
Alexa Web Information Service

Developer Guide

API Version 2005-07-11

Alexa Web Information Service: Developer Guide

Published

Copyright © 2007 Amazon Services, Inc. or its Affiliates

AMAZON and AMAZON.COM are registered trademarks of Amazon.com, Inc. or its Affiliates. All other trademarks are the property of their respective owners.

Third Party Information: This guide contains links to third-party websites that are not under the control of Amazon.com, and Amazon.com is not responsible for the content of any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. Amazon.com provides these links at your own convenience, and the inclusion of the link does not imply that Amazon.com endorses or accepts any responsibility for the content on those third-party sites.

Table of Contents

Introduction	1
Making Requests	2
Query Requests	3
Batch Requests	4
Calculating Signatures	6
Authentication Errors	8
API Reference	9
Actions	9
UrlInfo	10
TrafficHistory	14
CategoryBrowse	16
CategoryListings	18
SitesLinkingIn	20
Common Request Parameters	22

Introduction

The Alexa Web Information Service (AWIS) offers a platform for creating innovative Web solutions and services based on Alexa's vast information about web sites, accessible with a web services API. The following documentation describes how to use the Alexa Web Information Service.

This documentation includes the following sections:

- [Making Requests \(p. 2\)](#)

The Making Requests Guide contains information about how to make requests to Alexa Web Information Service.

- [API Reference \(p. 9\)](#)

The API Reference contains details about the Alexa Web Information Service, including the Actions, Response Groups, and other elements that make up the application programming interface (API).

In order to use the Alexa Web Information Service, you must first sign up for an Amazon Web Services account, and then subscribe to the Alexa Web Information Service. See the following links to sign up:

- [Sign up for the Alexa Web Information Service](#)

Making Requests

This chapter describes how to authenticate and send requests to the Alexa Web Information Service.



Important

The SOAP API for the Alexa Web Information Service was deprecated on May 25, 2011. As of November 25, 2011 you must use the query API.

How to Make Requests to the Alexa Web Information Service

1. Sign up for the Alexa Web Information Service at <http://aws.amazon.com/awis>.
2. Get your access keys from the [AWS Access Identifiers](#) page of the Amazon Web Services portal.
3. Download one of the [code samples](#) from the Resource Center.
4. Follow the instructions in the README file to:
 - Paste your access keys into the code.
 - Execute the code sample.

Query Requests

A query request is simply an HTTP GET request that returns XML.

Sample Query Request

Here is a request that looks up traffic rank data for yahoo.com:

```
http://awis.amazonaws.com/?
  AWSAccessKeyId=9876543212345123
  &Action=UrlInfo
  &ResponseGroup=Rank
  &SignatureMethod=HmacSHA1
  &SignatureVersion=2
  &Timestamp=2011-05-06T17%3A58%3A49.463Z
  &Url=yahoo.com
  &Signature=Wz2UT%2BtCYZghLBmqtkfEpg%2FqrK8%3D
```

See [Common Request Parameters \(p. 22\)](#) and [Actions \(p. 9\)](#) for descriptions of query parameters.

String to Sign

The *Signature* parameter is used to authenticate the request. It is calculated as the hash of the request's "string to sign" as described in [Calculating Signatures \(p. 6\)](#). For the sample request above, the string to sign is:

```
GET\n
awis.amazonaws.com\n
/>\n
AWSAccessKeyId=9876543212345123&Action=UrlInfo&ResponseGroup=Rank
&SignatureMethod=HmacSHA1&SignatureVersion=2
&Timestamp=2011-05-06T17%3A58%3A49.463Z&Url=yahoo.com
```

where \n represents a newline. The last three lines are actually a single line, with no newlines.

Batch Requests

A batch request is similar to a regular query request, except that it allows multiple service calls, or sub-requests, to be sent in one request. Batch requests save on network latency time.



Note

Batch requests save time, not money -- they are billed as though each sub-request was made separately.

All parameters except the core AWS parameters (*Action*, *AWSSecretAccessKey*, *Signature*, *SignatureMethod*, *SignatureVersion*, and *Timestamp*) are batched according to the following rules:

- If a query request contains at least one parameter of the format `[ActionName].[integer].[ParameterName]`, it is a batched request (example: `UrlInfo.1.Url`).
- The numbers for the request params must be contiguous (i.e. if `UrlInfo.3.Url` is present, `UrlInfo.1.Url` and `UrlInfo.2.Url` are required as well).
- A batch request can have shared parameters (parameters that have the same value for all service calls). Shared parameters have the form `[ActionName].Shared.[ParameterName]` (example: `TrafficHistory.Shared.Start`). A short-hand form of `Shared.[ParameterName]` is supported as well.
- Up to 5 service calls can be batched in a single request. If more service calls are needed, they must be divided into multiple batch requests.
- All service calls within a single batch request must be of the same *Action* type. You cannot mix different actions in a single batch request.

Sample Batch Request #1

The following example shows a batch query request that returns the rank for three URLs.

```
http://awis.amazonaws.com?
  AWSSecretAccessKey=[your AWSSecretAccessKey]
  &Action=UrlInfo
  &Signature=[signature calculated from request]
  &SignatureMethod=[HmacSha1 or HmacSha256]
  &SignatureVersion=2
  &Timestamp=[time now, format described in Common Request
Parameters (p. 22)]
  &UrlInfo.Shared.ResponseGroup=Rank
  &UrlInfo.1.Url=[website #1]
  &UrlInfo.2.Url=[website #2]
  &UrlInfo.3.Url=[website #3]
```

Sample Batch Request #2

The following example shows a batch query request that returns five sites linking in for three URLs.

```
http://awis.amazonaws.com?
  AWSSecretAccessKey=[your AWSSecretAccessKey]
  &Action=SitesLinkingIn
  &Signature=[signature calculated from request]
  &SignatureMethod=[HmacSha1 or HmacSha256]
```

Alexa Web Information Service Developer Guide

Batch Requests

```
&SignatureVersion=2
&SitesLinkingIn.1.Url=[website #1]
&SitesLinkingIn.2.Url=[website #2]
&SitesLinkingIn.3.Url=[website #3]
&SitesLinkingIn.Shared.Count=5
&SitesLinkingIn.Shared.ResponseGroup=SitesLinkingIn
&Timestamp=[time now, format described in Common Request Parameters \(p. 22\)]
```


Calculating Signatures



Note

The best way to learn how to sign requests is to see the [code samples](#) in the Resource Center for examples using Java, C#, PHP, Perl and Ruby.



Important

Signature version 2 is described below. Previous signature versions were deprecated on May 25, 2011. As of November 25, 2011 you must use 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 all other characters with `%XY`, where X and Y are hex characters 0-9 and uppercase A-F.
 - 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).



Note

Currently all AWS 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 (/).

3. Calculate an RFC 2104-compliant HMAC with the string you just created, your Secret Access Key as the key, and SHA256 or SHA1 as the hash algorithm.
For more information, go to <http://www.ietf.org/rfc/rfc2104.txt>.
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://www.ietf.org/rfc/rfc3986.txt>). 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.

Authentication Errors

If your request cannot be authenticated, you will get an error message explaining the reason.

```
<Response>
  <Errors>
    <Error>
      <Code>AuthFailure</Code>
      <Message>AWS was not able to validate the provided access credentials</
Message>
    </Error>
  </Errors>
  <RequestID>6bbc7909-375e-4bd6-a37f-692ce855c3c0</RequestID>
</Response>
```

If you get an authentication error then one of the following is probably true:

- You are not signed up for the Alexa Web Information service. Even if you already have an Amazon Web Services account, you must separately sign up for the Alexa Web Information service at <http://aws.amazon.com/awis>
- Your credit card is invalid or expired.
- You have not URL encoded your Timestamp and Signature values. If you are seeing intermittent authorization errors, this is the most likely problem. You should not see any "+" signs in your Signature, for example.
- You are not generating the Signature correctly. Please see the [code samples](#).
- The clock on the computer making the requests is not set correctly.

API Reference

This section contains details about Alexa Web Information Service, including the Actions, Response Groups, and other elements that make up the application programming interface (API).

Each Action listed contains at least one sample request to help you get started. Use the sample requests as a starting point for developing your own requests. Keep in mind that you should substitute your own Access Key ID (*AWSAccessKeyId*) into the sample requests before using them.

Actions

- [UrlInfo \(p. 10\)](#) - get information about pages and sites on the web - their traffic, content, and related sites
- [TrafficHistory \(p. 14\)](#) - get a history of traffic rank
- [CategoryBrowse \(p. 16\)](#), [CategoryListings \(p. 18\)](#) - get lists of sites within a specific category ordered by traffic rank, or create a browseable directory of websites
- [SitesLinkingIn \(p. 20\)](#) - get a list of sites linking in to a specified site

UrlInfo

Description

The UrlInfo action provides information about a website, such as:

- how popular the site is
- what sites are related
- contact information for the owner of the site

To learn more about Alexa traffic data please see [this explanation](#) on the Alexa website

Request Parameters

The UrlInfo Action takes the following parameters. Required parameters must be provided for the request to succeed.

Name	Description	Required
<i>Action</i>	UrlInfo	Yes
<i>Url</i>	Any valid URL.	Yes
<i>ResponseGroup</i>	Any valid response group. See the Response Groups section below for valid options.	Yes

Response Groups

By specifying one or more response groups when making the request, you can retrieve only the information you are interested in.

Response Group	Data Returned
RelatedLinks	Up to 11 related links
Categories	Up to 3 DMOZ (Open Directory) categories the site belongs to
Rank	The Alexa three month average traffic rank
RankByCountry	Percentage of viewers, page views, and traffic rank broken out by country
RankByCity	Percentage of viewers, page views, and traffic rank broken out by city
UsageStats	Usage statistics such as reach and page views
ContactInfo	Contact information for the site owner or registrar
AdultContent	Whether the site is likely to contain adult content ('yes' or 'no')
Speed	Median load time and percent of known sites that are slower
Language	Content language code and character-encoding (note that this may not match the language or character encoding of any given page on the website because the language and character set returned are those of the <i>majority</i> of pages)

Response Group	Data Returned
Keywords	Keywords relevant to site content
OwnedDomains	Other domains owned by the same owner as this site
LinksInCount	A count of links pointing in to this site
SiteData	Title, description, and date the site was created

Meta-Response Groups

For convenience, the following meta-response groups are also available, which group together several similar response groups from the table above:

Response Group	Data Returned
Related	Up to 11 related links and up to 3 DMOZ categories (equivalent to ResponseGroup=RelatedLinks,Categories)
TrafficData	Traffic rank and usage statistics (equivalent to ResponseGroup=Rank,UsageStats)
ContentData	Information about the site's content (equivalent to ResponseGroup=SiteData,AdultContent,Popups,Speed,Language)

Sample Request

The following example shows a query-style request and response.

Request

```
http://awis.amazonaws.com/?
  Action=UrlInfo
  &AWSAccessKeyId=[Your AWS Access Key ID]
  &Signature=[signature calculated from request]
  &SignatureMethod=[HmacSha1 or HmacSha256]
  &SignatureVersion=2
  &Timestamp=[timestamp used in signature]
  &Url=[Valid URL]
  &ResponseGroup=[Valid Response Group]
```

Response

```
<aws:UrlInfoResponse xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
<aws:Response xmlns:aws="http://awis.amazonaws.com/doc/2005-07-11">
<aws:OperationRequest>
<aws:RequestId>5486794a-0d03-4d47-a45b-e95764c3f0ee</aws:RequestId>
/aws:OperationRequest>
<aws:UrlInfoResult>
<aws:Alexa>

  <aws:ContentData>
    <aws:DataUrl type="canonical">yahoo.com</aws:DataUrl>
```

```
<aws:Asin>B00006D2TC</aws:Asin>
<aws:SiteData>
  <aws:Title>Yahoo!</aws:Title>
  <aws:Description>Personalized content and search options. Chatrooms,
free e-mail, clubs, and pager.</aws:Description>
  <aws:OnlineSince>18-Jan-1995</aws:OnlineSince>
</aws:SiteData>
<aws:Speed>
  <aws:MedianLoadTime>2242</aws:MedianLoadTime>
  <aws:Percentile>51</aws:Percentile>
</aws:Speed>
<aws:AdultContent>no</aws:AdultContent>
<aws:Language>
  <aws:Locale>en</aws:Locale>
</aws:Language>
<aws:LinksInCount>76894</aws:LinksInCount>
<aws:Keywords>
  <aws:Keyword>On the Web</aws:Keyword>
</aws:Keywords>
<aws:OwnedDomains>
  <aws:OwnedDomain>
    <aws:Domain>yahooligans.com</aws:Domain>
    <aws:Title>yahooligans.com</aws:Title>
  </aws:OwnedDomain>
</aws:OwnedDomains>
</aws:ContentData>

<aws:Related>
  <aws:DataUrl type="canonical">yahoo.com/</aws:DataUrl>
  <aws:Asin>B00006D2TC</aws:Asin>
  <aws:RelatedLinks>
    <aws:RelatedLink>
      <aws:DataUrl type="canonical">aol.com/</aws:DataUrl>
      <aws:NavigableUrl>http://aol.com/</aws:NavigableUrl>
      <aws:Asin>B00006ARD3</aws:Asin>
      <aws:Relevance>301</aws:Relevance>
    </aws:RelatedLink>
  </aws:RelatedLinks>
  <aws:Categories>
    <aws:CategoryData>
      <aws:Title>On the Web/Web Portals</aws:Title>
      <aws:AbsolutePath>Top/Computers/Internet/On_the_Web/Web_Portals</
aws:AbsolutePath>
    </aws:CategoryData>
  </aws:Categories>
</aws:Related>

<aws:TrafficData>
  <aws:DataUrl type="canonical">yahoo.com/</aws:DataUrl>
  <aws:Asin>B00006D2TC</aws:Asin>
  <aws:Rank>1</aws:Rank>
  <aws:UsageStatistics>

  <aws:UsageStatistic>
    <aws:TimeRange>
      <aws:Days>1</aws:Days>
    </aws:TimeRange>
    <aws:Rank>
      <aws:Value>1</aws:Value>
```

```
<aws:Delta>+0</aws:Delta>
</aws:Rank>
<aws:Reach>
  <aws:Rank>
    <aws:Value>2</aws:Value>
    <aws:Delta>+0</aws:Delta>
  </aws:Rank>
  <aws:PerMillion>
    <aws:Value>252,500</aws:Value>
    <aws:Delta>-1%</aws:Delta>
  </aws:PerMillion>
</aws:Reach>
<aws:PageViews>
  <aws:PerMillion>
    <aws:Value>51,400</aws:Value>
    <aws:Delta>-1%</aws:Delta>
  </aws:PerMillion>
  <aws:Rank>
    <aws:Value>1</aws:Value>
    <aws:Delta>+0</aws:Delta>
  </aws:Rank>
  <aws:PerUser>
    <aws:Value>13.7</aws:Value>
    <aws:Delta>-1%</aws:Delta>
  </aws:PerUser>
</aws:PageViews>
</aws:UsageStatistic>

</aws:UsageStatistics>
</aws:TrafficData>

</aws:Alexa>
</aws:UrlInfoResult>
<aws:ResponseStatus xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
<aws:StatusCode>Success</aws:StatusCode>
</aws:ResponseStatus>
</aws:Response>
</aws:UrlInfoResponse>
```


TrafficHistory

Description

The TrafficHistory action returns the daily Alexa Traffic Rank, Reach per Million Users, and Unique Page Views per Million Users for each day since August 2007. This same data is used to produce the traffic graphs found on alexa.com.

Request Parameters

The TrafficHistory Action takes the following parameters. Required parameters must be provided for the request to succeed.

Name	Description	Required
<i>Action</i>	TrafficHistory	Yes
<i>Url</i>	Any valid URL.	Yes
<i>ResponseGroup</i>	History is the only available response group.	Yes
<i>Range</i>	Number of days to return. Note that the response document may contain fewer results than this maximum if data is not available. Default value is '31'. Maximum value is '31'.	No
<i>Start</i>	Start date for results. The first start available date is 20070801 (August 1, 2007).	No

Example

The following example shows a query-style request and response

Request

```
http://awis.amazonaws.com/?
  Action=TrafficHistory
  &AWSAccessKeyId=[Your AWS Access Key ID]
  &Signature=[signature calculated from request]
  &SignatureMethod=[HmacSha1 or HmacSha256]
  &SignatureVersion=2
  &Timestamp=[timestamp used in signature]
  &Url=[Valid URL]
  &ResponseGroup=History
  &Range=[maximum number of results]
  &Start=[start date for results]
```

Response

The response contains an `aws:Data` element for each day in the date range specified. Note that no data will be returned for days when the daily Alexa traffic rank was greater than 100,000.

```
<aws:TrafficHistoryResponse xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
```

```
<aws:Response xmlns:aws="http://awis.amazonaws.com/doc/2005-07-11">
<aws:OperationRequest><aws:RequestId>39ab4736-e1a-492d-924b-d0d768d2692d</
aws:RequestId>
</aws:OperationRequest>
<aws:TrafficHistoryResult>
<aws:Alexa>
<aws:TrafficHistory>
  <aws:Range>31</aws:Range>
  <aws:Site>amazon.com</aws:Site>
  <aws:Start>2005-01-01</aws:Start>
  <aws:HistoricalData>
    <aws:Data>
      <aws:Date>2005-01-01</aws:Date>
      <aws:PageViews>
        <aws:PerMillion>2801</aws:PerMillion>
        <aws:PerUser>5.0</aws:PerUser>
      </aws:PageViews>
      <aws:Rank>18</aws:Rank>
      <aws:Reach>
        <aws:PerMillion>26041</aws:PerMillion>
      </aws:Reach>
    </aws:Data>
  </aws:HistoricalData>
</aws:TrafficHistory>
</aws:Alexa>
</aws:TrafficHistoryResult>
<aws:ResponseStatus xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
<aws:StatusCode>Success</aws:StatusCode>
</aws:ResponseStatus></aws:Response>
</aws:TrafficHistoryResponse>
```

CategoryBrowse

Description

The *CategoryBrowse* action and *CategoryListings* actions together provide a directory service based on the Open Directory, www.dmoz.org, and enhanced with Alexa traffic data.

For any given category, the *CategoryBrowse* action returns a list of sub-categories. Within a particular category you can use the [CategoryListings \(p. 18\)](#) action to get the documents within that category ordered by traffic.

Request Parameters

The CategoryBrowse Action takes the following parameters. Required parameters must be provided for the request to succeed.

Name	Description	Required
<i>Action</i>	CategoryBrowse	Yes
<i>ResponseGroup</i>	Any valid response group. See the Response Group section below for valid options.	Yes
<i>Path</i>	Valid category path (Top/Arts, Top/Business/Automotive)	Yes
<i>Descriptions</i>	Whether to return descriptions with categories: (True False)	No

Response Groups

By specifying one or more response groups when making the request, you can retrieve only the information you are interested in.

Response Group	Data Returned
Categories	All sub-categories within the specified category path
RelatedCategories	Categories that are related to the specified category path
LanguageCategories	Language categories in which the specified category path is available
LetterBars	"Letter Bars" (A, B, C, etc.) for categories that contain them

Sample Request

The following example shows a query-style request and response

Request

```
http://awis.amazonaws.com/?
  Action=CategoryBrowse
  &AWSAccessKeyId=[Your AWS Access Key ID]
  &Signature=[signature calculated from request]
  &SignatureMethod=[HmacSha1 or HmacSha256]
  &SignatureVersion=2
  &Timestamp=[timestamp used in signature]
```

```
&ResponseGroup=[Valid Response Group]
&Path=[Top/Arts, Top/Business/Automotive]
&Descriptions=[True | False]
```

Response

```
<aws:CategoryBrowseResponse xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
  <aws:Response xmlns:aws="http://awis.amazonaws.com/doc/2005-07-11">
    <aws:OperationRequest>
      <aws:RequestId>cadde770-fe83-483c-9bcc-465a77d4ea0c</aws:RequestId>
    </aws:OperationRequest>
    <aws:CategoryBrowseResult>
      <aws:Alexa>

        <aws:CategoryBrowse>

          <aws:Categories>
            <aws:Category>
              <aws:Path>Top/Business/Consumer_Goods_and_Services/Electronics/
Accessories</aws:Path>
              <aws:Title>Accessories</aws:Title>
              <aws:SubCategoryCount>2</aws:SubCategoryCount>
              <aws:TotalListingCount>186</aws:TotalListingCount>
            </aws:Category>
            <aws:Category>
              <aws:Path>Top/Business/Consumer_Goods_and_Services/Electronics/
Audio</aws:Path>
              <aws:Title>Audio</aws:Title>
              <aws:SubCategoryCount>8</aws:SubCategoryCount>
              <aws:TotalListingCount>1135</aws:TotalListingCount>
            </aws:Category>
          </aws:Categories>

          <aws:LetterBars/>

        </aws:CategoryBrowse>
      </aws:Alexa>
    </aws:CategoryBrowseResult>
    <aws:ResponseStatus xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
      <aws:StatusCode>Success</aws:StatusCode>
    </aws:ResponseStatus>
  </aws:Response>
</aws:CategoryBrowseResponse>
```

CategoryListings

Description

The CategoryListings action is a directory service based on the Open Directory, www.dmoz.org. For any given category, it returns a list of site listings contained within that category.

Request Parameters

The CategoryListings Action takes the following parameters. Required parameters must be provided for the request to succeed.

Name	Description	Required
<i>Action</i>	CategoryListings	Yes
<i>ResponseGroup</i>	Listings is the only available response group.	Yes
<i>Path</i>	Valid category path. Note that top-level categories will not return any listings unless Recursive=yes is specified (see below). Example values are Top/Arts , Top/Business/Automotive Note that the value should be URL-encoded. For example, Path=Top%2FBusiness%2FConsumer_Goods_and_Services%2FElectronics	Yes
<i>SortBy</i>	How to sort the results returned by this service: (Popularity Title AverageReview)	No
<i>Recursive</i>	Whether to return listings for the current category only, or for the current category plus all subcategories: (True False)	No
<i>Start</i>	1-based index of result at which to start. Note: An empty document will be returned if this value exceeds the total number of available results.	No
<i>Count</i>	Number of results to return for this request, beginning from specified Start number (maximum 20)	No
<i>Descriptions</i>	Whether to return descriptions with categories: (True False)	No

Sample Request

The following example shows a query-style request and response

Request

```
http://awis.amazonaws.com/?
  Action=CategoryListings
  &AWSAccessKeyId=[Your AWS Access Key ID]
  &Signature=[signature calculated from request]
  &SignatureMethod=[HmacSha1 or HmacSha256]
  &SignatureVersion=2
  &Timestamp=[timestamp used in signature]
  &ResponseGroup=Listings
```

```
&Path=[Top/Arts, Top/Business/Automotive]
&SortBy=[one of: ( Popularity | Title | AverageReview )]
&Recursive=[True | False]
&Start=[number to start at]
&Count=[Number of results to return]
&Descriptions=[True | False]
```

Response

```
<aws:CategoryListingsResponse xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
  <aws:Response xmlns:aws="http://awis.amazonaws.com/doc/2005-07-11">
    <aws:OperationRequest>
      <aws:RequestId>0bf0a4b0-a441-49e7-9cfe-44b71e0df086</aws:RequestId>
    </aws:OperationRequest>
    <aws:CategoryListingsResult>
      <aws:Alexa>

        <aws:CategoryListings>
          <aws:RecursiveCount>1804</aws:RecursiveCount>
          <aws:Count>217</aws:Count>
          <aws>Listings>
            <aws:Listing>
              <aws:DataUrl type="navigable">http://www.sony.com</aws:DataUrl>
              <aws>Title>Sony Electronics</aws>Title>
              <aws:PopularityRank>882</aws:PopularityRank>
            </aws:Listing>
            <aws:Listing>
              <aws:DataUrl type="navigable">http://www.samsung.com/</aws:DataUrl>
              <aws>Title>Samsung Electronics</aws>Title>
              <aws:PopularityRank>899</aws:PopularityRank>
            </aws:Listing>
          </aws>Listings>
        </aws:CategoryListings>

      </aws:Alexa>
    </aws:CategoryListingsResult>
    <aws:ResponseStatus xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
      <aws:StatusCode>Success</aws:StatusCode>
    </aws:ResponseStatus>
  </aws:Response>
</aws:CategoryListingsResponse>
```

SitesLinkingIn

Description

The SitesLinkingIn action returns a list of web sites linking to a given web site. Within each domain linking into the web site, only a single link - the one with the highest page-level traffic - is returned. The data is updated once every two months.

Request Parameters

The SitesLinkingIn Action takes the following parameters. Required parameters must be provided for the request to succeed.

Name	Description	Required
<i>Action</i>	SitesLinkingIn	Yes
<i>Url</i>	Any valid URL.	Yes
<i>ResponseGroup</i>	SitesLinkingIn is the only available response group.	Yes
<i>Count</i>	Maximum number of results per page to return. Note that the response document may contain fewer results than this maximum. Default value is '10' (maximum 20).	No
<i>Start</i>	Number of result at which to start. Used for paging through results. Default value is '0.'	No

Sample Request

The following example shows a query-style request and response

Request

```
http://awis.amazonaws.com/?
  Action=SitesLinkingIn
  &AWSAccessKeyId=[Your AWS Access Key ID]
  &Signature=[signature calculated from request]
  &SignatureMethod=[HmacSha1 or HmacSha256]
  &SignatureVersion=2
  &Timestamp=[timestamp used in signature]
  &Url=[Valid URL]
  &ResponseGroup=SitesLinkingIn
  &Count=[maximum number of results]
  &Start=[number to start]
```

Response

One `aws:Site` element is returned for each site linking in. The title and url are for the page on that site with the highest page-level traffic.


```
<aws:SitesLinkingInResponse xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
```

Alexa Web Information Service Developer Guide
SitesLinkingIn

```
<aws:Response xmlns:aws="http://awis.amazonaws.com/doc/2005-07-11">
  <aws:OperationRequest>
    <aws:RequestId>ca282ec6-2d08-4341-9f1d-50f8c1e3652b
    </aws:RequestId>
  </aws:OperationRequest>
<aws:SitesLinkingInResult>
<aws:Alexa>
  <aws:SitesLinkingIn>
    <aws:Site>
      <aws:Title>Google</aws:Title>
      <aws:Url>http://www.google.com:80/Top/Computers/Internet/On_the_Web/
Web_Portals/</aws:Url>
    </aws:Site>
    <aws:Site>
      <aws:Title>www.fotolog.com:80/TsR_BkR_TsR</aws:Title>
      <aws:Url>http://www.fotolog.com:80/TsR_BkR_TsR</aws:Url>
    </aws:Site>
  </aws:SitesLinkingIn>
</aws:Alexa>
</aws:SitesLinkingInResult>
<aws:ResponseStatus xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
  <aws:StatusCode>Success</aws:StatusCode>
</aws:ResponseStatus>
</aws:Response>
</aws:SitesLinkingInResponse>
```


Common Request Parameters

Each action in the Alexa Web Information Service API has its own specific set of parameters, but there is also a set of parameters that all actions use. This section describes those input parameters.

Parameter	Description	Required
<i>Action</i>	The API operation, for example, <i>UrlInfo</i> or <i>TrafficHistory</i> Type: String Constraint: Must be a valid action	Yes
<i>AWSAccessKeyId</i>	A string, distributed by Amazon when you sign up to be a developer, that uniquely identifies the caller Type: String	Yes
<i>Signature</i>	A value calculated using the request parameters and a SHA256 (preferred) or SHA1 HMAC encryption algorithm Type: String	Yes
<i>SignatureMethod</i>	A value that specifies the signing method Type: String Valid Values: HmacSHA256 (preferred) or HmacSHA1.	Yes
<i>SignatureVersion</i>	A value that specifies the <i>Signature</i> format Type: Integer Valid Value: 2  Important Previous signature versions were deprecated on May 25, 2011. As of November 25, 2011 you must use signature version 2. See Calculating Signatures (p. 6) .	Yes
<i>Timestamp</i>	An ISO 8601 date-time in UTC that marks the day and time the request was sent. Requests expire after a certain length of time to prevent malicious users from capturing requests and resubmitting them at a later time. Type: DateTime, for example, 2008-09-18T13:00:01Z	Yes
<i>Version</i>	The API version to use. Use the current API version number, 2005-07-11, to ensure that requests succeed even if the API changes in future versions. Type: Date Valid Value: 2005-07-11	No