

## PA-API Getting Started Guide

This guide provides a step by step explanation on how to send a request to the Product Advertising API (PA-API) and display the results on a website using php.

At the end of this guide you should be able to retrieve a **Product Title**, **Product Picture**, the current **Amazon Price** and the **Link to the Product Detail Page** via PA-API for a certain product and display this information on your website.

## What are the Requirements?

If not already done, join the [Amazon Associate Program](#) and [get your AWS Access Key](#). To implement PA-API on your website you need to be familiar with the following:

- XML
- Basic understanding of web services
- php

## Anatomy of a PA-API Request

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

```
http://ecs.amazonaws.de/onca/xml?Service=AWSECommerceService&AWSAccessKeyId=[Access Key ID]&AssociateTag=[ID] &Version=2011-08-01&Operation=ItemLookup&ItemId=B004Q3QSWQ&Timestamp=[YYYY-MM-DDThh:mm:ssZ&Signature=[Request Signature]
```

Let's have a closer look at each part of the request

1	<code>http://ecs.amazonaws.de/onca/xml?</code>	The first part describes the domain, where the request is being sent to. The domain varies for each marketplace.	} remains the same for each request
2	<code>Service=AWSECommerceService&amp;</code>	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.	
3	<code>AWSAccessKeyId=[Access Key ID]&amp;</code>	Your AWS Key	
4	<code>AssociateTag=[ID]&amp;</code>	Your Associate Tag. All Links returned from PA-API will include this Associate Tag. This is necessary to be eligible to receive a referral fee for a customer's purchase.	
5	<code>Version=2011-08-01&amp;</code>	Specifies the Version of PA-API. Make sure to always use the latest version.	
6	<code>Operation=ItemLookup&amp;</code>	Its value is one of the Product Advertising API operations. (e.g. ItemLookup or ItemSearch)	} paramters that change between requests
7	<code>ItemId=B004Q3QSWQ&amp;</code>	ItemId is an operational parameter. Depending on the Operation a request can contain up to ten parameters.	
8	<code>ResponseGroups=Large,Offers</code>	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.	

9	<b>Timestamp</b> =[YYYY-MM-DDThh:mm:ssZ&	Current timestamp	} mandatory for authentication purposes
10	<b>Signature</b> =[Request Signature]	Signature for authentication	

## Which marketplaces are supported by PA-API?

The following table lists all supported countries and endpoints for Product Advertising API requests.

Locale	Endpoint
CA	http://ecs.amazonaws.ca/onca/xml
CN	http://webservices.amazon.cn/onca/xml
DE	http://ecs.amazonaws.de/onca/xml
ES	http://webservices.amazon.es/onca/xml
FR	http://ecs.amazonaws.fr/onca/xml
IT	http://webservices.amazon.it/onca/xml
JP	http://ecs.amazonaws.jp/onca/xml
UK	http://ecs.amazonaws.co.uk/onca/xml
US	http://webservices.amazon.com/onca/xml

## What types of Operations are supported by PA-API?

The following Operations are supported by PA-API. Click on a link to learn more about each Operation.

- [BrowseNodeLookup](#)
- [CartAdd](#)
- [CartClear](#)
- [CartCreate](#)
- [CartGet](#)
- [CartModify](#)
- [ItemLookup](#)
- [ItemSearch](#)
- [SimilarityLookup](#)

**ItemSearch** & **ItemLookup** are the most common Operations. “CartX” Operations can be used for virtual shopping cart management.

In our example we use **ItemLookup** to return item attributes for a given item identifier. The item identifier can either be the ASIN, EAN, SKU, UPC or ISBN. In case you don’t have any of those identifiers use **ItemSearch** to look up a product.

## Which operational parameters are available?

The set of operational parameters depends on the type of Operation. Please refer to [the technical documentation for each Operation](#) to see which parameters are available. In our example we used **ItemId** as the only operational parameter. Optionally we could add a **Condition** and/or **MerchantId** parameter and limit the offers returned, for example, to new retail offers only.

## What are Response Groups?

The ResponseGroups parameter specifies the information returned from PA-API. Each Response group contains a set of attributes which will be returned. For example “Small” returns the following elements:

- Actor
- Artist
- ASIN
- Author
- Creator
- Director
- Keywords
- Manufacturer
- Message
- ProductGroup
- Role
- Title
- TotalPages
- TotalResults

The following Response Groups are available. Click on a link to learn which elements are returned by each Response Group.

- [Accessories](#)
- [AlternateVersions](#)
- [BrowseNodeInfo](#)
- [BrowseNodes](#)
- [Cart](#)
- [CartNewReleases](#)
- [CartTopSellers](#)
- [CartSimilarities](#)
- [Collections](#)
- [EditorialReview](#)
- [Images](#)
- [ItemAttributes](#)
- [ItemIds](#)
- [LargeMedium](#)
- [MostGifted](#)
- [MostWishedFor](#)
- [NewReleases](#)
- [OfferFull](#)
- [OfferListings](#)
- [Offers](#)
- [OfferSummary](#)
- [PromotionSummary](#)
- [RelatedItemsRequest](#)
- [Reviews](#)
- [SalesRank](#)
- [SearchBinsSimilarities](#)
- [SmallTopSellers](#)
- [Tracks](#)
- [Variations](#)
- [VariationImages](#)
- [VariationMatrix](#)
- [VariationOffers](#)
- [VariationSummary](#)

In our example we use the “Large” and “Offers” Response Groups.

## How can I use PA-API with php?

In the first step we need to set the variables according to our request.

```
$public_key = "your aws key";
$private_key = "your secret aws key";
$host = "ecs.amazonaws.de";
$request_parameters['Operation'] = "ItemLookup";
$request_parameters['ItemId'] = "B004Q3QSWQ";
$request_parameters['ResponseGroup'] = "Large,Offers";
$request_parameters['AssociateTag'] = "my_tracking_id-21";
```

In the second step we call a function which basically turns our input variables into a signed PA-API request, makes that request and returns the server response as an XML Object.

```
$response = aws_signed_request($host, $request_parameters, $public_key, $private_key);
```

Let’s have a closer look at the function `aws_signed_request`<sup>1</sup>:

<sup>1</sup> Source: [http://mierendo.com/software/aws\\_signed\\_query/](http://mierendo.com/software/aws_signed_query/)

```
function aws_signed_request($host, $params, $public_key, $private_key)
{
    // some parameters
    $method = "GET";
    $uri     = "/onca/xml";
    // additional parameters
    $params["Service"]      = "AWSECommerceService";
    $params["AWSAccessKeyId"] = $public_key;
    // GMT timestamp
    $params["Timestamp"]    = gmdate("Y-m-d\TH:i:s\Z");
    // API version
    $params["Version"]      = "2011-10-01";
}
```

In this first part of the function additional parameters are set. The next step is to build the signed request according to [the technical documentation](#):

### 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.

```
// sort the parameters
ksort($params);

// create the canonicalized query
$canonicalized_query = array();
foreach ($params as $param => $value)
{
    $param      = str_replace("%7E", "~", rawurlencode($param));
    $value      = str_replace("%7E", "~", rawurlencode($value));
    $canonicalized_query[] = $param . "=" . $value;
}
$canonicalized_query = implode("&", $canonicalized_query);
```

### 2. Create the string to sign

```
// create the string to sign
$string_to_sign = $method . "\n" . $host . "\n" . $uri . "\n" . $canonicalized_query;
```

### 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 + convert the resulting value to base64.

```
// calculate HMAC with SHA256 and base64-encoding
$signature = base64_encode(hash_hmac("sha256", $string_to_sign, $private_key, True));
```

#### 4. URL encoded the signature as specified in RFC 3986

```
// encode the signature for the request
$signature = str_replace("%7E", "~", rawurlencode($signature));
```

#### 5. Put together the final request

```
// create request
$request = "http://" . $host . $uri . "?" . $canonicalized_query . "&Signature=" .
$signature;
```

### Send the Request to the PA-API Server

After the signed request was created, it now needs to be sent to the PA-API Server. There are several ways to do that in php. In this example the [cURL library](#) is used.

```
// Initialize cURL Session
$ch = curl_init();
// tell cURL which URL to open
curl_setopt($ch, CURLOPT_URL, $request);
// suppress header output
curl_setopt($ch, CURLOPT_HEADER, 0);
// reply should be a string
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
// make request and save response in $response
$response = curl_exec($ch);
// close cURL session
curl_close($ch);
```

At this point the server response with the product information we are looking for is now stored as a string in the `$response` variable. The plain server response looks like the following:

```
<?xml version="1.0"?>
<ItemLookupResponse xmlns="http://webservices.amazon.com/AWSECommerceService/2011-08-01">
<OperationRequest>
  <RequestId>747dae65-9a0d-4179-aa7b-9fbe0059356a</RequestId>
  <Arguments>
    <Argument Name="Operation" Value="ItemLookup"/>
    <Argument Name="Service" Value="AWSECommerceService"/>
    <Argument Name="AssociateTag" Value="my-assoc-tag-21"/>
    <Argument Name="Version" Value="2011-08-01"/>
    <Argument Name="Signature" Value="RQkF48Nbv8mt/tp6w7yBJ5SICaKTOqAptkq1EMQ3atM="/>
    <Argument Name="ItemId" Value="B004Q3QSWQ"/>
    <Argument Name="AWSAccessKeyId" Value="your access key"/>
    <Argument Name="Timestamp" Value="2012-03-16T13:27:07Z"/>
    <Argument Name="ResponseGroup" Value="Large,Offers"/>
  </Arguments>
  <RequestProcessingTime>0.1060150000000000</RequestProcessingTime>
</OperationRequest>
<Items>
  <Request>
    <IsValid>True</IsValid>
    <ItemLookupRequest>
      <IdType>ASIN</IdType>
      <ItemId>B004Q3QSWQ</ItemId>
      <ResponseGroup>Large</ResponseGroup>
      <ResponseGroup>Offers</ResponseGroup>
    </ItemLookupRequest>
  </Request>
```

```

<Item>
  <ASIN>B004Q3QSWQ</ASIN>
  <DetailPageURL>http://www.amazon.de/Samsung-DualCore-Smartphone-Super-Amoled-
  Frontkamera/dp/B004Q3QSWQ%3FSubscriptionId%3DAKIAJAAORR6BKUW3IOZA%26tag%3Dmy-assoc-tag-
  21%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3DB004Q3QSWQ</DetailPa
  geURL>
  <ItemLinks>
    <ItemLink>
      <Description>Add To Wishlist</Description>
      <URL>http://www.amazon.de/gp/registry/wishlist/add-
      item.html%3Fasin.0%3DB004Q3QSWQ%26SubscriptionId%3DAKIAJAAORR6BKUW3IOZA%26tag%3Dmy-assoc-
      tag-21%26linkCode%3Dxm2%26camp%3D2025%26creative%3D12738%26creativeASIN%3DB004Q3QSWQ</URL>
    </ItemLink>
    <ItemLink>
      <Description>Tell A Friend</Description>
      <URL>http://www.amazon.de/gp/pdp/taf/B004Q3QSWQ%3FSubscriptionId%3DAKIAJAAORR6BKUW3IOZA%26t
      ag%3Dmy-assoc-tag-
      21%26linkCode%3Dxm2%26camp%3D2025%26creative%3D12738%26creativeASIN%3DB004Q3QSWQ</URL>
    </ItemLink>
    <ItemLink>
      <Description>All Customer Reviews</Description>
      <URL>http://www.amazon.de/review/product/B004Q3QSWQ%3FSubscriptionId%3DAKIAJAAORR6BKUW3IOZA
      %26tag%3Dmy-assoc-tag-
      21%26linkCode%3Dxm2%26camp%3D2025%26creative%3D12738%26creativeASIN%3DB004Q3QSWQ</URL>
    </ItemLink>
    <ItemLink>
      <Description>All Offers</Description>
      <URL>http://www.amazon.de/gp/offer-
      listing/B004Q3QSWQ%3FSubscriptionId%3DAKIAJAAORR6BKUW3IOZA%26tag%3Dmy-assoc-tag-
      21%26linkCode%3Dxm2%26camp%3D2025%26creative%3D12738%26creativeASIN%3DB004Q3QSWQ</URL>
    </ItemLink>
  </ItemLinks>
  ...

```

## Process the returned server response

The PA-API response is an XML document. The elements returned by a response group are structured. For example, the Item element has several child elements, one of which is Offers, which has a child element, Subtotal, which itself has three child elements: Amount, CurrencyCode, and FormattedPrice, as shown in the following response snippet:

```

<Item>
  <Offers>
    <Subtotal>
      <Amount>999</Amount>
      <CurrencyCode>EUR</CurrencyCode>
      <FormattedPrice>EUR 9,99</FormattedPrice>
    </Subtotal>
  </Offers>

```

The parent-to-child succession of structured elements is called an [XPath](#). To parse a result, the PA-API response is turned into an object and then XPaths are used as an efficient means of finding elements and their values.

[SimpleXML](#)<sup>2</sup> is a corresponding php function, which converts the XML document into an object.

```

// parse XML
$xml = simplexml_load_string($response);

```

<sup>2</sup> SimpleXML [php Manual](#)

We now reached the end of the `aws_signed_request` function and the XML Object is returned.

```
// return XML Object
return $pxml;
}
```

## Display Item Attributes on a website

At this point the XML Object is stored in `$response`. We can now use SimpleXML notation to access the information we need:

```
$response = aws_signed_request($host, $request_parameters, $public_key, $private_key);

$title = $response->Items->Item->ItemAttributes->Title;
$price = ($response->Items->Item->Offers->Offer->OfferListing->Price->Amount)/100;
$dpr_url = $response->Items->Item->DetailPageURL;
$img_url = $response->Items->Item->MediumImage->URL;
```

In the last step, the data is displayed on the website:

```
echo "<b>Title:</b> $title <br />";
echo "<b>Price:</b> $price EUR<br />";
echo "<p><img src='$img_url' /></p>";
echo "<a href='$dpr_url' target='_blank'>Buy now on Amazon.de</a>";
```

The final result should look like that:

**Title:** Samsung Galaxy S II (i9100G) DualCore Smartphone (10.9 cm (4.3 Zoll) Super-Amoled Plus Display, Android 2.3, 8 MP Full-HD Kamera, 2 MP Frontkamera) schwarz  
**Price:** 428.9 EUR



[Buy now on Amazon.de](#)

---

## Where to go from here?

Check out the [PA-API Documentations](#) for more information.

## Common PA-API Requests

---

### Example #1: Search by Keyword

```
AWSAccessKeyId=YOUR_ACCESS_KEY
AssociateTag=YOUR_ASSOC_ID-21
Keywords=Macbook
Operation=ItemSearch
ResponseGroup=Large,Offers
SearchIndex=All
Service=AWSECommerceService
Timestamp=2012-04-19T12:04:28Z
Version=2011-08-01
```

**Response:** PA-API returns the 10 most relevant items. Using the response groups **Large** and **Offers** you receive information such as: ASIN, Title, Brand, Price, DetailPageURL, Image URLs, Review URL


---

### Example #2: Search by Keyword – items 11 to 20

If you want to receive more than 10 items you need to make multiple requests in combination with the `ItemPage` parameter.

```
AWSAccessKeyId=YOUR_ACCESS_KEY
AssociateTag=YOUR_ASSOC_ID-21
ItemPage=2
Keywords=Macbook
Operation=ItemSearch
ResponseGroup=Large,Offers
SearchIndex=All
Service=AWSECommerceService
Timestamp=2012-04-19T12:04:28Z
```

**Response:** PA-API returns the same information as above, but items 11-20.

 Valid Values for `ItemPage`: 1 to 10 (1 to 5 when the search index = "All")

---

### Example #3: Search a product based on EAN

```
AWSAccessKeyId=YOUR_ACCESS_KEY
AssociateTag=YOUR_ASSOC_ID-21
IdType=EAN
ItemId=0885909533718
Operation=ItemLookup
ResponseGroup=Large,Offers
SearchIndex=All
Service=AWSECommerceService
Timestamp=2012-04-19T12:28:34Z
```

**Response:** PA-API returns one item with the same information as in example #1.



#### Example #4: Search products based on a browse node

```
AWSAccessKeyId=YOUR_ACCESS_KEY
AssociateTag=YOUR_ASSOC_ID-21
BrowseNode=430129031
Operation=ItemSearch
ResponseGroup=Large,Offers
SearchIndex=Electronics
Service=AWSECommerceService
Timestamp=2012-04-19T12:32:16Z
Version=2011-08-01
```

**Response:** PA-API returns the same information as in example #1.

☞ Whenever you specify a browse node you also must specify a SearchIndex other than “All”.

---

#### Example #5: Find top sellers for a browse node

```
AWSAccessKeyId=YOUR_ACCESS_KEY
AssociateTag=YOUR_ASSOC_ID-21
BrowseNodeId=430129031
Operation=BrowseNodeLookup
ResponseGroup=TopSellers
Service=AWSECommerceService
Timestamp=2012-04-19T13:06:05Z
Version=2011-08-01
```

**Response:** PA-API returns a list of 10 items with ASIN and Title information.

---

#### Example #6: Get price for a given ASIN (or up to 10 ASINs)

```
AWSAccessKeyId=YOUR_ACCESS_KEY
AssociateTag=YOUR_ASSOC_ID-21
IdType=ASIN
ItemId=B0031SZRZG,B004445JGS,[up to 10 items]
Operation=ItemLookup
ResponseGroup=Offers
Service=AWSECommerceService
Timestamp=2012-04-19T13:13:11Z
```

**Response:** PA-API returns pricing information for each ASIN.

## FAQ

---

### 1. Are there any limitations for PA-API Usage?

Yes, the amount of requests you can make within one hour is limited. By default every user can make 2,000 requests per hour. This limit increases up to 25,000 requests per hour, depending on the amount of revenue you generated during the last 30 days.

### 2. What types of ASINs are returned by ItemSearch?

When SearchIndex = All, child ASINs are returned. When SearchIndex is not ALL parent ASINs are returned.

### 3. What are child and parent ASINs?

Products with variations (for example size, color) have a Parent ASIN and Child ASINs. The Parent ASIN is the generic product (for example, Levi's 501 Jeans). The child ASIN is the specific variation of the product (Levi's 501 Jeans size large and color black).

### 4. Given a parent ASIN – how can I retrieve all variations?

Use ItemLookup with the ResponseGroup "Variations".

### 5. Given a child ASIN – how can I retrieve the parent ASIN and/or all variations?

This is currently not supported by PA-API.

Your question is not answered? Please send your question to [paapi-faq@amazon.com](mailto:paapi-faq@amazon.com). We will regularly update this FAQ section.