookup&
ContentType=text/xml&
AWSAccessKeyId=[AWS Access Key ID]&
ItemId=B000AQSMPO&
IdType=ASIN&
ResponseGroup=Offers,PromotionSummary&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by PromotionSummary.

```
<Promotions>
  <Promotion>
  <Summary>
    <PromotionId>A2QIQTNOFYRK5N</PromotionId>
    <Category>BuyAmountXGetAmountOffX</Category>
    <StartDate>2006 08 01 07:00:00 GMT</StartDate>
    <EndDate>2006 09 06 06:59:59 GMT</EndDate>
    <EligibilityRequirementDescription>Save $25.00 when you spend $125.00 or
more on Kitchen & Housewares or Bed & Bath products offered by Amazon.com. Enter
 code AUGSAVER at checkout.</EligibilityRequirementDescription>
    <BenefitDescription>Save $25.00 when you spend $125.00 or more on Kitchen
& Housewares or Bed & Bath products offered by Amazon.com. Enter code AUGSAVER
 at checkout.</BenefitDescription>
    <TermsAndConditions><STRONG>To receive the Best Value discount:</STRONG>
<OL> <LI>Add $125 of qualifying Kitchen & Housewares or Bed & Bath products
 to your Shopping Cart via the <STRONG>Add to Shopping Cart</STRONG> button on
 each respective product information page. <LI>At checkout, enter the promotional
 code and click the <STRONG>Apply</STRONG> button. <LI>The amount of the Best
Value savings ($25) will be reflected on the final order checkout page. <LI>If
 you remove any of the participating promotion items from your Shopping Cart
or violate any of the terms and conditions listed below, the promotion will be
 invalid, and the discount will be removed from the order. <LI>If you return
any of the items involved in the promotion, the discount previously applied to
 the order will be subtracted from the return credit. </LI></OL><BR><STRONG>Terms
 and Conditions:</STRONG> <UL> <LI>Promotional offer valid for a limited time
only. Items placed in Shopping Cart overnight may not be eligible for promotion
 at time of checkout. <LI>Promotion applies only to qualifying items displaying
 the offer message on their product information pages. Items that do not display
 the offer message do not qualify, regardless of the nature of the item. 
 All All Clad, Calphalon, Capresso, Emerilware, Le Creuset, Orrefors, Kosta
Boda, J.A. Henckels, Wusthof, Riedel, Marquis, Shun, Komachi, Waterford, Weber,
 Tassimo, Margaritaville, DKNY, Tommy Hilfiger, Nautica, Oscar by Oscar de
Larenta, Cath Kidston, Raymond Waites, and KitchenAid Proline products are
 excluded from this offer.  <LI>All Furniture & Decor and Outdoor Living
products are excluded from this offer. <LI>Applies only to products sold by the
 merchant indicated in the promotional offer message. Does not apply to items
```

```
sold by other merchants on the Amazon.com Web site. For example, where the
promotional offer applies to items offered by Amazon.com, items offered by other
 merchants on the Amazon.com Web site (e.g. Land's End, Target, etc.) do not
qualify. <LI>Does not apply to any products purchased in Amazon.com's other
sites, including Amazon.co.uk, Amazon.de, Amazon.fr, or Amazon.co.jp, or in
Marketplace, zShops, or Auctions. <LI>Unless the offer message indicates
otherwise, the promotional offer applies to the lowest priced qualifying item(s)
 <LI>Applies only when all qualifying and benefit products in the promotion are
 purchased in one order. <LI>Applies only to complete orders shipping to a
single address that meet all other promotional requirements. <LI>Offer good
while supplies last. <LI>No substitutions or rain checks. <LI>Offer must be
redeemed through the Shopping Cart. Does not apply to orders placed with 1
Click. <LI>Shipping and handling charges apply to all products, including bonus
 items. <LI>Offer may not be combined with other offers, including promotional
 certificates. <LI>Void where prohibited. </LI></UL></TermsAndConditions>
    </Summary>
  </Promotion>
</Promotions>
```

RelatedItems Response Group

The *RelatedItems* response group returns items related to an item specified in an ItemLookup (p. 200) request. .Related items could be, for example, all of the Unbox episodes in a TV season that are sold separately, or, for example, all of the MP3Download tracks on a MP3 album.

The data returned for *RelatedItems* is limited to ASINs and ItemAttributes. This remains true even if you add additional response groups, such as Large, that would otherwise return additional data.

The relationship between items is unidirectional. One item is the parent and one item is the child. Items, however, can have multiple children or multiple parents for a given relationship type.

The way in which the items are related is specified by the *RelationshipType* parameter. This parameter is required when you use the *RelatedItems* response group. Some values include Episode, Season, Tracks, and Variation. For a list of all relationship types, go to the ItemLookup (p. 200) page.

The relationship type is usually named after the child item in the relationship. For example, an MP3 Track is related to an MP3 album and the type of relationship is Tracks. In this relationship, the album is the parent. If you did an `ItemLookup` for an MP3 Track and requested *RelatedItems* using Tracks as the *RelationshipType*, you would receive the parent album (or albums) for that Track. Conversely, looking up an album using Tracks as the *RelationshipType* returns the list of Tracks on that album.

Each `ItemLookup` request can return, at most, ten related items. To return additional items, use the *RelateditemsPage* parameter. A value of 2, for example, returns the second set of ten related items.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)

Response Elements

The following elements are returned by *RelatedItems*:

- ItemAttributes (p. 315)

- RelatedItems (p. 320)

RelatedItems also returns the elements that all response groups return. For more information, go to Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of *RelatedItems*.

- None

Child Response Group

The following response groups are child response groups of RelatedItems.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by *RelatedItems*.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
RelationshipType=Tracks&
ItemId=B0013D8EQK&
ItemType=ASIN&
ResponseGroup=RelatedItems,Small&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by RelatedItems.

```
<Item>
  <ASIN>B0013D8EQK</ASIN>
  <ItemAttributes>
    <Creator Role="Primary Contributor">Johnny Cash</Creator>
    <Manufacturer>Columbia/Legacy</Manufacturer>
    <ProductGroup>Digital Music Album</ProductGroup>
    <Title>At San Quentin</Title>
  </ItemAttributes>
  <RelatedItems>
    <Relationship>Children</Relationship>
    <RelationshipType>Tracks</RelationshipType>
    <RelatedItemCount>31</RelatedItemCount>
    <RelatedItemPageCount>4</RelatedItemPageCount>
    <RelatedItemPage>1</RelatedItemPage>
    <RelatedItem>
      <Item>
```

```
        <ASIN>B0013D4KJK</ASIN>
        <ItemAttributes>
          <Creator Role="Primary Contributor">Johnny Cash</Creator>
          <Manufacturer>Columbia/Legacy</Manufacturer>
          <ProductGroup>Digital Music Track</ProductGroup>
          <Title>The Long Black Veil/Give My Love To Rose (Live)</Title>
        </ItemAttributes>
      </Item>
    </RelatedItem>
    <RelatedItem>
      <Item>
        <ASIN>B0013D7VG4</ASIN>
        <ItemAttributes>
          <Creator Role="Primary Contributor">Johnny Cash</Creator>
          <Manufacturer>Columbia/Legacy</Manufacturer>
          <ProductGroup>Digital Music Track</ProductGroup>
          <Title>Folsom Prison Blues (Live)</Title>
        </ItemAttributes>
      </Item>
    </RelatedItem>
    ...
  </RelatedItems>
</Item>
```

Request Response Group

The Request response group returns all of the parameters and their values that were submitted in a request. Use this information to debug requests.

All Product Advertising API operations return this response group by default. There can be up to ten parameters in each request.

Relevant Operations

Operations that can use this response group include:

- All Product Advertising API operations use the Request response group by default.

Response Elements

The following table describes the elements returned by Request. These elements are returned in all Product Advertising API responses because the Request response group is a default response group for all Product Advertising API operations.

- Code (p. 310)
- IsValid (p. 315)
- Message (p. 317)
- Name (p. 317)
- RequestId (p. 320)
- UserAgent (p. 324)

Parent Response Group

The following response groups are parent response groups of Request.

- None

Child Response Group

The following response groups are child response groups of Request.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by Request.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
SearchIndex=Books&
Title=Harry%20Potter&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Request.

```
<OperationRequest>
  <HTTPHeaders>
    <Header Name="UserAgent" Value="Mozilla/4.0 (compatible; MSIE 6.0; Windows
 NT 5.1; SV1; FunWebProducts; .NET CLR 1.1.4322; .NET CLR 2.0.50727)" />
  </HTTPHeaders>
  <RequestId>00KE1E5MWR4KXX0V1WYD</RequestId>
  <Arguments>
    <Argument Name="SearchIndex" Value="Books" />
    <Argument Name="Service" Value="AWSECommerceService" />
    <Argument Name="Title" Value="Harry Potter" />
    <Argument Name="Operation" Value="ItemSearch" />
    <Argument Name="AWSAccessKeyId" Value="[VALUE]" />
    <Argument Name="Version" Value="2011-08-01" />
  </Arguments>
  <RequestProcessingTime>
    0.3419508934021
  </RequestProcessingTime>
</OperationRequest>
  <Items>
    <Request>
      <IsValid>True</IsValid>
      <ItemSearchRequest>
        <SearchIndex>Books</SearchIndex>
        <Title>Harry Potter</Title>
```

```
    </ItemSearchRequest>
  </Request>
```

Reviews Response Group

The reviews response group returns the URL to an iframe that contains customer reviews. You can embed the iframe on any web page to display the response content.

Important

Each iframe URL is valid for 24 hours. If the iframe URL expires, you will receive a `403 Forbidden` error code.

Note

As of November 8, 2010, only the iframe URL is returned in the request content.

For more information about reviews, see Getting Customer Reviews (p. 122).

Each customer review contains the following:

- Rating
- Summary
- Date of review
- Text of the review

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following elements are returned by Reviews:

- IFrameURL (p. 314)

Reviews also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of Reviews.

- None

Child Response Group

The following response groups are child response groups of Reviews.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by Reviews.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=0316067938&
ResponseGroup=Reviews&
TruncateReviewsAt="256"&
IncludeReviewsSummary="False"&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Reviews.

```
<ItemLookupResponse>
 <OperationRequest>
  <RequestId>[Request ID]</RequestId>
  <Arguments>
   <Argument Name="Operation" Value="ItemLookup"/>
   <Argument Name="Service" Value="AWSECommerceService"/>
   <Argument Name="ItemId" Value="0316067938"/>
   <Argument Name="TruncateReviewsAt" Value="256"/>
   <Argument Name="AWSAccessKeyId" Value="[AWS Access Key ID]"/>
   <Argument Name="IncludeReviewsSummary" Value="False"/>
   <Argument Name="ResponseGroup" Value="Reviews"/>
   <Argument Name="Version" Value="2010-09-01"/>
  </Arguments>
  <RequestProcessingTime>0.0416880000000000</RequestProcessingTime>
 </OperationRequest>
 <Items>
  <Item>
   <ASIN>0316067938</ASIN>
   <CustomerReviews>
    <IFrameURL>http://www.amazon.com/reviews/iframe?akid=[AWS Access Key
ID]&asin=0316067938&exp=2010-09-
02T17%3A54%3A07Z&linkCode=xm2&summary=0&tag=ws&truncate=256&v=2&sig=[Signature]</IFrameURL>

   </CustomerReviews>
  </Item>
 </Items>
</ItemLookupResponse>
```

SalesRank Response Group

The SalesRank (p. 282) response group returns the sales rank for each item in the response. One is the highest rating; a large number means the item has not sold well. Sales rank is per Search Index so it is possible to have several items in one response ranked, for example, as 1.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by SalesRank.

- ASIN (p. 308)
- SalesRank (p. 320)
- TotalPages (p. 323)
- TotalResults (p. 323)

SalesRank also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of SalesRank.

- None

Child Response Group

The following response groups are child response groups of SalesRank.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by SalesRank.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=0976925524&
ResponseGroup=SalesRank&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by SalesRank.

```
<Item>
  <ASIN>0976925524</ASIN>
  <SalesRank>68</SalesRank>
</Item>
```

SearchBins Response Group

The SearchBins response group groups the items returned by ItemSearch (p. 207) into bins. A set of bins, for example, can be a set of price ranges for a product. In the case of women's shoes, for example, you might have a bin that returns ASINs for shoes that cost between $0 and $50, a second bin for shoes that cost $50 to $100, and a third bin for shoes that cost more than $100. The basis on which the items are split into bins is specified by the NarrowBy (p. 283) attribute in the SearchBinSet tag. To refine the search, you make repeated requests using the NarrowBy (p. 283) values.

`ItemSearch` returns the first twenty-five bins of results.

For more information about search bins, see Using Search Bins to Find Items (p. 99).

Availability

US locale only.

NarrowBy

The basis on which the items are split into bins is specified by the NarrowBy (p. 283) attribute in the SearchBinSet (p. 321) tag. In the following example, the NarrowBy (p. 283) attribute shows that the bins are based on price range:

```
<SearchBinSet NarrowBy="PriceRange">
```

For another product category, the NarrowBy (p. 283) attribute might be different, for example:

```
<SearchBinSet NarrowBy="BrandName">
```

You cannot specify NarrowBy (p. 283) values nor can you specify the values they encompass. When SearchBins is included as a Response Group in a request, `ItemSearch` automatically divides the `ItemSearch` results into bins.

NarrowBy Values

NarrowBy values include:

- Subject—BrowseNode IDs of all topics related to items returned by `ItemSearch`. For example, searching for books about dogs returns, in the Subject bins, BrowseNodes for "Home & Garden," "Animal Care & Pets," "Dogs," and "Educational."
- BrandName—Brands, such as Levi's, Reebok, and Nike, that create the item. Use the name of a brand to filter out similar items made by other companies.
- PriceRange—Minimum and maximum prices for a bin of items. Use the minimum and maximum price values in each bin to filter out items outside of the price range you want.

- SpecialSize—Uncommon sizes an item comes in. Examples are "Plus Size & Tall," "Misses," "Maternity," "Husky," "Petites," and "Big & Tall."

All Search Indices return the Subject bin. Most return all of the bins.

NarrowBy Values by Search Index

The following list shows which NarrowBy values are returned by each Search Index:

- Apparel—Subject, BrandName, PriceRange, SpecialSize
- Baby—Subject, BrandName, PriceRange, SpecialSize
- Beauty—Subject, BrandName, PriceRange, SpecialSize
- Blended—Not supported
- Books—Subject
- Classical—Subject
- DVD—Subject
- Electronics—Subject, BrandName, PriceRange, SpecialSize
- Garden—Subject, BrandName, PriceRange, SpecialSize
- GourmetFood—Subject, BrandName, PriceRange, SpecialSize
- HealthPersonalCare—Subject, BrandName, PriceRange, SpecialSize
- Jewelry—Subject, Brand, PriceRange, SpecialSize
- Kitchen—Subject, BrandName, PriceRange, SpecialSize
- Magazines—Subject
- Marketplace—Subject
- Miscellaneous—Subject, BrandName, PriceRange, SpecialSize
- Music—Subject
- MusicalInstruments—Subject, BrandName, PriceRange, SpecialSize
- OfficeProducts—Subject, BrandName, PriceRange, SpecialSize
- PCHardware—Subject, BrandName, PriceRange, SpecialSize
- Photo—Subject, BrandName, PriceRange, SpecialSize
- Showtimes—Theater
- Software—Subject, BrandName, PriceRange, SpecialSize
- SportingGoods—Subject, BrandName, PriceRange, SpecialSize
- Theatrical—Subject
- Tools—Subject, BrandName, PriceRange, SpecialSize
- Toys—Subject, BrandName, PriceRange, SpecialSize
- Travel—Subject
- VHS—Subject
- Video—Subject
- VideoGames—Subject, BrandName, PriceRange, SpecialSize
- Wireless—Subject
- WirelessAccessories—Subject

Element Tags in a Bin

The element tags in a bin vary according to the bin. For example, in bins based on price, the elements and Name values are:

```
<BinName>
 <BinItemCount>
 <BinParameter>
    <Name>MinimumPrice</Name>
  <Value>
 <BinParameter>
  <Name>MaximumPrice</Name>
  <Value>
```

The elements show the minimum and maximum price for items in that bin, and the number of items in that bin.

Other NarrowBy values use other element tags in their bins. For example, when NarrowBy is “Brand,” the element tags and Name values are:

```
<Bin>
 <BinName>
 <BinItemCount>
 <BinParameter>
  <Name>Brand</Name>
  <Value>
```

You cannot specify the element tags returned in a bin.

Drilling Down

You can take the values in a bin and add them to the `ItemSearch` query to filter out of the response items that are outside of that bin. To narrow the search results to shirts that cost between $0 and $25, for example, you would add the following Name and its corresponding value as additional parameters in the original ItemSearch request:

```
&MinimumPrice=0
&MaximumPrice=2499
```

You could narrow the search results even further by adding an additional parameter to the query. For example, you could specify

```
&Brand=Levi’s
```

The response would then only include shirts by Levi’s that cost under $25. You could continue to drill down by adding additional parameters to the request.

Notice that in each case the results were narrowed down by adding the value tagged with Name and its corresponding value to the original `ItemSearch` request.

Relevant Operations

Operations that can use this response group include:

- ItemSearch (p. 207)

Response Elements

The following table describes the elements returned by SearchBins.

- BinItemCount (p. 309)
- BinName (p. 309)
- Name (p. 317)
- SearchBinSets (p. 321)

SearchBins also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of SearchBins.

- None

Child Response Group

The following response groups are child response groups of SearchBins.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by SearchBins.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=All&
SearchIndex=SportingGoods&
Keywords=Glove&
ResponseGroup=SearchBins&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by SearchBins.

```
<Item>
  <ASIN>B00005R2GR</ASIN>
</Item>
<Item>
  <ASIN>B00076ZDV8</ASIN>
</Item>
<Item>
  <ASIN>B00092FEEG</ASIN>
</Item>
<Item>
  <ASIN>B000ADTP1W</ASIN>
</Item>
```

```
<SearchBinSets>
  <SearchBinSet NarrowBy="PriceRange">
    <Bin>
        <BinName>$25-$49</BinName>
        <BinItemCount>316</BinItemCount>
    <BinParameter>
        <Name>MinimumPrice</Name>
        <Value>2500</Value>
     </BinParameter>
    <BinParameter>
        <Name>MaximumPrice</Name>
        <Value>4999</Value>
      </BinParameter>
    </Bin>
  </SearchBinSet>
    <Bin>
      <BinName>$0-$24</BinName>
      <BinItemCount>280</BinItemCount>
    <BinParameter>
      <Name>MinimumPrice</Name>
      <Value>0</Value>
      </BinParameter>
    <BinParameter>
      <Name>MaximumPrice</Name>
      <Value>2499</Value>
      </BinParameter>
      </Bin>
 </SearchBinSet>

<SearchBinSet NarrowBy="BrandName">
  <Bin>
    <BinName>Rawlings</BinName>
    <BinItemCount>71</BinItemCount>
  <BinParameter>
    <Name>Brand</Name>
    <Value>Rawlings</Value>
    </BinParameter>
    </Bin>
  <Bin>
    <BinName>Body Glove</BinName>
    <BinItemCount>53</BinItemCount>
  <BinParameter>
    <Name>Brand</Name>
    <Value>Body Glove</Value>
    </BinParameter>
    </Bin>
</SearchBinSet>

<SearchBinSet NarrowBy="Subject">
  <Bin>
    <BinName>Categories</BinName>
    <BinItemCount>860</BinItemCount>
  <BinParameter>
    <Name>BrowseNode</Name>
    <Value>3375301</Value>
    </BinParameter>
    </Bin>
  <Bin>
```

```
    <BinName>Baseball</BinName>
    <BinItemCount>311</BinItemCount>
  <BinParameter>
    <Name>BrowseNode</Name>
    <Value>3395731</Value>
    </BinParameter>
    </Bin>
</SearchBinSet>
```

Notice that the NarrowBy (p. 283) value is PriceRange.

Similarities Response Group

The Similarities response group returns titles and ASINs of five items that are similar to the one specified in the request.

This response group is often used in conjunction with ItemLookup (p. 200) to return items that are similar to the one specified in the request.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by Similarities.

- ASIN (p. 308)
- Title (p. 322)
- TotalPages (p. 323)
- TotalResults (p. 323)

Similarities also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of Similarities.

- None

Child Response Group

The following response groups are child response groups of Similarities.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by Similarities.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKey=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=All&
SearchIndex=Blended&
Keywords=Mustang&
Merchant=All&
ResponseGroup=Similarities
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Similarities.

```
<SimilarProduct>
  <ASIN>B00004GJVO</ASIN>
  <Title>Minor Move</Title>
</SimilarProduct>
```

Small Response Group

The Small response group returns basic information about items in a response. The information includes the item's ASIN, title, product group, and author.

This response group is often used with ItemLookup (p. 200), ItemSearch (p. 207), or SimilarityLookup (p. 218) to return basic information about the items in the response.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by Small.

- Actor (p. 307)
- Artist (p. 308)
- ASIN (p. 308)
- Author (p. 308)
- CorrectedQuery (p. 311)
- Creator (p. 311)
- Director (p. 312)

- Keywords (p. 315)
- Manufacturer (p. 316)
- Message (p. 317)
- ProductGroup (p. 319)
- Role (p. 320)
- Title (p. 322)
- TotalPages (p. 323)
- TotalResults (p. 323)

Small also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of Small.

- Large (p. 250)
- Medium (p. 255)

Child Response Group

The following response groups are child response groups of Small.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by Small.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKey=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=All&
SearchIndex=Blended&
Keywords=Mustan&
Merchant=All&
ResponseGroup=Small&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Small.

```
<CorrectedQuery>
  <Keywords>mustang</Keywords>
  <Message>We found no matches for (keywords=Mustan). Below are results for
(keywords=mustang).</Message>
 </CorrectedQuery>
```

```
<Item>
  <ASIN>B0002Y69UO</ASIN>
<ItemAttributes>
    <Actor>America's Favorite Cars</Actor>
    <ProductGroup>DVD</ProductGroup>
    <Title>America's Favorite Cars - The Complete Mustang 40th
Anniversary</Title>
  </ItemAttributes>
</Item>
```

Notice that the Keywords (p. 315) value, Mustan, was considered misspelled. The search results were for the word, Mustang, instead.

TopSellers Response Group

The TopSellers response group returns the ASINs and titles of the ten best sellers within a specified browse node.

Availability

This response group is available in all locales.

Relevant Operations

Operations that can use this response group include:

- BrowseNodeLookup (p. 177)

Response Elements

The following table describes the elements returned by TopSellers.

- Actor (p. 307) for the Video search index
- Artist (p. 308) for the Music search index
- ASIN (p. 308)
- Authors (p. 308) for the Books search index
- ProductGroup (p. 319)
- Title (p. 322)
- TopItemSet (p. 322)

TopSellers also returns the elements that all response groups return, which is described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of TopSellers.

- None

Child Response Group

The following response groups are child response groups of TopSellers.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by TopSellers.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=BrowseNodeLookup&
BrowseNodeId=20&
ResponseGroup=TopSellers&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by TopSellers.

```
<TopItemSet>
   <Type>TopSellers</Type>
   <TopItem>
            <ASIN>0553576399</ASIN>
            <Title>Distraction</Title>
            <ProductGroup>Book</ProductGroup>
            <Author>Bruce Sterling</Author>
       </TopItem>
       <TopItem>
        ....
       .....
</TopItemSet>
```

Tracks Response Group

The Tracks response group returns the title and number of each track on each CD in the response. For example, you could use ItemLookup (p. 200) to return Tracks information about a specified CD.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by Tracks.

- Number (p. 318)
- TotalPages (p. 323)
- TotalResults (p. 323)
- Track (p. 323)

Tracks also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of Tracks.

- None

Child Response Group

The following response groups are child response groups of Tracks.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by Tracks.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
Condition=All&
SearchIndex=Blended&
Keywords=GodSmack&
Merchant=All&
ResponseGroup=Tracks&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Tracks.

```
<Item>
  <ASIN>B000EXOAAO</ASIN>
  <Tracks>
    <Disc Number="1">
      <Track Number="1">Livin In Sin</Track>
      <Track Number="2">Speak</Track>
      <Track Number="3">The Enemy</Track>
```

```
      <Track Number="4">Shine Down</Track>
      <Track Number="5">Hollow</Track>
      <Track Number="6">No Rest For The Wicked</Track>
      <Track Number="7">Bleeding Me</Track>
      <Track Number="8">Voodoo Too</Track>
      <Track Number="9">Temptation</Track>
      <Track Number="10">Mama</Track>
      <Track Number="11">One Rainy Day</Track>
    </Disc>
  </Tracks>
</Item>
```

Variations Response Group

The Variations response group is a parent response group that returns the contents of the VariationSummary response group plus other variation details, such as item attributes, offers, and offer listings for each variation in the response.

Variation Dimensions

A variation is a child ASIN. The parent ASIN is an abstraction of the children items. For example, Shirt is a parent ASIN. Parent ASINs cannot be sold. A child ASIN of it would be a blue shirt, size 16, sold by MyApparelStore. This child ASIN is one of potentially many variations. The ways in which variations differ are called dimensions. In the preceding example, size and color are the dimensions. Parent ASINs therefore return two related elements:

- VariationDimensions
- VariationDimension

For example,

```
<VariationDimensions>
  <VariationDimension>ClothingSize</VariationDimension>
  <VariationDimension>Color</VariationDimension>
</VariationDimensions>
```

The values returned by these elements are the dimensions listed in the child ASIN's response, for example,

```
<Item>
  ...
  <ItemAttributes>
   ...
  </ItemAttributes>
  <VariationAttributes>
    <VariationAttribute>
      <Name>Color</Name>
      <Value>Black<Value>
    </VariationAttribute>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>Large</Value>
    </VariationAttribute>
  </VariationAttributes>
```

```
  ...
</Item>
```

The following lists some of the Product Advertising API variation dimensions:

- GemType (string)
- HandOrientation (string)
- HardwarePlatform (string)
- PackageQuantity (nonNegativeInteger)
- ItemDimensions/Length (DecimalWithUnits)
- ItemDimensions/Width (DecimalWithUnits)
- LensColor (string)
- MetalType (string)
- Model (string)
- NumberOfLicenses (string)
- OperatingSystem (string)
- ProductTypeSubcategory (string)
- ScentName (string) StyleName (string)
- TotalDiamondWeight (DecimalWithUnits)
- TotalGemWeight (DecimalWithUnits)

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Response Elements

The following table describes the elements returned by Variations.

- Amount (p. 308)
- ASIN (p. 308)
- CurrencyCode (p. 311)
- FormattedPrice (p. 313)
- GolfClubFlex
- GolfClubLoft

Variations also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of Variations.

- None

Child Response Group

The following response groups are child response groups of Variations.

- VariationSummary (p. 304)

Sample REST Use Case

The following request illustrates the XML response elements returned by Variations.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
SearchIndex=Apparel&
Keywords=Hooded%20Short%20Down%20Jacket&
ResponseGroup=Variations&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by Variations.

```
<ItemSearchRequest>
  <Keywords>Hooded Short Down Jacket</Keywords>
  <ResponseGroup>Variations</ResponseGroup>
  <SearchIndex>Apparel</SearchIndex>
</ItemSearchRequest>
</Request>
<Item>
  <ASIN>B000CCIIT6</ASIN>
</Item>
<Item>
  <ASIN>B000CCIITQ</ASIN>
</Item>
<Item>
  <ASIN>B000CCONPO</ASIN>
</Item>
<Item>
  <ASIN>B000CCMQ1W</ASIN>
</Item>
<Item>
  <ASIN>B000CDDRJ6</ASIN>
</Item>
<Item>
  <ASIN>B000CCIISC</ASIN>
</Item>
<Item>
  <ASIN>B000CCIIRI</ASIN>
</Item>
<Item>
  <ASIN>B000BVA9AE</ASIN>
</Item>
```

```
<Item>
  <ASIN>B000CCMQFS</ASIN>
</Item>
<Item>
  <ASIN>B000CDDRJG</ASIN>
</Item>
</Items>
```

This response shows that the ASIN in the response has ten variations.

VariationImages Response Group

The VariationImages response group displays different image variations of the same item in four sizes: swatch, small, medium, and large, where the swatch image is smaller than the small image. `VariationImages` is used when there are variations of an item for sale. For example, a shirt for sale might come in four colors. If VariationImages is used as the response group, the shirt will be displayed in sixteen images: four image sizes of each of the four colors.

Parent ASINs do not always have images associated with them; their children, however, should.

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)

Response Elements

The following table describes the elements returned by VariationImages. The elements on the left side of a slash mark are the parents of the elements on the right side of the slash mark.

- Height (p. 314)
- LargeImage (p. 316)
- MediumImage (p. 317)
- SmallImage (p. 321)
- SwatchImage (p. 322)
- ThumbnailImage (p. 322)
- TinyImage (p. 322)
- URL (p. 324)
- Width (p. 325)

VariationImages also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of VariationImages.

- None

Child Response Group

The following response groups are child response groups of VariationImages.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by VariationImages.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=0239409223&
ResponseGroup=VariationImages
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by VariationImages.

```
<ImageSets>
  <ImageSet Category="primary">

  <SmallImage>
    <URL>
    http://images.amazon.com/images/P/B99999999A.01._SCTHUMBZZZ_.jpg
    </URL>
    <Height Units="pixels">60</Height>
    <Width Units="pixels">60</Width>
  </SmallImage>

  <MediumImage>
    <URL>
    http://images.amazon.com/images/P/B99999999A.01._SCMZZZZZZZ_.jpg
    </URL>
    <Height Units="pixels">140</Height>
    <Width Units="pixels">140</Width>
  </MediumImage>

  <LargeImage>
    <URL>
    http://images.amazon.com/images/P/B99999999A.01._SCLZZZZZZZ_.jpg
    </URL>
    <Height Units="pixels">500</Height>
    <Width Units="pixels">500</Width>
    </LargeImage>
    </ImageSet>
  <ImageSet Category="variant">
```

```
  <SmallImage>
    <URL>
    http://images.amazon.com/images/P/B99999999A.01.PT01._SCTHUMBZZZ_.jpg
    </URL>
    <Height Units="pixels">48</Height>
    <Width Units="pixels">60</Width>
  </SmallImage>

  <MediumImage>
    <URL>
    http://images.amazon.com/images/P/B99999999A.01.PT01._SCMZZZZZZZ_.jpg
    </URL>
    <Height Units="pixels">120</Height>
    <Width Units="pixels">150</Width>
  </MediumImage>

  <LargeImage>
    <URL>
    http://images.amazon.com/images/P/B99999999A.01.PT01._SCLZZZZZZZ_.jpg
    </URL>
    <Height Units="pixels">400</Height>
    <Width Units="pixels">500</Width>
  </LargeImage>

</ImageSet>
```

VariationMatrix Response Group

The VariationMatrix response group returns, for a given parent ASIN, the variation dimension name and value of each child ASIN. If a returned item does not have variations, the VariationMatrix response group will not return any data. For example, rings vary by ring size. RingSize, therefore, is the dimension related to this item. The VariationMatrix response group would only return, in this example, <RingSize> elements. The VariationMatrix response group returns nothing if the returned items do not have variations. For more information, see, Returning Variation Dimension Information Only (p. 42).

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)

Response Elements

The following table describes the elements returned by VariationMatrix.

- ClothingSize (p. 310)
- Color (p. 310)
- GemType
- GolfClubFlex
- GolfClubLoft
- HardwarePlatform (p. 313)

- ItemDimensions/Length (p. 316)
- ItemDimensions/Width (p. 325)
- MaterialType (p. 317)
- MetalType (p. 317)
- Model (p. 317)
- OperatingSystem (p. 318)
- PackageQuantity (p. 318)
- ProductTypeSubcategory (p. 319)
- Size (p. 321)
- TotalDiamondWeight
- TotalGemWeight
- VariationDimension (p. 325)

VariationMatrix also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of VariationMatrix.

- None

Child Response Group

The following response groups are child response groups of VariationMatrix.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by VariationMatrix.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemLookup&
ItemId=B0008G23PQ&
Condition=All&
ResponseGroup=VariationMatrix&
Version=2011-08-01
```

Sample Response Snippet

The following response snippet shows the elements returned by VariationMatrix.

```
<Item>
  <ASIN>B0008G23PQ</ASIN>
  <Variations>
  <VariationDimensions>
    <VariationDimension>ClothingSize</VariationDimension>
```

```
    <VariationDimension>Color</VariationDimension>
  </VariationDimensions>
<Item>
  <ASIN>B0008EOA9U</ASIN>
  <VariationAttributes>
    <VariationAttribute>
      <Name>Color</Name>
      <Value>Grey<Value>
    </VariationAttribute>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>29W x 30L</Value>
    </VariationAttribute>
  </VariationAttributes>
</Item>
<Item>
  <ASIN>B0008EO9J6</ASIN>
  <VariationAttributes>
    <VariationAttribute>
      <Name>Color</Name>
      <Value>Navy<Value>
    </VariationAttribute>
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>29W x 30L</Value>
    </VariationAttribute>
  </VariationAttributes>
</Item>
```

VariationOffers Response Group

The VariationOffers response group enables you to retrieve the offers for the children of a parent ASIN. VariationOffers is similar to the Variations response group, however, item attributes are not returned for the individual variations.Variations is the parent response group of VariationOffers.

The following table shows the expected behavior of the parent and child variation offers for the corresponding MerchantId input parameter:

Merchant ID	Semantics
(Unspecified)	All variations with offers
Amazon	Variations with Amazon offers only
Featured, Featured Buy Box Winner, or All	All variations with offers. The specified MerchantId parameter will not have any effect on the response.
Any other value	Error: invalid value

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)

Response Elements

The following table describes the elements returned by VariationOffers. In the Ancestry column, In the Ancestry column, the elements on the left side of a slash mark are the parents of the elements on the right side of the slash mark.

VariationOffers also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of VariationOffers.

- Variations (p. 304)

Child Response Group

The following response groups are child response groups of VariationOffers.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by VariationOffers.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=[AWS Access Key ID]&
Operation=ItemSearch&
ItemId=B000P4VW1M&
Condition=All&
ResponseGroup=VariationOffers&
```

```
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by VariationOffers.

```
<Item>
  <ASIN>B000P4VW1M</ASIN>
  <VariationSummary>
    <LowestPrice>
      <Amount>12900</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$129.00</FormattedPrice>
    </LowestPrice>
    <HighestPrice>
      <Amount>12900</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$129.00</FormattedPrice>
    </HighestPrice>
  </VariationSummary>
  <Variations>
    <TotalVariations>5</TotalVariations>
    <TotalVariationPages>1</TotalVariationPages>
    <Item>
      <ASIN>B000N637Y2</ASIN>
      <Offers>
        <Offer>
          <Merchant>
            <Name>Amazon.com</Name>
          </Merchant>
          <OfferAttributes>
            <Condition>New</Condition>
          </OfferAttributes>
          <OfferListing>

<OfferListingId>ct%2FbvUk%2F9aRAxMyfX4EnsglywwdEGTLsR7XZwmRt9Aw%2Fe4yKOeVDEIdVSBUhBFoR4VZwR1aLBZ%2FhzSDQ%3D%3D</OfferListingId>

            <Price>
              <Amount>12900</Amount>
              <CurrencyCode>USD</CurrencyCode>
              <FormattedPrice>$129.00</FormattedPrice>
            </Price>
            <AmountSaved>
              <Amount>28428</Amount>
              <CurrencyCode>USD</CurrencyCode>
              <FormattedPrice>$284.28</FormattedPrice>
            </AmountSaved>
            <PercentageSaved>69</PercentageSaved>
            <Availability>In stock soon. Order now to get in line. First come,
 first served.</Availability>
            <AvailabilityAttributes>
              <AvailabilityType>unknown</AvailabilityType>
              <MinimumHours>672</MinimumHours>
              <MaximumHours>1008</MaximumHours>
            </AvailabilityAttributes>
```

```
          <IsEligibleForSuperSaverShipping>1</IsEligibleForSuperSaverShipping>

         </OfferListing>
        </Offer>
    </Offers>
</Item>
```

VariationSummary Response Group

The VariationSummary response group provides the lowest price, highest price, lowest sale price, and highest sale price for all child ASINs in a response.

Parent ASINs do not have offers; their children do. For example, you cannot buy a shirt (the parent ASIN). You can, however, buy a shirt that is a certain color and size (the child ASIN).

Relevant Operations

Operations that can use this response group include:

- ItemLookup (p. 200)
- ItemSearch (p. 207)
- SimilarityLookup (p. 218)

Variation Dimensions

A variation is a child ASIN. The parent ASIN is an abstraction of the children items. For example, Shirt is a parent ASIN. Parent ASINs cannot be sold. A child ASIN of it would be a blue shirt, size 16, sold by MyApparelStore. This child ASIN is one of potentially many variations. The ways in which variations differ are called dimensions. In the preceding example, size and color are the dimensions. Parent ASINs therefore return two related elements:

- VariationDimensions
- VariationDimension

For example,

```
<VariationDimensions>
  <VariationDimension>ClothingSize</VariationDimension>
  <VariationDimension>Color</VariationDimension>
</VariationDimensions>
```

The values returned by these elements are the dimensions listed in the child ASIN's response, for example,

```
<Item>
  <ItemAttributes>
   ...
  </ItemAttributes>
  <VariationAttributes>
    <VariationAttribute>
      <Name>Color</Name>
      <Value>Black<Value>
    </VariationAttribute>
```

```
    <VariationAttribute>
      <Name>ClothingSize</Name>
      <Value>Large</Value>
    </VariationAttribute>
  </VariationAttributes>
  ...
</Item>
```

Response Elements

The following table describes the elements returned by VariationSummary.

- Amount (p. 308)
- CurrencyCode (p. 311)
- FormattedPrice (p. 313)

VariationSummary also returns the elements that all response groups return, as described in Elements Common to All Response Groups (p. 306).

Parent Response Group

The following response groups are parent response groups of VariationSummary.

- None

Child Response Group

The following response groups are child response groups of VariationSummary.

- None

Sample REST Use Case

The following request illustrates the XML response elements returned by VariationSummary.

```
http://ecs.amazonaws.com/onca/xml?
Service=AWSECommerceService&
AWSAccessKeyId=<replaceable>
<replaceable>[AWS Access Key ID]</replaceable>
  </replaceable>&
Operation=ItemLookup&
ItemId=B00006XYAA&
ResponseGroup=VariationSummary&
Version=2011-08-01
&Timestamp=[YYYY-MM-DDThh:mm:ssZ]
&Signature=[Request Signature]
```

Sample Response Snippet

The following response snippet shows the elements returned by VariationSummary.

```
<Item>
  <ASIN>B00006XYAA</ASIN>
  <VariationSummary>
    <LowestPrice>
      <Amount>1450</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$14.50</FormattedPrice>
    </LowestPrice>
    <HighestPrice>
      <Amount>1750</Amount>
      <CurrencyCode>USD</CurrencyCode>
      <FormattedPrice>$17.50</FormattedPrice>
    </HighestPrice>
  </VariationSummary>
</Item>
```

Response Elements Common to All Response Groups

The following table describes and shows the parentage of the elements returned by all response groups.

Response Element	Definition
ASIN	An token distributed by Amazon that uniquely identifies an item. Type: String Ancestry: Item/ASIN
Code	Error code if there is an error Type: Integer Ancestry: Errors/Error/Code
IsValid	Is True if the request is valid Type: String Ancestry: Request/IsValid
Message	Error message that corresponds with error code Type: String Ancestry: Errors/Error/Message
Name	Name of a parameter in the request Type: String Ancestry: Arguments/Argument/Name
RequestId	Unique number that identifies the request Type: String Ancestry: OperationRequest/RequestId
TotalPages	Number of pages found. There are up to ten items per page. Type: String Ancestry: List/TotalPages

Response Element	Definition
TotalResults	The total number of items found. Up to ten are returned per request. . By default, the first ten items are returned. Type: String Ancestry: List/TotalResults
UserAgent	The name and version of the Web browser Type: String Ancestry: OperationRequest/UserAgent
Value	A value of a parameter in the request Type: String Ancestry: Arguments/Argument/Value

Response Elements

This chapter provides a description of all response elements. In the Ancestry paragraphs, the elements on the left side of a slash mark are the parents of the elements on the right side of the slash mark.

Response Element	Definition
`About`	Describes the seller. Ancestry: Seller/About
`AboutMe`	Information a customer supplies about themselves Ancestry: Seller/About
`Actor`	Actor associated with the item. Ancestry: ItemAttributes
`AdditionalName`	For baby registries, this field is used for the name of the other parent. Ancestry: List/AdditionalName
`AlternateVersion`	Container for AlternateVersion information, including ASIN, Title, and Binding. Ancestry: None

Response Element	Definition
`Amount`	Price in terms of the lowest currency denomination, for example, pennies. The price, in terms of the lowest currency denomination, for example, pennies, of an item in the cart. The total price, in terms of the lowest currency denomination, of one or more of the same item in the Saved For Later area. If, for example, Saved For Later contains two orders of the same book, this amount would be the sum total of those two books. Ancestry: OfferSummary/LowestCollectiblePrice OfferSummary/LowestNewPrice OfferSummary/LowestRefurbishedPrice/Amount Offers/Offer/OfferListing/Price/Amount Offers/Offer/OfferListing/SalePrice/Amount Transaction/TransactionItems/TransactionItem/UnitPrice VariationSummary/LowestPrice VariationSummary/HighestSalePrice
`Artist`	Artist associated with the item. Ancestry: ItemAttributes
`ASIN`	A positive integer distributed by Amazon that uniquely identifies an item. ASIN of the new release, item, or similar item Ancestry: Item/ASIN CartItem SavedForLaterItem Cart/NewReleases/NewRelease NewReleases/NewRelease/ASIN OtherCategoriesSimilarProduct/ASIN SimilarProduct SimilarViewedProduct, CustomerReviews/Review
`AspectRatio`	The ratio of an item's length to its width. Ancestry: ItemAttributes
`AudienceRating`	Audience rating for a movie. The rating suggests the age for which the movie is appropriate. The rating format varies by locale. Ancestry: ItemAttributes
`AudioFormat`	Format, such as MP3, of the audio media. Ancestry: ItemAttributes/Languages/Language
`Author`	Author associated with the item. Ancestry: ItemAttributes
`Availability`	How soon the item can be shipped. Ancestry: Offers/Offer/OfferListing/Availability
`AvailabilityAttributes`	Container for availability information, including AvailabilityType, MaximumHours and MinimumHours. Ancestry: Offers/Offer/OfferListing
`Benefit`	Container for information relating to a promotional benefit. Ancestry: Benefits
`Benefits`	Container for one or more Benefit elements. Ancestry: None

Response Element	Definition
`BenefitType`	Specifies the type of promotion, for example, Free, FixedAmountOff, PercentOff, and TieredFixedAmountOff. Ancestry: Benefits/Benefit/BenefitType
`BenefitDescription`	Text that describes the promotion. Ancestry: Offers/Offer/Promotions/Promotion/Summary
`Bin`	Container for Bin elements. Children: BinItemCount, BinName, BinParameter
`Binding`	Typically but not always similar to the product category. Ancestry: ItemAttributes
`BinItemCount`	Number of items in a bin. Ancestry: SearchBinSets/SearchBinSet/Bin/BinItemCount
`BinName`	Name of the bin. Ancestry: SearchBinSets/SearchBinSet/Bin/BinName
`BinParameter`	Container for the BrowseNodeId and it's value. Ancestry: Bin
`Brand`	An item's brand. Ancestry: ItemAttributes, Large, Medium
`BrowseNodeId`	A positive integer that uniquely identifies a parent product category. Ancestry: BrowseNode/Ancestors/BrowseNode BrowseNode/Children/BrowseNode
`CartId`	A positive integer that uniquely identifies a cart. All operations on a cart must include this value, which is generated by CartCreate. Ancestry: Cart
`CartItem`	A parent element for many child elements, including CartItemId, Quantity, Title, ProductGroup, Price, and ItemTotal. Ancestry: Cart/CartItems
`CartItemId`	A positive integer that uniquely identifies an item in a cart or in the Saved For Later area. You must use this value to modify cart items or Saved For Later items. Other identifiers, such as ASINs, do not work. Ancestry: CartItem SavedForLaterItem
`CartItems`	A parent element for many child elements, including SubTotal, and CartItem. Ancestry: Cart

Response Element	Definition
`Category`	Specifies the kind of promotion. Valid values include ForEachQuantityXGetQuantityFreeX , BuyAmountXGetSimpleShippingFreeX, and BuyAmountXGetAmountOffX. For more information, see Promotion Types (p. 273). Ancestry: Offers/Offer/Promotions/Promotion/Summary
`CEROAgeRating`	The Computer Entertainment Rating Organization sets ratings, advertising guidelines, and online privacy principles for video and computer games in the United States and Canada. The value of CEROAgeRating specifies whether a video or software game (search indices Software and VideoGames) is suitable for everyone to play. Ancestry: ItemAttributes
`ClothingSize`	Size of clothes Ancestry: ItemAttributes
`Code`	Number that uniquely identifies an error. Ancestry: Errors/Error
`Collection`	Container for items that are part of a collection. For more information, see Collections (p. 44) Ancestry: None
`CollectionItem`	An item that is part of a collection of items. For more information, see Collections (p. 44) Ancestry: None
`CollectionParent`	The parent item has an ASIN but it cannot be purchase. It names the collection. CollectionItems are children of the Collection Parent and can be purchased. For more information, see Collections (p. 44) Ancestry: None
`Collections`	Container for one or more collection objects. For more information, see Collections (p. 44) Ancestry: None
`Color`	Color Ancestry: ItemAttributes
`Comment`	Comment typically about the purpose of the list. Ancestry: List
`ComponentType`	Specifies what the promotion applies to, for example, Shipping, ItemPrice, Subtotal. Ancestry: Benefits/Benefit
`Condition`	Specifies the condition of the item, such as new, used, collectible, or refurbished. Ancestry: SellerListing/Condition Offers/Offer/OfferAttributes/Condition

Response Element	Definition
`CorrectedQuery`	A parent element that contains the elements related to a corrected keyword. See Keywords. Ancestry: Items
`CouponCombinationType`	Specifies the kinds of promotional coupons that can be combined, for example, Unrestricted, Preferential, and Exclusive. Ancestry: None
`Creator`	Creator associated with the item. Ancestry: ItemAttributes
`CurrencyAmount`	Price in terms of the lowest currency denomination, for example, pennies. Ancestry: EligibilityRequirements/EligibilityRequirement/
`CurrencyCode`	An abbreviation that specifies the format of the price for the associated locale. Ancestry: Cart/SavedForLaterItems/SubTotal Cart/CartItems/SubTotal SavedForLaterItem/ItemTotal SavedForLaterItem/Price CartItem/Price Cart/SubTotal EligibilityRequirements/EligibilityRequirement/CurrencyAmount Benefits/Benefit/FixedAmount OfferSummary/LowestCollectiblePrice/CurrencyCode OfferSummary/LowestNewPrice/CurrencyCode OfferSummary/LowestRefurbishedPrice/CurrencyCode OfferSummary/LowestUsedPrice/CurrencyCode Offers/Offer/OfferListing/Price/CurrencyCode Offers/Offer/OfferListing/SalePrice/CurrencyCode Transaction/Totals/Total VariationSummary/HighestPrice VariationSummary/LowestSalePrice VariationSummary/HighestSalePrice VariationSummary/LowestPrice
`Date`	The date the review was created. Ancestry: CustomerReviews/Review
`DateAdded`	Date the item was added to the list. Ancestry: ListItem/DateAdded
`DateCreated`	Date the list was created, in the form yyyy-mm-dd. Ancestry: List/DateCreated
`Department`	Department Ancestry: ItemAttributes
`Details`	Container for all of the elements that describe a promotion. Ancestry:

Response Element	Definition
`Director`	Director Ancestry: ItemAttributes
`EAN`	European Article Number, which is a number that uniquely identifies an item. Ancestry: ItemAttributes
`EANList`	The container for one or more EANListElement (p. 312) attributes. Ancestry: ItemAttributes/EANList
`EANListElement`	A possible EAN for the ASIN. Ancestry: ItemAttributes/EANList
`Edition`	Edition Ancestry: ItemAttributes
`EditorialReviewIsLinkSuppressed`	Boolean value that specifies whether or not the link to the editorial review is displayed. Ancestry: EditorialReview
`EISBN`	Electronic ISBN number for digital books Ancestry: ItemAttributes
`EligibilityRequirement`	Container for one or more EligibilityRequirementType elements. Ancestry: EligibilityRequirements
`EligibilityRequirementDescription`	Specifies the conditions necessary to qualify for the promotion. Ancestry: Offers/Offer/Promotions/Promotion/Summary
`EligibilityRequirements`	Container for one or more EligibilityRequirement elements. Ancestry: None
`EligibilityRequirementType`	Specifies the eligibility requirements to qualify for the promotion, for example, MinPurchase, MinQuantity, or, ForEachQuantity. Ancestry: EligibilityRequirements/EligibilityRequirement
`EndDate`	Specifies a date when the item will stop being sold. Or, specifies the last day and ending time (GMT) of the promotion. Ancestry: SellerListing/EndDate, Offers/Offer/Promotions/Promotion/Summary
`EpisodeSequence`	Television shows are broadcast in sequence. Each show is called an episode. EpisodeSequence specifies the number of the show in the series of episodes.
`ESRBAgeRating`	The Entertainment Software Rating Board (ESRB) sets ratings, advertising guidelines, and online privacy principles for video and computer games in the United States and Canada. The value of ESRBAgeRating specifies whether a video or software game (search indices Software and VideoGames) is suitable for everyone to play. Ancestry: ItemAttributes

Response Element	Definition
`Feature`	An item's feature Ancestry: ItemAttributes
`Feedback`	Provides customer feedback about the seller. Ancestry:
Fitment	A fitment is a part that works in (fits into/onto) a car. Children: Bed, BodyStyle, Brakes, DriveType, Engine, Make, MfrBodyCode, Model, Notes, Position, SpringTypes, Steering, Transmission, Trim, Wheelbase, Year Ancestor: FitmentAttributes: FitmentAttribute
`FitmentAttribute`	Container for one or more Fitment elements. Child: Fitment Ancestry: FitmentAttributes
`FitmentAttributes`	Container for one or more FitmentAttribute elements. Child: FitmentAttribute
`FixedAmount`	Specifies the price in terms of the lowest currency denomination. Ancestry: Benefits/Benefit
`Format`	An item's format Ancestry: ItemAttributes
`FormattedPrice`	The price formatted as it should be displayed to the user. The formatting is specified by CurrencyCode and varies by country. Or, the discounted price formatted for display. Ancestry: EligibilityRequirements/EligibilityRequirement/CurrencyAmount Benefits/Benefit/FixedAmount OfferSummary/LowestUsedPrice/FormattedPrice OfferSummary/LowestCollectiblePrice/FormattedPrice OfferSummary/LowestNewPrice/FormattedPrice Transaction/TransactionItems/TransactionItem/TotalPrice
`Genre`	Specifies the genre, such as romance, of a digital item. Ancestry: ItemAttributes
`GroupClaimCode`	An alphanumeric token to use to claim the promotional benefit. Ancestry: None
`HardwarePlatform`	Hardware Ancestry: Variations/Item/ItemAttributes
`HazardousMaterialType`	Type of hazardous material that is present in the product. Ancestry: ItemAttributes

Response Element	Definition
`Height`	Height of an item, package, or image. Ancestry: ImageSets/ImageSet/LargeImage ImageSets/ImageSet/MediumImage ImageSets/ImageSet/SmallImage ImageSets/ImageSet/TinyImaget ImageSets/ImageSet/SwatchImage/Height
`HelpfulVotes`	The number of customers who felt that the review was helpful. Ancestry: CustomerReviews/Review/HelpfulVotes
`HMAC`	Hashing for Message Authentication (HMAC) is a keyed hash function that is used with a CryptographicHashFunction such as SHA-1 or MD5 that cannot be computed with out the key. This is used to identify and secure a cart. Each cart operation must include this value. Ancestry: Cart
`IFrameURL`	URL to an iframe that contains customer reviews. To embed the iframe on a web page, add the following to your HTML: `<iframe src="reviews_iframe_url" />` **Note** The URL expires in 24 hours. Ancestry: CustomerReviews
`Image`	An image associated with a seller's item. Ancestry: SellerListing/Image
`IsAdultProduct`	Indicates if the product is considered to be for adults only. Ancestry: ItemAttributes
`IsAutographed`	Specifies whether or not the item is autographed. Ancestry: ItemAttributes
`ISBN`	ISBN number Ancestry: ItemAttributes
`IsCategoryRoot`	Boolean value that specifies if the browse node is at the top of the browse node tree. Ancestry: BrowseNodes
`IsEligibleForSuperSaverShipping`	Specifies whether the item is eligible for super saving shipping. Ancestry: Offers/Offer/OfferListing
`IsEligibleForTradeIn`	Specifies whether or not the item is eligible for trade-in. Ancestry: ItemAttributes

Response Element	Definition
`IsEmailNotifyAvailable`	The value of IsEmailNotifyAvailable is a boolean where 1 means that an Email can be sent when the item is in stock, 0 when an Email cannot be sent. This functionality is not available for all offers.
`IsFit`	YES, NO, or MAYBE values specify whether ot not a part will work in a specified vehicle.
`IsInBenefitSet`	A boolean value. When True, the associated ASIN is what the customer receives as part of the promotion. Ancestry: ItemApplicability
`IsInEligibilityRequirementSet`	A boolean value. When True, the associated ASIN is what the customer must purchase to qualify for the promotion. Ancestry: ItemApplicability
`IsLinkSuppressed`	"1" (true) if there were any HTML links in editorial reviews that have been suppressed. Ancestry: EditorialReview
`IsMemorabilia`	Specifies whether the item is considered memorabilia. Ancestry: ItemAttributes
`IsNext`	Indicates that there are more vehicle parts to return with higher ASIN values than those already returned.
`IsPrevious`	Indicates that there are more vehicle parts to return with lower ASIN values than those already returned.
`ItemApplicability`	Container for ASIN, IsInBenefitSet, and IsInEligibilityRequirementSet. This group of values specifies whether the ASIN is what the customer receives as part of the promotion, whether the ASIN is what the customer needs to purchase to qualify for the promotion, or both. Ancestry: ItemApplicability/ItemApplicability
`ItemDimensions`	Container for Height, Length, Weight, and Width. Ancestry: ItemAttributes
`IssuesPerYear`	Number of issues per year, usually, in a subscription. Ancestry: ItemAttributes
`IsValid`	Boolean value that specifies whether the request syntax was correct. Ancestry: Request
`ItemAttributes`	Container for many attributes that describe an item. Children: 88 elements.
`ItemPartNumber`	The item part number. Ancestry: ItemAttributes
`Keywords`	The (corrected) words used in a search. Ancestry: Items/CorrectedQuery

Response Element	Definition
`Label`	Label Ancestry: ItemAttributes
`Language`	Container for the Name, Type, and AudioFormat response elements, which together describe a language. Ancestry: ItemAttributes/Languages
`Languages`	Container for one or more Language response elements. Ancestry: ItemAttributes
`LargeImage`	Container for a large image Ancestry: ImageSets/ImageSet
`LastModified`	Date when the list was last modified Ancestry: List/LastModified
`LegalDisclaimer`	Legal disclaimer Ancestry: ItemAttributes
`Length`	Length Ancestry: ItemAttributes/ItemDimensions ItemAttributes/PackageDimensions/Length ItemAttributes
`ListItemId`	A number that uniquely identifies an item on a list. Ancestry: ListItem/ListItemId
`ListPrice`	The manufacturer's suggested retail price for a product. Ancestry: ItemAttributes
LoyaltyPoints	In the JP locale only, loyalty points are returned. Loyalty points are used to generate sales. Ancestry: Offers/Offer
`LoyaltyPoints`	The number of points awarded as part of a purchase. Points translate into rewards. Ancestry: Offers, VariationOffers
`Manufacturer`	Manufacturer Ancestry: ItemAttributes
`ManufacturerMaximumAge`	Defines the maximum age in months the user should be to enjoy the use of the item. For example, for a toy targeted at kids from ages 2 to 4, 4 would be the value for the ManufacturerMaximumAge. Ancestry: ItemAttributes
`ManufacturerMinimumAge`	Defines the minimum age in months the user should be to enjoy the use of the item. For example, for a toy targeted at kids from ages 2 to 4, 2 would be the value for the ManufacturerMinimumAge. Ancestry: ItemAttributes
`ManufacturerPartsWarrantyDescription`	Describes the manufacturer's parts warranty Ancestry: ItemAttributes

Response Element	Definition
`MaterialType`	The type of material used in the item. Ancestry: ItemAttributes
`MaximumHours`	The maximum number of hours for which the item might be available. Ancestry: Offers/Offer/OfferListing/AvailabilityAttributes
`MediaType`	The type of media used by a recording device. Ancestry: ItemAttributes
`MediumImage`	Container for a medium image Ancestry: ImageSets/ImageSet
`MerchandisingMessage`	MerchandisingMessage contains text entered by a merchant that describes a promotion. Ancestry: None
`MerchantId`	A positive integer distributed by Amazon that uniquely identifies a merchant. Ancestry: CartItem SavedForLaterItem
`Message`	Message, or, a description of an error. Ancestry: Items/CorrectedQuery, Errors/Error
`MetalType`	The type of metal used in the item. Ancestry: ItemAttributes
`MinimumHours`	The minimum number of hours for which the item is available. Ancestry: Offers/Offer/OfferListing/AvailabilityAttributes
`Model`	Model Ancestry: ItemAttributes
`MoreOffersUrl`	The URL where all offers for an item are displayed. The URLs provided on the page are the exact ones that you should use when you link back to Amazon.com. They are tagged with your Associate tag and contain other tracking information to increase your hourly request limit as the sales that you generate increase. Ancestry: OfferFull, OfferListing, Offers
`MPN`	Manufacturer's part number. Ancestry: ItemAttributes
`Name`	Name of customer of item in a bin. Or, the name of a parameter passed in the request. Ancestry: BrowseNode/Ancestors/BrowseNode BrowseNode/Children/BrowseNode, Seller, SearchBinSets, SearchBinSet/Bin/BinParameter, Arguments/Argument
`Nickname`	Seller's nickname Ancestry: SellerListing/Seller

`Number`	An attribute in the Disc tag that denotes the CD number in the item, for example, <Disc Number="1">. Items with multiple CDs would have multiple values for Disc. Or, an attribute in the Track tag that denotes the track number on the CD, for example, <Track Number="9">Now Or Never</Track>. Ancestry: Tracks/Disc Tracks/Disc/Track
`NumberOfDiscs`	Number of discs an item can hold or use. Ancestry: ItemAttributes
`NumberOfIssues`	Number of issues in a subscription. Ancestry: ItemAttributes
`NumberOfItems`	Number of items. Ancestry: ItemAttributes
`NumberOfPages`	Number of pages. Ancestry: ItemAttributes
`NumberOfTracks`	Number of recorded track on a CD. Ancestry: ItemAttributes
`OccasionDate`	Specifies the date of an occasion, such as a wedding or birthday. Ancestry: List/OccasionDate
`OfferListingId`	A number that uniquely identifies an offer listing. This number represents a sales offer from a specific merchant. Ancestry: Offers/Offer/OfferListing
`OperatingSystem`	Specifies the name of the operating system on a computer. Ancestry: ItemAttributes, VariationMatrix
`OtherCategoriesSimilarProducts`	Parent element for Title and ASIN of similar products in other product groups Ancestry: Cart
`PackageQuantity`	Quantity of items in a package Ancestry: Variations/Item/ItemAttributes/
`ParentASIN`	A parent ASIN for an item in Saved For Later or the Active areas of a cart.. Ancestry: CartItem SavedForLaterItem
`PartBrandBins`	Container for one or more Bin elements.
`PartBrowseNodeBins`	Container for one or more Bin elements.
`PartNumber`	The part number. Ancestry: ItemAttributes
`PartnerName`	For wedding registries, this is the name of the bride or groom. Ancestry: List/PartnerName

`Platform`	Operating system. Ancestry: ItemAttributes
`Price`	A parent element for Amount, CurrencyCode, and FormattedPrice. In this case, the price is for an item in the Active or Saved For Later areas, respectively. Ancestry: CartItem SavedForLaterItem
`ProductGroup`	The product category an item belongs to. The name of a category, such as sporting goods, to which an item in the cart belongs. The name of a category, such as sporting goods, to which an item in Saved For Later belongs. Ancestry: ItemAttributes/ProductGroup CartItem/ProductGroup SavedForLaterItem
ProductTypeSubcategory	Subcategory of product type Ancestry: Variations/Item/ItemAttributes
`Promotion`	Container for one or more Details elements. Ancestry: Offers/OfferPromotions
`PromotionId`	An alphanumeric token that uniquely identifies a promotion. Ancestry: Offers/Offer/Promotions/Promotion/Summary
`Promotions`	A container for one or more Promotion elements. Ancestry: Offers/Offer
`PublicationDate`	Publication date. Ancestry: ItemAttributes
`Publisher`	Publisher Ancestry: ItemAttributes
`PurchaseURL`	A URL that the customer must use to purchase the items in their cart. Ancestry: Cart
`Quantity`	The number of a particular item in a cart. Or, the number of items in stock.Or, the number of a particular item in Saved For Later. Ancestry: CartItem, SellerListing SavedForLaterItem
`Rating`	A customer's rating in their review where a rating of 5 is the best. Ancestry: CustomerReviews/Review
`RegionCode`	A code that specifies a region. Ancestry: ItemAttributes
`RegistryName`	Name of a wedding or baby registry. Ancestry: List/RegistryName

`RelatedItem`	Container for an item that is related to the one specified in the ItemLookup request. Ancestry: RelatedItems Children: Item, ASIN , ItemAttributes
`RelatedItems`	A container for one or more items that are related to the one specified in an ItemLookup request. This value is only returned with the Relateditems response group. Children: Relationship, Relationshiptype, RelatedItemCount, RelatedItemPageCount, RelatedItemPage, RelatedItem.
`RelatedItemsCount`	Specifies the number of related items found. This number, however, can be larger than the actual number returned because not all related items found are available.
`RelatedItemPage`	Specifies which page of related items has been returned. Ancestry: RelatedItems
`RelatedItemPageCount`	Specifies the number of pages of related items found. There are up to ten items per page. Ancestry: RelatedItems
`Relationship`	Child or parent. Relationships are uni-directional: either parent to child, or child to parent. Ancestry: RelatedItems
`RelationshipType`	Specifies how the related item relates to the specified item in the ItemLookup request, for example, Tracks. For more information, go to the ItemLookup page. Ancestry: RelatedItems
`ReleaseDate`	Date on which the item was latest released. Items that have been released multiple times have both release dates and original release dates. See ReleaseDate. Ancestry: ItemAttributes
`RequestId`	A number that uniquely identifies a request Ancestry: OperationRequest
`Role`	Role Ancestry: ItemAttributes/Creator
`RunningTime`	The duration of a presentation. Ancestry: ItemAttributes
`SalesRank`	Indicates how well an item is selling within its product category. The lower the number, the better the item has sold. Ancestry: Item
`SavedForLaterItem`	A positive integer that uniquely identifies an item in Saved For Later. Ancestry: Cart/SavedForLaterItems Cart

`SearchBinSet`	A means by which to categorize results, such as price range. Ancestry: SearchBinSets
`SearchBinSets`	Parent element for SearchBins element. Ancestry: SearchBinSets
`SeikodoProductCode`	The Seikodo catalog number (JP only). Ancestry: ItemAttributes
`ShipmentItems`	The items in a specific shipment. Ancestry: Transaction/Shipments/Shipment
`Shipments`	The shipments Ancestry: Transaction
`SimilarProducts`	Parent element for Title and ASIN of similar products in the same product group Ancestry: Cart
`SimilarViewedProducts`	Parent element for Title and ASIN of similar products in the same product group that have been viewed Ancestry: Cart
`Size`	Size Ancestry: ItemAttributes
`SKU`	Stock Keeping Unit (SKU) is a number that uniquely identifies an item. Ancestry: ItemAttributes SellerListing
`SmallImage`	Container for a small image. Thumbnail and Small images are the same size. Ancestry: ImageSets/ImageSet
`Source`	Where the review was entered. Typically, this is Amazon.com. It is possible, however, for customers to enter reviews from other sites. Ancestry: EditorialReviews/EditorialReview
`StartDate`	The data an item goes on sale. Or, specifies the first day and beginning time (GMT) of the promotion. Ancestry: SellerListing, Offers/Offer/Promotions/Promotion/Summary
`StoreId`	A number that uniquely identifies the seller's store. Ancestry: SellerListing/Seller
`StoreName`	The name of the seller's store. Ancestry: SellerListing/Seller
`Studio`	The name of the studio, such as Warner Brothers, that produced a digital item. Ancestry: ItemAttributes

`SubscriptionLength`	The duration of a subscription. Ancestry: ItemAttributes
`Summary`	A short summary of the content of the review. Or, a container for all promotion information, including the elements BenefitDescription, Category, EndDate, EligibilityRequirementsDescription, PromotionId, StartDate, and TermsAndConditions. Ancestry: CustomerReviews/Review/Summary, Offers/Offer/Promotions/Promotion
`SwatchImage`	Container for a swatch image, which is smaller than a Small image. Ancestry: ImageSets/ImageSet
`TermsAndConditions`	Specifies the terms and conditions of the promotion. Ancestry: Offers/Offer/Promotions/Promotion/Summary
`ThumbnailImage`	Container for a Thumbnail image. Thumbnail and Small images are the same size. Ancestry: ImageSets/ImageSet
`TinyImage`	Container for a tiny image Ancestry: ImageSets/ImageSet
`Title`	Title or the name of the accessory, item, new release, similar products in other product groups, similar products in the same product group, similar products in the same product group that have been viewed Ancestry: ItemAttributes Accessories/Accessory SavedForLaterItem Cart/NewReleases/NewRelease/Title CartItem OtherCategoriesSimilarProduct SimilarProduct SimilarViewedProduct TopSellers/TopSeller
`TopItem`	Container object for information related to ranked responses, including MostGifted, MostWishedFor, TopSellers, and NewReleases. Information contained includes ASIN, Title, ProductGroup, Author, Artist, and Actor. Ancestry: MostGifted, MostWishedFor, TopSellers, NewReleases Children: ASIN, Title, ProductGroup, Actor, Artist, Author
`TopItemSet`	Container for one or more TopItem elements.
`TotalCollectible`	The total number of collectible items for sale. Ancestry: OfferSummary
`TotalItems`	The total number of items found on the list. Each page holds up to ten items. Ancestry: Items/TotalItems
`TotalNew`	The total number of new items for sale. Ancestry: OfferSummary

`TotalOfferPages`	Number of pages of offers. By default, the first ten offers are returned in the response. Ancestry: Offers
`TotalOffers`	Total number of offers. Ancestry: Offers
`TotalPages`	The total number of pages found in a response. Each page holds up to ten items. Type: nonNegativeInteger Ancestry: Items, Fitment
`TotalRatings`	Catalogs all customer ratings of the seller. Ancestry: Seller/TotalRatings
`TotalRefurbished`	The total number of refurbished items for sale. Ancestry: OfferSummary
`TotalResults`	Total number of items found by the request. Only ten are returned at a time. Ancestry: Items, List
`TotalReviewPages`	The number of pages of reviews found. Up to ten reviews are returned in a response Ancestry: CustomerReviews
`TotalReviews`	Total number of reviews of an item. Ancestry: CustomerReviews
`Totals`	Container object for all other Total* elements, including Total, SubTotal, Tax, ShippingCharges, Promotion, and so forth. Ancestry: Transaction
`TotalTimesRead`	The number of times a list has been viewed. Ancestry: List
`TotalUsed`	The total number of used items for sale. Ancestry: OfferSummary
`TotalVotes`	The total number of review votes cast. Ancestry: CustomerReviews/Review/TotalVotes
`Track`	Refers to each track on a CD. On a music CD, each track corresponds to a song. Ancestry: Tracks/Disc
`TradeInValue`	The trade-in value of this item. Ancestry: ItemAttributes
`TransactionDate`	The date on which the transaction began of the form yyyy-mm-ddThh.mm.ss, for example, 2011-09-24T19:19:27. Ancestry: Transaction

`TransactionDateEpoch`	The date on which the transaction began in the form of epoch seconds. Ancestry: Transaction
`TransactionId`	A series of dash-separated integers, such as 111-222-333, that uniquely identify a transaction. Ancestry: Transaction
`TransactionItem`	Container for all of the information related to a specified transaction. Ancestry: Transaction
`TransactionItemId`	A string that uniquely identifies a transaction item. Ancestry: Transaction/TransactionItems/TransactionItem
`TransactionItems`	Container for TransactionItem Ancestry: Transaction
`Type`	Type, depending on the context. If the ancestor is TopItemSet, the Type is the ranking criteria, such as MostGifted. Ancestry: ItemAttributes/Languages/Language, TopItemSet
`UPC`	Universal Product Code, which is a 12 digit number, 6 of which represents an item's manufacturer. These numbers are translated into a bar code that is printed on an item or its packaging. Ancestry: ItemAttributes SellerListing
`UPCList`	The container for one or more UPCListElement (p. 324) attributes. Ancestry: ItemAttributes
`UPCListElement`	A possible UPC for the ASIN. Ancestry: ItemAttributes/UPCList
`URL`	URL of an image. Ancestry: Variations/Item/ImageSets/ImageSet/LargeImage Variations/Item/ImageSets/ImageSet/SwatchImage Variations/Item/ImageSets/ImageSet/MediumImage Variations/Item/ImageSets/ImageSet/SmallImage
`URLEncodedHMAC`	A URL-encoded version of the HMAC that can be used directly in a request. Ancestry: Cart
`UserAgent`	The client application, for example, web browsers, search engine crawlers, mobile phones, screen readers and braille browsers. The text associated with UserAgent contains the application name, version, host operating system, and language. Ancestry: OperationRequest
`UserId`	An alphanumeric token that uniquely identifies the customer that tagged the entity. Ancestry:

VariationAttribute	Container for a variation name and value. Ancestry: Item/VariationAttributes/
VariationDimension	Container for dimensions Ancestry: Variations/VariationDimensions/
`Warranty`	Warranty terms Ancestry: ItemAttributes
`WEEETaxValue`	The Waste Electrical and Electronic Equipment tax for the item. Ancestry: ItemAttributes
`Weight`	Weight Ancestry: ItemAttributes/ItemDimensions
`Width`	Width Ancestry: ItemAttributes/ItemDimensions Variations/Item/ImageSets/ImageSet/SwatchImage Variations/Item/ImageSets/ImageSet/LargeImage Variations/Item/ImageSets/ImageSet/SmallImage Variations/Item/ImageSets/ImageSet/MediumImage
`Year`	Year, for example, 2008.

ItemSearch Sort Values By Locale

The `ItemSearch` *`Sort`* parameter enables you to choose the order of the items in a response. Available sort values vary by locale and search index.

There are many sort values. The majority are not applied unless the Sort parameter is included in the request. There are two sort values, however, that are used by default:

- For an ItemSearch requests that do not use the BrowseNode parameter, results are sorted by Relevance.
- For an ItemSearch requests that do use the BrowseNode parameter, results are sorted by BestSeller ranking.

To see the sort values that can be used, click on one of the following locales:

Sort Values for CA

Topics

SearchIndex: All

You cannot use any sort parameters with the All search index.

SearchIndex: Books

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
inverse-pricerank	Price: high to low
daterank	Pub Date: New to Old
titlerank	Alphabetical: A to Z

SearchIndex: Classical

Value	Description
salesrank	Bestselling
titlerank	Alphabetical: A to Z
orig-rel-date	Rel Date: New to Old

SearchIndex: DVD

Value	Description
salesrank	Bestselling
titlerank	Alphabetical: A to Z

SearchIndex: Electronics

Value	Description
salesrank	Bestselling
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: ForeignBooks

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
inverse-pricerank	Price: high to low
daterank	Publication date: newer to older
titlerank	Alphabetical: A to Z

SearchIndex: Kitchen

Value	Description
salesrank	Bestselling
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
price	Price: low to high
-price	Price: high to low
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Music

Value	Description
salesrank	Bestselling
titlerank	Alphabetical: A to Z
orig-rel-date	Rel Date: New to Old

SearchIndex: Software

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
inverse-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-daterank	Rel Date: Old to New

SearchIndex: SoftwareVideoGames

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
inverse-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-daterank	Rel Date: Old to New

SearchIndex: VHS

Value	Description
salesrank	Bestselling
-titlerank	Alphabetical: Z to A

SearchIndex: Video

Value	Description
salesrank	Bestselling
titlerank	Alphabetical: A to Z

Value	Description
-titlerank	Alphabetical: Z to A

SearchIndex: VideoGames

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
inverse-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

Sort Values for DE

Topics

SearchIndex: All

You cannot use any sort parameters with the All search index.

SearchIndex: Apparel

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low
salesrank	Bestselling to worst selling

SearchIndex: Automotive

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
reviewrank	Average customer review: high to low

SearchIndex: Baby

Value	Description
price	Price: low to high

Value	Description
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling to worst selling

SearchIndex: Beauty

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low
salesrank	Bestselling to worst selling

SearchIndex: Books

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
pricerank	Price: low to high
inverse-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Classical

Value	Description
salesrank	Bestselling
price	Price: low to high

Value	Description
-price	Price: high to low
pubdate	Publication date: most recent to oldest
-pubdate	Publication date: oldest to most recent
publication_date	Publication date: most recent to oldest
-publication_date	Publication date: oldest to most recent
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: DVD

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Electronics

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: ForeignBooks

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
pricerank	Price: low to high
inverse-pricerank	Price: high to low
titlerank	Alphabetical: A to Z

Value	Description
-titlerank	Alphabetical: Z to A

SearchIndex: Grocery

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
reviewrank	Average customer review: high to low

SearchIndex: HealthPersonalCare

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: HomeGarden

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: HomeImprovement

Value	Description
price	Price: low to high
-price	Price: high to low
reviewrank	Average customer review: high to low
salesrank	Bestselling

SearchIndex: Jewelry

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: KindleStore

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
-edition-sales-velocity	Quickest to slowest selling products.
price	Price: low to high
-price	Price: high to low
reviewrank	Average customer review: high to low
daterank	Publication date: newer to older
salesrank	Bestselling

SearchIndex: Kitchen

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling to worst selling

SearchIndex: Magazines

Value	Description
salesrank	Bestselling
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Jewelry

Value	Description
salesrank	Bestselling
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
price	Price: low to high
-price	Price: high to low
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: MP3Downloads

Value	Description
salesrank	Bestselling

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.
price	Price: low to high
-price	Price: high to low
artistalbumrank	Artist: A to Z
-artistalbumrank	Artist: Z o A
albumrank	Album: A to Z
-albumrank	Album: Z to A
runtime	Time: Short to Long
-runtime	Time: Long to Short
-releasedate	Release date: newer to older
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Music

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
pubdate	Publication date: most recent to oldest
-pubdate	Publication date: oldest to most recent
publicationdate	Publication date: most recent to oldest
-publicationdate	Publication date: oldest to most recent
releasedate	Release date: most recent to oldest
-releasedate	Release date: oldest to most recent
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: MusicalInstruments

Value	Description
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.

SearchIndex: MusicTracks

Value	Description
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: OfficeProducts

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling

SearchIndex: OutdoorLiving

Value	Description
salesrank	Bestselling
price	Price: low to high

Value	Description
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Outlet

You cannot use any sort parameters with the Outlet search index.

SearchIndex: PCHardware

Value	Description
psrank	The sales rank taking into consideration projected sales.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
reviewrank	Average customer review: high to low

SearchIndex: Photo

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Shoes

Value	Description
-launch-date	Newest arrivals
price	Price: low to high
-price	Price: high to low

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling first
reviewrank	Average customer review: high to low

SearchIndex: Software

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
-date	Publication date: newer to older
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: SportingGoods

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
release-date	Release date: older to newer
-release-date	Release date: newer to older

SearchIndex: SoftwareVideoGames

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high

Value	Description
inverse-pricerank	Price: high to low
-date	Publication date: newer to older
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Tools

Value	Description
salesrank	Bestselling
price	Price: low to high
-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Toys

Value	Description
pmrank	Featured items
salesrank	Bestselling first
price	Price: low to high
-price	Price: high to low
-date	Pub Date: Old to New
titlerank	Alphabetical: A to Z

SearchIndex: VHS

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Video

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: VideoGames

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
-date	Publication date: newer to older
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Watches

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

Sort Values for FR

Topics

SearchIndex: All

You cannot use any sort parameters with the All search index.

SearchIndex: Apparel

Value	Description
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.

SearchIndex: Baby

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling

SearchIndex: Beauty

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling

SearchIndex: Books

Value	Description
salesrank	Bestselling to worse selling
pricerank	Price: low to high
inverse-pricerank	Price: high to low
-daterank	Pub Date: Old to New
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Classical

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
price	Price: low to high
inverse-pricerank	Price: high to low
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: A to Z

SearchIndex: DVD

Value	Description
amzrank	Alphabetical: A to Z
availability	Most to least available
salesrank	Bestselling
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Electronics

Value	Description
salesrank	Bestselling
price	Price: Low to high
-price	Price: High to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: ForeignBooks

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
inverse-pricerank	Price: high to low

Value	Description
-daterank	Publication date: Old to New
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: HealthPersonalCare

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling

SearchIndex: Jewelry

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Kitchen

Value	Description
price	Price: low to high
-price	Price: high to low

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling to worst selling

SearchIndex: Jewelry

Value	Description
salesrank	Bestselling
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
price	Price: low to high
-price	Price: high to low
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: MP3Downloads

Value	Description
albumrank	Album: A to Z
-albumrank	Album: Z to A
artistalbumrank	Artist: A to Z
-artistalbumrank	Artist: Z o A
price	Price: low to high
-price	Price: high to low
runtime	Time: Short to Long
-runtime	Time: Long to Short
-releasedate	Release date: newer to older

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Music

Value	Description
releasedate	Release date: most recent to oldest
-releasedate	Release date: oldest to most recent
salesrank	Bestselling
pricerank	Price: low to high
inverse-pricerank	Price: high to low
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: MusicalInstruments

Value	Description
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.

SearchIndex: MusicTracks

Value	Description
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: OfficeProducts

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling

SearchIndex: Outlet

You cannot use any sort parameters with the Outlet search index.

SearchIndex: Shoes

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: PCHardware

Value	Description
psrank	
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z

SearchIndex: Software

Value	Description
salesrank	Bestselling
price	Price: low to high
-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: A to Z

SearchIndex: SoftwareVideoGames

Value	Description
salesrank	Bestselling
price	Price: low to high
-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: A to Z
-date	Rel Date: Old to New

SearchIndex: SportingGoods

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
reviewrank	Highest to lowest ratings in customer reviews.
-launch-date	Newest arrivals

SearchIndex: VHS

Value	Description
amzrank	Most to least available
availability	Most to least available
salesrank	Bestselling
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Video

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: VideoGames

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
date	Rel Date: new to old
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Watches

Value	Description
price	Price: low to high
-price	Price: high to low
salesrank	Bestselling to worst selling
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.

Sort Values for IT

Topics

SearchIndex: All

You cannot use any sort parameters with the All search index.

SearchIndex: Books

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Best to worse selling
price	Price: Low to high
-price	Price: High to low
reviewrank	Average customer review: high to low

Value	Description
-pubdate	Publish date: oldest to recent

SearchIndex: DVD

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Best to worse selling
price	Price: Low to high
-price	Price: High to low
reviewrank	Average customer review: high to low
-releasedate	Release date: oldest to recent

SearchIndex: Electronics

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Best to worse selling
price	Price: Low to high
-price	Price: High to low
reviewrank	Average customer review: high to low
reviewrank	Average customer review: high to low

SearchIndex: ForeignBooks

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Best to worse selling
price	Price: Low to high
-price	Price: High to low
reviewrank	Average customer review: high to low
-pubdate	Publish date: oldest to recent

SearchIndex: Music

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Best to worse selling
price	Price: Low to high
-price	Price: High to low
reviewrank	Average customer review: high to low
reviewrank	Average customer review: high to low
-releasedate	Average customer review: high to low

SearchIndex: Software

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Best to worse selling

Value	Description
price	Price: Low to high
-price	Price: High to low
reviewrank	Average customer review: high to low
reviewrank	Average customer review: high to low
-releasedate	Average customer review: high to low

SearchIndex: VideoGames

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Best to worse selling
price	Price: Low to high
-price	Price: High to low
reviewrank	Average customer review: high to low
reviewrank	Average customer review: high to low
-releasedate	Average customer review: high to low

Sort Values for JP

Topics

SearchIndex: All

You cannot use any sort parameters with the All search index.

SearchIndex: Apparel

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling to worst selling

SearchIndex: Automotive

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low

Value	Description
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Baby

Value	Description
psrank	Bestseller ranking taking into consideration projected sales.The lower the value, the better the sales.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z

SearchIndex: Beauty

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Books

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
inverse-pricerank	Price: high to low
daterank	Publication date: newer to older
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Classical

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
-pricerank	Price: high to low
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-orig-rel-date	Release date: newer to older
orig-rel-date	Release date: older to newer
releasedate	Release date: newer to older
-releasedate	Release date: older to newer

SearchIndex: DVD

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
-pricerank	Price: high to low
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-orig-rel-date	Release date: newer to older
orig-rel-date	Release date: older to newer
releasedate	Release date: older to newer
-releasedate	Release date: newer to older

SearchIndex: Electronics

Value	Description
salesrank	Bestselling
price	Price: low to high

Value	Description
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-releasedate	Release date: newer to older
releasedate	Release date: older to newer

SearchIndex: ForeignBooks

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
inverse-pricerank	Price: high to low
daterank	Publication date: newer to older
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Grocery

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
reviewrank	Average customer review: high to low

SearchIndex: HealthPersonalCare

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Hobbies

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
release-date	Release date: older to newer
-release-date	Release date: newer to older
mfg-age-min	Minimum age of user sorted from youngest to oldest.
-mfg-age-min	Minimum age of user sorted from oldest to youngest.

SearchIndex: HomeImprovement

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Jewelry

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.

Value	Description
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Kitchen

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-release-date	Release date: newer to older
release-date	Release date: older to newer
releasedate	Items are sorted such that the most recently released item is listed first. Same as release-date.
-releasedate	Items are sorted such that the last released item is listed first. Same as -release-date.

SearchIndex: MP3Downloads

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
artistalbumrank	Artist album rank: high to low
-artistalbumrank	Artist album rank: low to high
albumrank	Album rank: high to low
-albumrank	Album rank: low to high
runtime	Track length: high to low
-runtime	Track length: low to high
price	Price: low to high

Value	Description
-price	Price: high to low
price-new-bin	Price: low to high
-price-new-bin	Price: high to low
reviewrank_authority	Review rank: high to low
-releank	Review rank: low to high
releasedate	Release date: older to newer

SearchIndex: Music

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
-pricerank	Price: high to low
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-orig-rel-date	Release date: newer to older
orig-rel-date	Release date: older to newer
releasedate	Release date: older to newer
-releasedate	Release date: newer to older

SearchIndex: MusicalInstruments

Value	Description
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.

SearchIndex: MusicTracks

Value	Description
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: OfficeProducts

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Shoes

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Software

Value	Description
salesrank	Bestselling
price	Price: low to high

Value	Description
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-release-date	Release date: newer to older
release-date	Release date: older to newer
releasedate	Items are sorted such that the most recently released item is listed first. Same as release-date.
-releasedate	Items are sorted such that the last released item is listed first. Same as -release-date.

SearchIndex: SportingGoods

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
releasedate	Release date: older to newer
-releasedate	Release date: newer to older

SearchIndex: Toys

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-release-date	Release date: newer to older
release-date	Release date: older to newer
releasedate	Items are sorted such that the most recently released item is listed first. Same as release-date.
-releasedate	Items are sorted such that the last released item is listed first. Same as -release-date.

SearchIndex: VHS

Value	Description
salesrank	Bestselling
pricerank	Price: low to high
-pricerank	Price: high to low
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-orig-rel-date	Release date: newer to older
orig-rel-date	Release date: older to newer
releasedate	Release date: older to newer
-releasedate	Release date: newer to older

SearchIndex: Video

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
pricerank	Price: low to high
-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-orig-rel-date	Release date: newer to older
orig-rel-date	Release date: older to newer
releasedate	Release date: older to newer
-releasedate	Release date: newer to older

SearchIndex: VideoGames

Value	Description
salesrank	Bestselling
price	Price: low to high

Value	Description
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
-release-date	Release date: newer to older
release-date	Release date: older to newer
releasedate	Items are sorted such that the most recently released item is listed first. Same as release-date.
-releasedate	Items are sorted such that the last released item is listed first. Same as -release-date.

SearchIndex: Watches

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

Sort Values for UK

Topics

SearchIndex: All

You cannot use any sort parameters with the All search index.

SearchIndex: Apparel

Value	Description
-launch-date	The date the apparel was introduced from the most recent to the least recent
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low
salesrank	Bestselling to worst selling

SearchIndex: Automotive

Value	Description
price	Price: low to high
-price	Price: high to low

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low
salesrank	Bestselling to worst selling

SearchIndex: Baby

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
salesrank	Bestselling to worst selling

SearchIndex: Beauty

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low
salesrank	Bestselling to worst selling

SearchIndex: Books

Value	Description
salesrank	Bestselling

Value	Description
reviewrank	Average customer review: high to low
pricerank	Price: low to high
inverse-pricerank	Price: high to low
daterank	Publication date: older to newer
pubdate	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Classical

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
inverse-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: DVD

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
inverse-pricerank	Price: high to low
daterank	Publication date: older to newer
releasedate	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Electronics

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
inverse-pricerank	Price: high to low
daterank	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Grocery

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
reviewrank	Average customer review: high to low

SearchIndex: HealthPersonalCare

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
daterank	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: HomeGarden

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
daterank	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: HomeImprovement

Value	Description
price	Price: low to high
-price	Price: high to low
reviewrank	Average customer review: high to low
salesrank	Bestselling

SearchIndex: Jewelry

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling

SearchIndex: KindleStore

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
price	Price: low to high
-price	Price: high to low
reviewrank	Average customer review: high to low
daterank	Publication date: newer to older
salesrank	Bestselling

SearchIndex: Kitchen

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
daterank	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Jewelry

Value	Description
salesrank	Bestselling
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
price	Price: low to high
-price	Price: high to low
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Outlet

You cannot use any sort parameters with the Outlet search index.

SearchIndex: MP3Downloads

Value	Description
price	Price: low to high
-price	Price: high to low
-releasedate	Release date: most recent to oldest
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low
salesrank	Bestselling

SearchIndex: Music

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
releasedate	Release date: most recent to oldest
-releasedate	Release date: oldest to most recent
price	Price: low to high
-price	Price: high to low
inverse-pricerank	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: MusicalInstruments

Value	Description
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling
price	Price: low to high

Value	Description
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.

SearchIndex: MusicTracks

Value	Description
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: OfficeProducts

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.
salesrank	Bestselling

SearchIndex: OutdoorLiving

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
daterank	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: PCHardware

Value	Description
psrank	
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z

SearchIndex: Shoes

Value	Description
-launch-date	The date the apparel was introduced from the most recent to the least recent
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low
salesrank	Bestselling to worst selling

SearchIndex: Software

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
inverse-pricerank	Price: high to low
daterank	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: SoftwareVideoGames

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
inverse-pricerank	Price: high to low
daterank	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Toys

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
-mfg-age-min	Age: high to low
mfg-age-min	Age: low to high

SearchIndex: VHS

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
inverse-pricerank	Price: high to low
daterank	Publication date: older to newer
releasedate	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Video

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
inverse-pricerank	Price: high to low
daterank	Publication date: older to newer
releasedate	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: VideoGames

Value	Description
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
inverse-pricerank	Price: high to low
daterank	Publication date: older to newer
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Watches

Value	Description
-launch-date	The date the watch was introduced from the most recent to the least recent
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

Sort Values for US

Topics

SearchIndex: All

You cannot use any sort parameters with the All search index.

SearchIndex: Apparel

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
pricerank	Price: low to high
inverseprice	Price: high to low
-launch-date	Newest arrivals
sale-flag	On sale

SearchIndex: Appliances

Value	Description
salesrank	Bestselling
pmrank	Featured items
price	Price: low to high
-price	Price: high to low

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low

SearchIndex: ArtsAndCrafts

Value	Description
pmrank	Featured items
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: Automotive

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Baby

Value	Description
psrank	Bestseller ranking taking into consideration projected sales.The lower the value, the better the sales.
salesrank	Bestselling
price	Price: low to high

Value	Description
-price	Price: high to low
titlerank	Alphabetical: A to Z

SearchIndex: Beauty

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
-launch-date	Newest arrivals
sale-flag	On sale

SearchIndex: Books

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
reviewrank	Average customer review: high to low
pricerank	Price: low to high
inverse-pricerank	Price: high to low
daterank	Publication date: newer to older
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Classical

Value	Description
psrank	Bestseller ranking taking into consideration projected sales.The lower the value, the better the sales.
salesrank	Bestselling

Value	Description
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
orig-rel-date	Release date: newer to older
-orig-rel-date	Release date: older to newer
releasedate	Release date: newer to older
-releasedate	Release date: older to newer

SearchIndex: DigitalMusic

Value	Description
songtitlerank	Most popular
uploaddaterank	Date added

SearchIndex: DVD

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-video-release-date	Release date: newer to older
releasedate	Release date: newer to older

SearchIndex: Electronics

Value	Description
pmrank	Featured items
salesrank	Bestselling

Value	Description
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z

SearchIndex: Grocery

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
pricerank	Price: low to high
inverseprice	Price: high to low
launch-date	Newest launches first
sale-flag	On sale

SearchIndex: HealthPersonalCare

Value	Description
pmrank	Featured items
salesrank	Bestselling
pricerank	Price: low to high
inverseprice	Price: high to low
launch-date	Newest arrivals
sale-flag	On sale

SearchIndex: HomeImprovement

Value	Description
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low

Value	Description
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex:Industrial

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Jewelry

Value	Description
pmrank	Featured items
salesrank	Bestselling
pricerank	Price: low to high
inverseprice	Price: high to low
launch-date	Newest arrivals

SearchIndex: KindleStore

Value	Description
daterank	Publication date: newer to older
-edition-sales-velocity	Quickest to slowest selling products.
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low

SearchIndex: Kitchen

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Magazines

Value	Description
subslot-salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
daterank	Publication date: newer to older
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Merchants

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
pricerank	Price: low to high
inverseprice	Price: high to low
-launch-date	Newest arrivals
sale-flag	On sale

SearchIndex: Miscellaneous

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: MobileApps

Value	Description
pmrank	Featured items
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears, for example, the ranking is higher when keywords are found in titles, and, if there are multiple keywords, how closely they occur in descriptions, and, finally, how often customers purchased the products they found using the keyword.
reviewrank	Highest to lowest ratings in customer reviews.

SearchIndex: MP3Downloads

Value	Description
price	Price: low to high
-price	Price: high to low
-releasedate	Release date: most recent to oldest
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling

SearchIndex: Music

Value	Description
psrank	Bestseller ranking taking into consideration projected sales.The lower the value, the better the sales.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A
artistrank	Artist name: A to Z
orig-rel-date	Original release date of the item listed from newer to older. See release-date, which sorts by the latest release date.
release-date	Sorts by the latest release date from newer to older. See orig-rel-date, which sorts by the original release date.
releasedate	Release date: most recent to oldest
-releasedate	Release date: oldest to most recent
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.

SearchIndex: MusicalInstruments

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
-launch-date	Newest arrivals
sale-flag	On sale

SearchIndex: MusicTracks

Value	Description
titlerank	Alphabetical: A to Z

Value	Description
-titlerank	Alphabetical: Z to A

SearchIndex: OfficeProducts

Value	Description
pmrank	Featured items
salesrank	Bestselling
reviewrank	Average customer review: high to low
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z

SearchIndex: OutdoorLiving

Value	Description
psrank	Bestseller ranking taking into consideration projected sales.The lower the value, the better the sales.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: PCHardware

Value	Description
psrank	Bestseller ranking taking into consideration projected sales.The lower the value, the better the sales.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z

SearchIndex: PetSupplies

Value	Description
+pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Photo

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Shoes

Value	Description
-launch-date	Newest arrivals
pmrank	Featured items
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
reviewrank	Average customer review: high to low

SearchIndex: Software

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z

SearchIndex: SportingGoods

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
pricerank	Price: low to high
inverseprice	Price: high to low
launch-date	Newest arrivals
sale-flag	On sale

SearchIndex: Tools

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: Toys

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-age-min	Age: high to low

SearchIndex: UnboxVideo

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-video-release-date	Release date: newer to older

SearchIndex: VHS

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-video-release-date	Release date: newer to older

Value	Description
-releasedate	Release date: newer to older

SearchIndex: Video

Value	Description
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z
-video-release-date	Release date: newer to older
-releasedate	Release date: newer to older

SearchIndex: VideoGames

Value	Description
pmrank	Featured items
salesrank	Bestselling
price	Price: low to high
-price	Price: high to low
titlerank	Alphabetical: A to Z

SearchIndex: Watches

Value	Description
price	Price: low to high
-price	Price: high to low
relevancerank	Items ranked according to the following criteria: how often the keyword appears in the description, where the keyword appears (for example, the ranking is higher when keywords are found in titles), how closely they occur in descriptions (if there are multiple keywords), and how often customers purchased the products they found using the keyword.

Value	Description
reviewrank	Average customer review: high to low
salesrank	Bestselling to worst selling

SearchIndex: Wireless

Value	Description
daterank	Publication date: newer to older
pricerank	Price: low to high
inverse-pricerank	Price: high to low
reviewrank	Average customer review: high to low
salesrank	Bestselling
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

SearchIndex: WirelessAccessories

Value	Description
psrank	Bestseller ranking taking into consideration projected sales.The lower the value, the better the sales.
salesrank	Bestselling
titlerank	Alphabetical: A to Z
-titlerank	Alphabetical: Z to A

Search Index and ItemSearch Parameter Combinations

For a given search index, only some `ItemSearch` parameters are valid. Also, each locale supports only a subset of all search index values. The following sections explain, by locale and by search index, which `ItemSearch` parameters can be used in a request. For example, in the US locale, when SearchIndex is "Blended," the only parameter that can be used in an ItemSearch request is Keywords.

SearchIndex-ItemSearch Parameter Combinations for CA

This Search Index Matrix shows you which `ItemSearch` parameters may be used with each of the available *`SearchIndex`* values for the CA locale.

Topics

SearchIndex: All

In `ItemSearch` request, when *`SearchIndex`* equals "All", only the following parameters can be used in a request.

- Keywords

SearchIndex: Blended

- Keywords

SearchIndex: Books

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VALUE", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • Count	• ItemPage • Keywords • MaximumPrice • MinimumPrice	• Power • Publisher • Sort • Title

SearchIndex: Classical

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Classical", only the following parameters can be used in the request.

• Author • BrowseNode • Composer • Condition • Conductor	• Count • ItemPage • Keywords • MaximumPrice	• MinimumPrice • MusicLabel • Sort • Title

SearchIndex: DVD

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "DVD", only the following parameters can be used in the request.

• Actor • AudienceRating • BrowseNode • Condition • Count	• Director • ItemPage • Keywords • MaximumPrice	• MinimumPrice • Publisher • Sort • Title

SearchIndex: Electronics

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals Electronics only the following parameters can be used in the request.

• BrowseNode • Condition • Count • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • ReleaseDate • Sort • Title

SearchIndex: ForeignBooks

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "ForeignBooks", only the following parameters can be used in the request.

• BrowseNode • Condition • Count • ItemPage	• Keywords • MaximumPrice • MinimumPrice	• Power • Publisher • Sort • Title

SearchIndex: Kitchen

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Kitchen", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • Keywords	• Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Music

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Music", only the following parameters can be used in the request.

• Artist • BrowseNode • Condition • Count	• ItemPage • Keywords • MaximumPrice • MinimumPrice	• MusicLabel • Power • Sort • Title

SearchIndex: Software

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Software", only the following parameters can be used in the request.

• BrowseNode • Condition • Count	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: SoftwareVideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "SoftwareVideoGames", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • Count	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: VHS

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VHS", only the following parameters can be used in the request.

• Actor • BrowseNode • Condition • Count	• Director • ItemPage • Keywords • MaximumPrice	• MinimumPrice • Publisher • Sort • Title

SearchIndex: Video

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Video", only the following parameters can be used in the request.

• Actor • AudienceRating • BrowseNode • Condition • Count	• Director • ItemPage • Keywords • MaximumPrice	• MinimumPrice • Publisher • Sort • Title

SearchIndex: VideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VideoGames", only the following parameters can be used in the request.

• Author • Brand • BrowseNode • Condition • Count	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex-ItemSearch Parameter Combinations for DE

This Search Index Matrix shows you which `ItemSearch` parameters may be used with each of the available *`SearchIndex`* values for the DE locale.

Topics

SearchIndex: All

In `ItemSearch` request, when *`SearchIndex`* equals "All", only the following parameters can be used in a request.

- Keywords

SearchIndex: Apparel

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "apparel", only the following parameters can be used in the request.

• Author • Availability • Brand • BrowseNode • Condition • Count • DeliveryMethod	• Format • ISPUPostalCode • ItemPage • Keywords • Magazines • Manufacturer • MerchantId	• Performer • PostalCode • ReviewRank • Sort • State • Title

SearchIndex: Automotive

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Automotive", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • State • TextStream • Title

SearchIndex: Baby

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Baby", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MerchantId • MinimumPrice • Sort • State • Title

SearchIndex: Blended

- Keywords

SearchIndex: Beauty

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Beauty", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Count	• ItemPage • Keywords • MaximumPrice • MinimumPrice	• Manufacturer • MerchantId • Sort • Title

SearchIndex: Books

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Books", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • ItemPage	• Keywords • MaximumPrice • MerchantId • MinimumPrice	• Power • Publisher • Sort • Title

SearchIndex: Classical

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Music", only the following parameters can be used in the request.

• Artist • Availability • BrowseNode • Composer • Conductor • Count	• Format • ItemPage • Keywords • Magazines • MusicLabel • Orchestra	• Performer • PostalCode • Sort • State • Title

SearchIndex: DVD

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "DVD", only the following parameters can be used in the request.

• Actor • Availability • BrowseNode • Composer • Conductor • Count	• Format • ItemPage • Keywords • Magazines • MusicLabel • Orchestra	• Performer • PostalCode • Sort • State • Title

SearchIndex: Electronics

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Electronics", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: ForeignBooks

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "ForeignBooks", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • ItemPage	• Keywords • MaximumPrice • MerchantId • MinimumPrice	• Publisher • Sort • Title

SearchIndex: Grocery

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Grocery", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• MaximumPrice • MerchantId • MinimumPrice	• Keywords • Sort • Title

SearchIndex: HealthPersonalCare

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "HealthPersonalCare", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: HomeGarden

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "HomeGarden", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: Jewelry

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Jewelry", only the following parameters can be used in the request.

• Availability • BrowseNode • DeliveryMethod	• Keywords • MerchantId	• Sort • Title

SearchIndex: KindleStore

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "kindlestore", only the following parameters can be used in the request.

• Author • Availability • BrowseNode • Keywords	• MaximumPrice • MerchantID • MinimumPrice	• Publisher • Sort • Title

SearchIndex: Kitchen

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Kitchen", only the following parameters can be used in the request.

• Author • Availability • Brand • BrowseNode • Condition • Count • DeliveryMethod	• Format • ISPUPostalCode • ItemPage • Keywords • Magazines • Manufacturer • MerchantId	• Performer • PostalCode • Sort • State • Title

SearchIndex: Lighting

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Lighting", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • DeliveryMethod	• Keywords • MaximumPrice • MerchantId • MinimumPrice	• Sort • Title

SearchIndex: Magazines

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Magazines", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• MaximumPrice • MerchantId • MinimumPrice	• Publisher • Sort • Title

SearchIndex: MP3Downloads

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MP3Downloads", only the following parameters can be used in the request.

• Availability • BrowseNode • DeliveryMethod	• Keywords • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: Music

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Music", only the following parameters can be used in the request.

• Artist • Availability • BrowseNode • Count • Format	• ItemPage • Keywords • Magazines • MusicLabel • Performer	• PostalCode • Sort • State • Title

SearchIndex: MusicalInstruments

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicalInstruments", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• MaximumPrice • MerchantId • MinimumPrice	• Keywords • Sort • Title

SearchIndex: MusicTracks

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicTracks", only the following parameters can be used in the request.

• Condition • ItemPage • Keywords	• MaximumPrice • MerchantId	• MinimumPrice • Sort

SearchIndex: OfficeProducts

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "OfficeProducts", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • DeliveryMethod	• Keywords • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: OutdoorLiving

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "OutdoorLiving", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: Outlet

In `ItemSearch` request, when *`SearchIndex`* equals "Outlet", only the following parameters can be used in a request.

- Keywords

SearchIndex: PCHardware

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "PCHardware", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: Photo

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Photo", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: Software

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Software", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: SoftwareVideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "SoftwareVideoGames", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: SportingGoods

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "SportingGoods", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: Tools

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Tools", only the following parameters can be used in the request.

• Condition • ItemPage • Keywords	• MaximumPrice • MerchantId • MinimumPrice	• Sort • Title

SearchIndex: Toys

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Toys", only the following parameters can be used in the request.

• BrowseNode • Condition • Count	• ItemPage • Keywords	• Sort • MerchantId • Title

SearchIndex: VHS

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VHS", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• PostalCode • Publisher • Sort • State • Title

SearchIndex: Video

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Music", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• PostalCode • Publisher • Sort • State • Title

SearchIndex: VideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Music", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice • MerchantId • MinimumPrice	• MPAARating • Sort • Title

SearchIndex: Watches

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Watches", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• MerchantId • MinimumPrice • Sort	• State • TextStream • Title

SearchIndex-ItemSearch Parameter Combinations for FR

This Search Index Matrix shows you which `ItemSearch` parameters may be used with each of the available *`SearchIndex`* values for the FR locale.

Topics

SearchIndex: All

In `ItemSearch` request, when *`SearchIndex`* equals "All", only the following parameters can be used in a request.

- Keywords

SearchIndex: Apparel

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Apparel", only the following parameters can be used in the request.

• Availability • Author • Brand • BrowseNode	• DeliveryMethod • Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Baby

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Baby", only the following parameters can be used in the request.

• Availability • Author • Brand • BrowseNode	• DeliveryMethod • Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Beauty

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals " Beauty", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• DeliveryMethod • Keywords • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Blended

- Keywords

SearchIndex: Books

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Books", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • Count	• ItemPage • Keywords • MaximumPrice • MinimumPrice	• Power • Publisher • Sort • Title

SearchIndex: Classical

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Classical", only the following parameters can be used in the request.

• Actor • Availability • BrowseNode • Composer • Conductor • Count	• Format • ItemPage • Keywords • Magazines • MusicLabel • Orchestra	• Performer • Sort • State • Title

SearchIndex: DVD

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "DVD", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• Publisher • Sort • State • Title

SearchIndex: Electronics

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Electronics", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: ForeignBooks

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "ForeignBooks", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • Count	• ItemPage • Keywords • MaximumPrice • MinimumPrice	• Power • Publisher • Sort • Title

SearchIndex: HealthPersonalCare

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "HealthPersonalCare", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • DeliveryMethod	• KeyWords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Jewelry

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Jewelry", only the following parameters can be used in the request.

• Availability • BrowseNode • DeliveryMethod	• Keywords • Sort	• Title

SearchIndex: Kitchen

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Kitchen", only the following parameters can be used in the request.

• Author • Availability • Brand • BrowseNode • Condition • Count • DeliveryMethod	• Format • ItemPage • Keywords • Magazines • Manufacturer	• Performer • Sort • State • Title

SearchIndex: Lighting

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Lighting", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • DeliveryMethod	• Keywords • MaximumPrice • MinimumPrice	• Sort • Title

SearchIndex: MP3Downloads

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MP3Downloads", only the following parameters can be used in the request.

• Availability • BrowseNode • DeliveryMethod	• Keywords • MaximumPrice • MinimumPrice	• Sort • Title

SearchIndex: Music

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Music", only the following parameters can be used in the request.

• Artist • Availability • BrowseNode • Count • Format	• ItemPage • Keywords • Magazines • MusicLabel • Performer	• Sort • State • Title

SearchIndex: MusicalInstruments

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicalInstruments", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• MaximumPrice • MinimumPrice • Keywords	• Sort • Title

SearchIndex: MusicTracks

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicTracks", only the following parameters can be used in the request.

• BrowseNode • Condition • Count	• ItemPage • Keywords	• MinimumPrice • Sort

SearchIndex: OfficeProducts

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "OfficeProducts", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• DeliveryMethod • Keywords • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Outlet

In `ItemSearch` request, when *`SearchIndex`* equals "Outlet", only the following parameters can be used in a request.

- Keywords

SearchIndex: Shoes

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Shoes", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • Keywords	• Manufacturer • MaximumPrice • MinimumPrice	• Sort • Title

SearchIndex: Software

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Software", only the following parameters can be used in the request.

• BrowseNode • Condition • Count • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: SoftwareVideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "SoftwareVideoGames", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • Count	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: VHS

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VHS", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• Publisher • Sort • State • Title

SearchIndex: Video

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Video", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode	• Director • ItemPage • Keywords	• Publisher • Sort • State • Title

SearchIndex: VideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VideoGames", only the following parameters can be used in the request.

• Author • Brand • BrowseNode • Condition • Count	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Watches

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Watches", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage	• Keywords • MinimumPrice • Sort	• State • Title

SearchIndex-ItemSearch Parameter Combinations for JP

This Search Index Matrix shows you which `ItemSearch` parameters may be used with each of the available *`SearchIndex`* values for the JP locale.

Topics

SearchIndex: All

In `ItemSearch` request, when *`SearchIndex`* equals "All", only the following parameters can be used in a request.

- Keywords

SearchIndex: Apparel

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "apparel", only the following parameters can be used in the request.

• Author • Availability • Brand • BrowseNode • Condition • Count • DeliveryMethod	• Format • ItemPage • Keywords • Magazines • Manufacturer	• Performer • Sort • State • Title

SearchIndex: Automotive

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Automotive", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • Keywords	• Manufacturer • MaximumPrice • MinimumPrice	• Sort • Title

SearchIndex: Baby

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Baby", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • State • Title

SearchIndex: Beauty

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals " Beauty", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Blended

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Blended", only the following parameters can be used in the request.

- Keywords

SearchIndex: Books

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VALUE", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • ItemPage	• Keywords • MaximumPrice • MinimumPrice	• Power • Publisher • Sort • Title

SearchIndex: Classical

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Classical", only the following parameters can be used in the request.

• Actor • Availability • BrowseNode • Composer • Conductor • Count	• Format • ItemPage • Keywords • Magazines • MusicLabel • Orchestra	• Performer • Sort • State • Title

SearchIndex: DVD

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "DVD", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• Publisher • Sort • State • Title

SearchIndex: Electronics

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VALUE", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage	• Keywords • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: ForeignBooks

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "ForeignBooks", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • ItemPage	• Keywords • MaximumPrice • MinimumPrice	• Publisher • Sort • Title

SearchIndex: Grocery

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Beauty", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Count	• ItemPage • Keywords • MaximumPrice • MinimumPrice	• Manufacturer • Sort • Title

SearchIndex: HealthPersonalCare

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "HealthPersonalCare", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Hobbies

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Hobbies", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: HomeImprovement

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "HomeImprovement", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • DeliveryMethod	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Jewelry

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Jewelry", only the following parameters can be used in the request.

• Availability • BrowseNode	• DeliveryMethod • Keywords	• Sort • Title

SearchIndex: Kitchen

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VALUE", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: MP3Downloads

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MP3Downloads", only the following parameters can be used in the request.

• Availability • BrowseNode • DeliveryMethod	• Keywords • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Music

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Music", only the following parameters can be used in the request.

• Artist • Availability • BrowseNode • Count • Format	• ItemPage • Keywords • Magazines • MusicLabel • Performer	• Sort • State • Title

SearchIndex: MusicalInstruments

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicalInstruments", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• MaximumPrice • MinimumPrice	• Keywords • Sort • Title

SearchIndex: MusicTracks

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicTracks", only the following parameters can be used in the request.

• Condition • ItemPage	• Keywords • MaximumPrice	• MinimumPrice • Sort

SearchIndex: OfficeProducts

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "OfficeProducts", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • DeliveryMethod	• Keywords • MaximumPrice • MinimumPrice	• Sort • Title

SearchIndex: Shoes

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Shoes", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • DeliveryMethod	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Software

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Software", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: SportingGoods

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "SportingGoods", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Toys

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Toys", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: VHS

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VHS", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• Publisher • Sort • State • Title

SearchIndex: Video

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Video", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• Publisher • Sort • State • Title

SearchIndex: VideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VideoGames", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Watches

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Watches", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage	• Keywords • MinimumPrice • Sort	• State • Title

SearchIndex-ItemSearch Parameter Combinations for UK

This Search Index Matrix shows you which `ItemSearch` parameters may be used with each of the available *`SearchIndex`* values for the UK locale.

Topics

SearchIndex: All

In `ItemSearch` request, when *`SearchIndex`* equals "All", only the following parameters can be used in a request.

- Keywords

SearchIndex: Automotive

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Automotive", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • DeliveryMethod	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Blended

- Keywords

SearchIndex: Apparel

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "apparel", only the following parameters can be used in the request.

• Author • Availability • Brand • BrowseNode • Condition • Count • DeliveryMethod	• Format • ItemPage • Keywords • Magazines • Manufacturer	• Performer • Sort • State • Title

SearchIndex: Baby

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Baby", only the following parameters can be used in the request.

• Author • Availability • Brand • BrowseNode • Condition • Count • DeliveryMethod	• Format • ItemPage • Keywords • Magazines • Manufacturer	• Performer • Sort • State • Title

SearchIndex: Beauty

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Beauty", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Count	• ItemPage • Keywords • MaximumPrice • MinimumPrice	• Manufacturer • Sort • Title

SearchIndex: Books

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VALUE", only the following parameters can be used in the request.

• Author • BrowseNode • Condition • ItemPage	• Keywords • MaximumPrice • MinimumPrice	• Power • Publisher • Sort • Title

SearchIndex: Classical

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Classical", only the following parameters can be used in the request.

• Actor • Availability • BrowseNode • Composer • Conductor • Count	• Format • ItemPage • Keywords • Magazines • MusicLabel • Orchestra	• Performer • Sort • State • Title

SearchIndex: DVD

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals DVD", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• Publisher • Sort • State • Title

SearchIndex: Electronics

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals Electronics", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Grocery

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Grocery", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• MaximumPrice • MinimumPrice	• Keywords • Sort • Title

SearchIndex: HealthPersonalCare

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VALUE", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage • Brand	• BrowseNode • Condition • ItemPage • Keywords • Manufacturer	• MaximumPrice • MinimumPrice • Sort • Title

SearchIndex: HomeGarden

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "HomeGarden", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Jewelry

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Jewelry", only the following parameters can be used in the request.

• Availability • BrowseNode • DeliveryMethod	• Keywords	• Sort • Title

SearchIndex: Kitchen

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VALUE", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Lighting

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Lighting", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • DeliveryMethod	• Keywords • MaximumPrice • MinimumPrice	• Sort • Title

SearchIndex: MP3Downloads

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MP3Downloads", only the following parameters can be used in the request.

• Author • Availability • BrowseNode • Count • DeliveryMethod • ItemPage	• Keywords • MaximumPrice • MinimumPrice • ResponseGroup • ReviewSort	• SearchIndex • Sort • TagPage • TagsPerPage • TagSort • Title

SearchIndex: Music

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals " Music", only the following parameters can be used in the request.

• Artist • Availability • BrowseNode • Count • Format	• ItemPage • Keywords • Magazines • MusicLabel • Performer	• Sort • State • Title

SearchIndex: MusicalInstruments

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicalInstruments", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• MaximumPrice • MinimumPrice	• Keywords • Sort • Title

SearchIndex: MusicTracks

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicTracks", only the following parameters can be used in the request.

• Condition • ItemPage	• Keywords • MaximumPrice	• MinimumPrice • Sort

SearchIndex: OfficeProducts

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "OfficeProducts", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• DeliveryMethod • Keywords • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: OutdoorLiving

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "OutdoorLiving", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Outlet

In `ItemSearch` request, when *`SearchIndex`* equals "Outlet", only the following parameters can be used in a request.

- Keywords

SearchIndex: Shoes

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Shoes", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Count • ItemPage	• Keywords • MaximumPrice • MinimumPrice • Manufacturer	• Power • Sort • Title

SearchIndex: Software

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Software", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: SoftwareVideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "SoftwareVideoGames", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Toys

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Toys", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: VHS

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VHS", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• Publisher • Sort • State • Title

SearchIndex: Video

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VALUE", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• Publisher • Sort • State • Title

SearchIndex: VideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VALUE", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice	• MinimumPrice • Sort • Title

SearchIndex: Watches

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Watches", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage	• Keywords • MinimumPrice • Sort	• State • Title

SearchIndex-ItemSearch Parameter Combinations for US

This Search Index Matrix shows you which `ItemSearch` parameters may be used with each of the available *`SearchIndex`* values for the US locale.

Topics

SearchIndex: All

In `ItemSearch` request, when *`SearchIndex`* equals "All", only the following parameters can be used in a request.

- Keywords

SearchIndex: Apparel

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "apparel", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • State • TextStream • Title

SearchIndex: Appliances

In `ItemSearch` request,when *`SearchIndex`* equals "Appliances", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • Keywords	• Manufacturer • MaximumPrice • MerchantId • MinimumPrice	• Sort • Title

SearchIndex: ArtsAndCrafts

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "ArtsAndCrafts", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• Keywords • Manufacturer • MaximumPrice	• MerchantId • MinimumPrice • Sort • Title

SearchIndex: Automotive

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Automotive", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • State • TextStream • Title

SearchIndex: Baby

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Baby", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MerchantId • MinimumPrice • Sort • State • Title

SearchIndex: Beauty

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals " Beauty", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MerchantId • MinimumPrice • Sort • Title

SearchIndex: Blended

- Keywords

SearchIndex: Books

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Books", only the following parameters can be used in the request.

• Author • BrowseNode • Condition	• ItemPage • Keywords • MaximumPrice • MerchantId • MinimumPrice	• Power • Publisher • Sort • State • TextStream • Title

SearchIndex: Classical

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Classical", only the following parameters can be used in the request.

• Artist • Author • Availability • BrowseNode • Composer • Conductor	• Count • Format • ItemPage • Keywords • Magazines • MusicLabel	• Orchestra • Performer • PostalCode • Sort • State • Title

SearchIndex: DigitalMusic

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "DigitalMusic", only the following parameters can be used in the request.

• Actor • BrowseNode • Condition • Director • ItemPage	• Keywords • MaximumPrice • MerchantId • MinimumPrice	• Sort • State • Title

SearchIndex: Grocery

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Grocery", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode	• MaximumPrice • MerchantId • MinimumPrice	• Keywords • Sort • Title

SearchIndex: MP3Downloads

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MP3Downloads", only the following parameters can be used in the request.

- Browsenode
- Keyword
- Title

SearchIndex: DVD

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "DVD", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• PostalCode • Publisher • Sort • State • Title

SearchIndex: Electronics

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Electronics", only the following parameters can be used in the request.

• BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MerchantId • MinimumPrice • Sort • State • TextStream • Title

SearchIndex: HealthPersonalCare

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "HealthPersonalCare", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• Manufacturer • MaximumPrice • MerchantId • MinimumPrice	• ItemPage • Keywords • Sort • State • Title

SearchIndex: HomeGarden

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: Industrial

- BrowseNode
- Keywords
- Manufacturer
- Title

SearchIndex: Jewelry

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Jewelry", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• MaximumPrice • MerchantId • MinimumPrice • Sort	• State • TextStream • Title

SearchIndex: KindleStore

• Author • BrowseNode • Condition • ItemPage	• Keywords • MaximumPrice • MerchantId • MinimumPrice	• Power • Publisher • Sort • TextStream • Title

SearchIndex: Kitchen

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Kitchen", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Manufacturer	• MaximumPrice • MerchantId • MinimumPrice • ItemPage	• Keywords • Sort • State • Title

SearchIndex: Magazines

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Magazines", only the following parameters can be used in the request.

• BrowseNode • Condition	• ItemPage • Keywords • MaximumPrice • MerchantId	• MinimumPrice • Publisher • State • Sort • Title

SearchIndex: Merchants

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Merchants", only the following parameters can be used in the request.

• Keywords • Title • BrowseNode • Artist • Author • Actor • Director	• AudienceRating • Manufacturer • MusicLabel • Composer • Publisher • Brand	• Conductor • City • Neighborhood • Orchestra • TextStream

SearchIndex: Miscellaneous

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Miscellaneous", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • MaximumPrice	• MerchantId • MinimumPrice • Sort • State • Title

SearchIndex: MobileApps

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MobileApps", only the following parameters can be used in the request.

• Availability • Author • Brand • BrowseNode	• Keywords • Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

SearchIndex: Music

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Music", only the following parameters can be used in the request.

• Artist • Availability • BrowseNode • Count • Format	• ItemPage • Keywords • Magazines • MusicLabel • Performer	• PostalCode • Sort • State • Title

SearchIndex: MusicalInstruments

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicalInstruments", only the following parameters can be used in the request.

• Availability • Brand • BrowseNode • Condition • Manufacturer	• MaximumPrice • MerchantId • MinimumPrice	• ItemPage • Keywords • Sort • Title

SearchIndex: MusicTracks

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "MusicTracks", only the following parameters can be used in the request.

• Condition • ItemPage	• Keywords • MaximumPrice • MerchantId	• MinimumPrice • Sort • State

SearchIndex: OfficeProducts

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "OfficeProducts", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Manufacturer	• MaximumPrice • MerchantId • MinimumPrice • ItemPage	• Keywords • Sort • State • Title

SearchIndex: OutdoorLiving

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "OutdoorLiving", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Manufacturer	• MaximumPrice • MerchantId • MinimumPrice • ItemPage	• Keywords • Sort • State • Title

SearchIndex: PCHardware

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "PCHardware", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Manufacturer	• MaximumPrice • MerchantId • MinimumPrice • ItemPage	• Keywords • Sort • State • Title

SearchIndex: PetSupplies

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "PetSupplies", only the following parameters can be used in the request.

• Author • Brand • BrowseNode • Condition • Count	• ItemPage • Keywords • Manufacturer • MaximumPrice	• MerchantId • MinimumPrice • Sort • State • Title

SearchIndex: Photo

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Photo", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Manufacturer • MaximumPrice	• MerchantId • MinimumPrice • ItemPage • Keywords • Sort	• State • TextStream • Title

SearchIndex: Shoes

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "shoes", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition	• ItemPage • Keywords • Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • State • TextStream • Title

SearchIndex: Software

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Software", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • Manufacturer	• MaximumPrice • MerchantId • MinimumPrice • ItemPage	• Keywords • Sort • State • Title

SearchIndex: SportingGoods

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "SportingGoods", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • State • Title

SearchIndex: Tools

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Tools", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice • MerchantId	• MinimumPrice • Sort • State • Title

SearchIndex: Toys

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Toys", only the following parameters can be used in the request.

• BrowseNode • Condition	• ItemPage • Keywords • MaximumPrice	• MerchantId • MinimumPrice • Sort • State • TextStream • Title

SearchIndex: UnboxVideo

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "UnboxVideo", only the following parameters can be used in the request.

• Actor • BrowseNode • Condition • Director	• ItemPage • Keywords • MaximumPrice • MerchantId • MinimumPrice	• MPAARating • Publisher • Sort • State • TextStream • Title

SearchIndex: VHS

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VHS", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• PostalCode • Publisher • Sort • State • Title

SearchIndex: Video

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Video", only the following parameters can be used in the request.

• Actor • AudienceRating • Availability • BrowseNode • Count • Director	• Format • ItemPage • Keywords • Magazines • Performer	• PostalCode • Publisher • Sort • State • Title

SearchIndex: VideoGames

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "VideoGames", only the following parameters can be used in the request.

• Brand • BrowseNode • Condition • ItemPage	• Keywords • Manufacturer • MaximumPrice • MerchantId • MinimumPrice	• Sort • State • TextStream • Title

SearchIndex: Watches

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Watches", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage • Keywords	• MerchantId • MinimumPrice • Sort	• State • TextStream • Title

SearchIndex: Wireless

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "Wireless", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage	• Keywords • MaximumPrice • MerchantId • MinimumPrice	• Sort • State • Title

SearchIndex: WirelessAccessories

In an `ItemSearch` request, when the *`SearchIndex`* parameter equals "WirelessAccessories", only the following parameters can be used in the request.

• BrowseNode • Condition • ItemPage	• Keywords • MaximumPrice • MerchantId	• MinimumPrice • Sort • Title

Search Index Support by Locale

The following table provides a list of the search indices available by locale. A check mark in the following table indicates that a search index is available in a particular locale.

Note

All search index names are case-sensitive.

Search Index Name	US	UK	DE	JP	FR	CA	
All	✓	✓	✓	✓	✓	✓	
Apparel	✓	✓	✓	✓	✓		
Appliances	✓						
ArtsAndCrafts	✓						
Automotive	✓	✓	✓	✓			
Baby	✓	✓	✓	✓	✓		
Beauty	✓	✓	✓	✓	✓		

Search Index Name	US	UK	DE	JP	FR	CA	
Blended	✓	✓	✓	✓	✓	✓	
Books	✓	✓	✓	✓	✓	✓	
Classical	✓	✓	✓	✓	✓	✓	
DigitalMusic	✓						
DVD	✓	✓	✓	✓	✓	✓	
Electronics	✓	✓	✓	✓	✓	✓	
ForeignBooks			✓	✓	✓	✓	
GourmetFood	✓						
Grocery	✓	✓	✓	✓			
HealthPersonalCare	✓	✓	✓	✓	✓		
Hobbies				✓			
HomeGarden	✓	✓	✓				
HomeImprovement		✓	✓	✓			
Industrial	✓						
Jewelry	✓	✓	✓	✓	✓		
KindleStore	✓	✓					
Kitchen	✓	✓	✓	✓	✓		
Lighting		✓	✓		✓		
Magazines	✓		✓				
Miscellaneous	✓						
MobileApps	✓						
MP3Downloads	✓	✓	✓	✓	✓		
Music	✓	✓	✓	✓	✓	✓	

Search Index Name	US	UK	DE	JP	FR	CA	
MusicalInstruments	✓	✓	✓	✓	✓		
MusicTracks	✓	✓	✓	✓	✓		
OfficeProducts	✓	✓	✓	✓	✓		
OutdoorLiving	✓	✓	✓				
Outlet		✓	✓				
PCHardware	✓	✓	✓		✓		
PetSupplies	✓						
Photo	✓		✓				
Shoes	✓	✓	✓	✓	✓		
Software	✓	✓	✓	✓	✓	✓	
SoftwareVideoGames		✓	✓		✓	✓	
SportingGoods	✓	✓	✓	✓	✓		
Tools	✓	✓	✓				
Toys	✓	✓	✓	✓	✓		
UnboxVideo	✓						
VHS	✓	✓	✓	✓	✓	✓	
Video	✓	✓	✓	✓	✓	✓	
VideoGames	✓	✓	✓	✓	✓	✓	
Watches	✓	✓	✓		✓		
Wireless	✓						
WirelessAccessories	✓						

Browse Node IDs

The following table presents browse node IDs by search index and locale. These IDs represent the top level browse nodes only, You can use these IDs in a `BrowseNodeLookup` request to get additional browse node IDs.

These IDs were valid as of the publication date of this guide.

	CA	DE	FR	JP	UK	US
Apparel		78689031	340855031	361299011	83451031	1036682
Appliances						2619525011
Arts and Crafts						2617941011
Automotive		78194031		2017304051	248877031	15690151
Baby		357577011	206617031	13331821	60032031	1036682
Beauty		64257031	197858031	52391051	66280031	11055981
Books	927726	541686	468256	465610	1025612	1000
Classical	962454	542676	537366	562032	505510	301668
DigitalMusic						195208011
DVD	14113311	547664	578608	562002	283926	130
Electronics	677211011	569604	1058082	3210991	560800	493964
ForeignBooks	927726	54071011	69633011	388316011		
GourmetFood						3580501
Grocery		340846031		57239051	340834031	16310101
HealthPersonalCare		64257031	197861031	161669011	66280031	3760931
Hobbies				13331821		
HomeGarden		10925241			11052591	285080
Homeimprovement				2016929051		
Industrial						228239
Jewelry		327473011	193711031	85896051	193717031	3880591
KindleStore		530484031			341677031	133141011
Kitchen	2206275011	3169011	57686031	3839151	11052591	1063498
Lighting		213083031	213080031		213077031	
Magazines		1198526				599872
Miscellaneous						10304191
MobileApps						2350149011

	CA	DE	FR	JP	UK	US
MP3Downloads		77256031	206442031	2128134051	77198031	195211011
Music	962454	542676	537366	562032	505510	301668
MusicalInstruments		11965861	11965861	11965861	11965861	11965861
OfficeProducts		16291311	192420031		560800	1084128
OutdoorLiving		10925051			11052591	1063498
PCHardware		569604				493964
PetSupplies						1063498
Photo		569604				493964
Shoes			215934031	2016926051		
Software	3234171	542064	548012	637630	1025614	409488
SoftwareVideoGames	3323751	541708	548014		1025616	
SportingGoods		16435121		14315361	319530011	1079730
Tools					11052591	468240
Toys		12950661	548014	13331821	712832	493964
VHS	962072	547082	578610	561972	283926	404272
Video	962454	547664	578608	561972	283926	130
VideoGames	110218011	541708	548014	637872	1025616	493964
Watches		193708031	60937031	14315361	595312	1079730
Wireless						508494
WirelessAccessories						13900851

Glossary

Access Key ID	An alphanumeric token that uniquely identifies a request sender. This ID is associated with your Secret Access Key.
accessory	An related items, such as a camera, its carrying case and USB cord.
active area	In a shopping cart, this area contains the items that are ready to be purchased.
Amazon marketplace	The feature by which Amazon relates seller items directly to retail catalog items and content. This feature leverages the retail buying experience and content to third-party seller items. .
ASIN	(Amazon Standard Item Number) An alphanumeric token generated and assigned by Amazon that uniquely identifies an item for sale by Amazon. ASINs are unique within a locale, not across locales..
buyer	A person(s) agreeing to exchange money for an item delivered by a seller.
collection	An ensemble of related items, such as a coordinated set of sheets, pillow cases, and curtains.
EISBN	(Electronic International Standard Book Number) A token that uniquely identifies a digital book.
ISBN	(International Standard Book Number) An alphanumeric token that uniquely identifies a book. A book's EAN is typically set equal to the book's ISBN
JAN	(Japanese Article Number) The equivalent of the EAN that is used in Japan for products and barcodes.
locale	A region in which Product Advertising API is offered. In each locale, Product Advertising API offers support in the local language, currency, customs, and local shipping. Currently, there are seven locales: CA, DE, FR, JP, UK, and US.
merchant	For the purposes of this document, a merchant is a third party who is selling on the Amazon.com web site but handles their own fulfillment. Merchants help expand the Amazon.com catalog of sellable items by providing Product Data for products Amazon.com currently has no data on, or relate their products to products already identified in the Amazon retail catalogs. Examples include Circuit City (both from a drop ship and in-store pickup point of view) and Ingram Micro (PC Store). For the purposes of the Bulk DP Creation project, "Merchant" does not include Toys R Us (since we carry all of their online inventory in our DCs).

marketplace	a place, physical or otherwise, where merchants offer goods for sale, and customers accept offers, place orders and make purchases. It is the logical container for all data and attributes associated with an eCommerce site. A marketplace is identified by a globally unique MarketplaceId. Marketplaces may be specific to data centers, but they should be considered global. They have a default language code and currency code, but they should be considered capable of multi-lingual and multi-currency support in designs. Only Amazon's Marketplaces are country-specific due to legacy reasons.
MarketplaceId	An alphanumeric token that identifies a locale, for example, DE: A1PA6795UKMFR9), FR: A13V1IB3VIYZZH, or a large merchant, for example, Amazon: ATVPDKIKX0DER.
minimum advertised price (MAP)	The minimum price that a manufacturer allows a product to be advertised at.
offer	Item offered for sale. An offer is not associated with a merchant.
Offering	Offerings are the specific instances of a Merchant's possible stock items. A Merchant "can" have unique offerings for each product they carry and have either. A specific quantity associated with each product A stock availability state {In-Stock, Out-Of-Stock, Back-Ordered, Discontinued, Obsolete, On-Order, etc?} Saleable status is determined by positive inventory states of available quantity > 0, or ?In stock?. Offerings will have transactional rules (max quantity purchasable), condition types ('new', 'used'), and tax codes associated with them An offering is not a buyable entity on its own. An offering with an immediate availability relationship can be purchased. An Offering with a delayed availability relationship can only be ordered.
OfferListing	An item offered for sale by a specific seller or merchant.This is the inventory level specific buyable entity. It ties the merchant/customer, item, item location, pricing, and availability together. These currently represent items ?in-stock? that are essentially shipped at point of sale.
Product	Something offered for sale. Each product has an ASIN. There may be many offerings for a particular product. Product is synonymous with "item".
SaveForLater	Contains items that a customer has chosen to buy but are currently unavailable, or items that a customer has designated they want to save and buy later.
Secret Access Key	A key assigned to you by Amazon Web Services (AWS) when you sign up for an AWS account. Used for request authentication.
seller	Sellers are individuals or companies that sell their own items. For most sellers, the items they sell must already be for sale on Amazon. The exception is the Amazon Advantage seller, who sells items that are not on Amazon.
Seller Central	A web site that allows merchants to manage their web sites and listings. Go to http://sellercentral.amazon.com
Single Detail Page (SDP)	The purpose of Single Detail Page is to expose, for any given retail ASIN, all instances of a product that can be purchased from the Amazon.com site. The display for an ASIN would incorporate links to alternate purchase channels for the same item. For example a books detail page would list a new book from Amazon.com and used/rare/collectable editions from fixed price MarketPlace sellers.
SKU	(Stock Keeping Unit) A merchant-specific identifier for a purchasable good, like a shirt or chair. Amazon's version of the SKU is the ASIN.

UPC	(Universal Product Code) A 12-digit item identifier used in the US and CA locales. The UPC is identifier used in barcodes.
variant	A single version of a given product that has variations. A variant is an item defined by its product, and the unique set of property values from each property in the full set of required properties for that item.
variation	Different versions of a given product. If a pair of pants is available in khaki and olive, then the pants are said to have two variations. Variations are one of the two types of product relationships we support (ensembles is the other).

Document Conventions

This section lists the common typographical and symbol use conventions for AWS technical publications.

Typographical Conventions

This section describes common typographical use conventions.

Convention	Description/Example
Call-outs	A call-out is a number in the body text to give you a visual reference. The reference point is for further discussion elsewhere. You can use this resource regularly. 1
Code in text	Inline code samples (including XML) and commands are identified with a special font. You can use the command `java -version`.

Code blocks	Blocks of sample code are set apart from the body and marked accordingly.
	`# ls -l /var/www/html/index.html`
	`-rw-rw-r--  1 root root 1872 Jun 21 09:33 /var/www/html/index.html`
	`# date`
	`Wed Jun 21 09:33:42 EDT 2006`

Emphasis	Unusual or important words and phrases are marked with a special font. You *must* sign up for an account before you can use the service.
Internal cross references	References to a section in the same document are marked.
Logical values, constants, and regular expressions, abstracta	A special font is used for expressions that are important to identify, but are not code. If the value is `null`, the returned response will be `false`.

Convention	Description/Example
Product and feature names	Named AWS products and features are identified on first use. Create an *Amazon Machine Image* (AMI).
Operations	In-text references to operations. Use the `GetHITResponse` operation.
Parameters	In-text references to parameters. The operation accepts the parameter *`AccountID`*.
Response elements	In-text references to responses. A container for one `CollectionParent` and one or more `CollectionItems`.
Technical publication references	References to other AWS publications. If the reference is hyperlinked, it is also underscored. For detailed conceptual information, see the *Amazon Mechanical Turk Developer Guide*.
User entered values	A special font marks text that the user types. At the password prompt, type **`MyPassword`**.
User interface controls and labels	Denotes named items on the UI for easy identification. On the **File** menu, click **Properties**.
Variables	When you see this style, you must change the value of the content when you copy the text of a sample to a command line. % ec2-register *`<your-s3-bucket>`*/image.manifest See also the following symbol convention.

Symbol Conventions

This section describes the common use of symbols.

Convention	Symbol	Description/Example
Mutually exclusive parameters	(Parentheses | and | vertical | bars)	Within a code description, bar separators denote options from which one must be chosen. `% data = hdfread (start | stride | edge)`

Optional parameters
XML variable text
	[square brackets]	Within a code description, square brackets denote completely optional commands or parameters.
		`% sed [-n, -quiet]`
		Use square brackets in XML examples to differentiate them from tags.
		`<Date>[date]</Date>`

Variables	<arrow brackets>	Within a code sample, arrow brackets denote a variable that must be replaced with a valid value. `% ec2-register <your-s3-bucket>/image.manifest`

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X