JV Amazon Advertising Documentation

Version 1.0 beta - 2010 Jérôme Vieilledent

INTRODUCTION

JV Amazon Advertising allows you to query Amazon huge product catalog by using **Amazon Product Advertising API**.

Monetize your website by silently querying Amazon catalog and providing sponsored links !

With JV Amazon Advertising, you can :

- Do an **Item Search** by keywords, facetting by category (*SearchIndex*)
- Do an **Item Lookup** by an Amazon Product ID (*ASIN*) or *ISBN*
- Look for products related to another
- And all operations supported by Amazon Product Advertising API !

To be able to use this extension, you need to be registered as a developer at Amazon. To provide **sponsored links** and therefore start making money by advertising Amazon Products, you need to subscribe to Amazon affiliate program.

Amazon Product Advertising API documentation.

USAGE

JV Amazon Advertising mainly consists in fetch functions you can use in templates and in PHP :

- *item_search* (search products by keywords)
- *item_lookup* (lookup product information by product ID)

These functions return either a single or an array of **JVAmazonAdvertisingResultItem** object(s) (default, can be customized in *amazonadvertising.ini*). Object attributes depend on Response Groups. Main default attributes are :

- **id** Product ID
- **url** Product URL, localized and with your associate tag if enabled
- **title** Product name
- **image** *JVAmazonAdvertisingImageHolder* object containing image variations

Attributes can be used in PHP and in templates as object properties. Objects can be "var_dumped" in PHP and in templates (with *attribute* operator)

Item Search

Returns an array of **JVAmazonAdvertisingResultItem**

Syntax

eZ Publish template

```
{def $amazonItems = fetch( 'amazonadvertising', 'item_search', hash(
  'keywords'      => 'my_keywords',
  'search_index'  => 'ProductCategory' )}
```

Available parameters

- **keywords** (string) - **required** Keywords to search with
- **search_index** (string) The product category to search. See http://docs.amazonwebservices.com/AWSECommerceService/latest/DG/APPNDX_SearchIndexValues.html
- **item_page** (integer) Retrieves a specific page of items from all of the items in a response
- **sort** Means by which the items in the response are ordered. See http://docs.amazonwebservices.com/AWSECommerceService/latest/DG/APPNDX_SortValuesArticle.html
- **response_group** (array) Specifies the types of values to return. See http://docs.amazonwebservices.com/AWSECommerceService/latest/DG/CHAP_ResponseGroupsList.html
- **raw_params** (array) Params transferred "as is" to AWS. Key is the raw param name. See http://docs.amazonwebservices.com/AWSECommerceService/latest/DG/ItemSearch.html

Item Lookup

Returns a single **JVAmazonAdvertisingResultItem**

Syntax

eZ Publish template

```
{def $amazonItemInfo = fetch( 'amazonadvertising', 'item_lookup', hash(
  'id', '1904811647' )}
```

Available parameters

- **id** (string) - **required** ProductID to lookup
- **id_type** (string) Valid Values: ASIN|SKU|UPC|EAN|ISBN
- **search_index** (string) The product category to search. See http://docs.amazonwebservices.com/AWSECommerceService/latest/DG/APPNDX_SearchIndexValues.html
- **response_group** (array) Specifies the types of values to return. See http://docs.amazonwebservices.com/AWSECommerceService/latest/DG/CHAP_ResponseGroupsList.html
- **raw_params** (array) Params transferred "as is" to AWS. Key is the raw param name. See http://docs.amazonwebservices.com/AWSECommerceService/latest/DG/ItemLookup.html

CONFIGURATION & DEVELOPMENT

For *jvamazonadvertising* to work as expected, you need to configure it in **amazonadvertising.ini** :

Amazon Web Services (AWS) account

When registering to Amazon Product Advertising API, you are given 2 identifiers :

- Access Key ID (your developer ID)
- Secret Access Key (private key used to authenticate your requests)

You can find these identifiers in your developer account page on Amazon (there is a special section for Amazon Product Advertising API) Enter your identifiers in an override of *amazonadvertising.ini*

```
[AWSSettings]
AccessKeyID=your_access_key
SecretAccessKey=your_secret_access_key
```

API version and host

Amazon API is versionned. Currently, *jvamazonadvertising* works with **2009-10-01** version (default). Other versions might work but were not tested

Amazon API is localized. To differentiate AWS calls, Amazon has 2 endpoints (hosts) per locale (http/https). See comments in **amazonadvertising.ini** for more info.

Associate settings

In order to provide sponsored links, you need an Amazon partner account. You can enable/disable sponsored links in config file. To enable :

```
[AssociateSettings]
AssociateEnabled=true
AssociateTag=your_associate_id
```

Result Items and Response Groups

When querying *Amazon Product Advertising API*, all items piece of information returned can be used to *hydrate* PHP value objects (*JVAmazonAdvertisingResultItem* class by default). However, not all information is returned by default.You can ask for more info by specifying one ore several **ResponseGroup** values in your fetch call.

As Response Groups make the response data structure vary, every ResponseGroup need to be handled by a **ResponseGroupHandler**. By default, *jvamazonadvertising* comes with 2 *ResponseGroupHandlers* :

- JVAmazonAdvertisingItemAttributesResponseHandler (will handle *ItemAttributes* - main product info)
- JVAmazonAdvertisingImagesResponseHandler (will handle product Images)

Response Groups configuration

Every AWS query embeds *ResponseGroups* defined in **DefaultResponseGroups[]** config array. Every *ResponseGroup* declared *DefaultResponseGroups[]* MUST be handled by a *ResponseGroupHandler*. *ResponseGroupHandlers* have to be declared in *ResponseGroupHandlers[]* config array. See comments in *amazonadvertising.ini**.

Response Group Handler development

A **ResponseGroupHandler** consists in a simple PHP class implementing *IJVAmazonAdvertisingResponseGroupHandler* interface. This class has at least one method : *handleResult()* that must return an associative array.

```
interface IJVAmazonAdvertisingResponseGroupHandler
{
    /**
     * Must return an associative array to be embedded with the hydrated object (result item)
     * Key is the attribute name (beware of duplicate names). Value can be anything.
     * @param SimpleXMLElement $currentItem Current <Item> XML element returned by AWS
     * @return array
     */
    public function handleResult(SimpleXMLElement $currentItem);
}
```

With a *ResponseGroupHandler*, you enrich your *JVAmazonAdvertisingResultItem* objects by the attributes returned by your handler. These attributes will be available as PHP properties. They will also be available in templates. Example :

```
class dummyResponseHandler implements IJVAmazonAdvertisingResponseGroupHandler
{
    public function handleResult(SimpleXMLElement $currentItem)
    {
        $productID = $currentItem->ASIN;
        $newAttributes = array(
          'product_id'          => $productID,
          'generated_timestamp' => time()
        );

        return $newAttributes;
    }
}
```

Now, your attributes will be available in templates :

```
{def $amazonItems = fetch('amazonadvertising', 'item_search', hash(
         'keywords', 'harry potter',
         'search_index', 'Books',
     ))
     $firstProductID = $amazonItem.0.product_id
     $firstProductGenTimestamp = $amazonIteM.0.generated_timestamp}
```

And in PHP :

```
$result = eZFunctionHandler::execute('amazonadvertising', 'item_search', array(
  'keywords'          => 'harry potter',
  'search_index'      => 'Books'
));
if( $result )
{
  echo 'ProductID : ', $result[0]->product_id, ' / Generated Timestamp : ', $result[0]->generated_timestamp;
}
```