
Alexa Top Sites

Developer Guide

API Version 2005-11-21

Alexa Top Sites: Developer Guide

Published

Copyright © 2005 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	5
Authentication Errors	7
API Reference	8
Actions	8
TopSites	9
Common Request Parameters	12

Introduction

Alexa Top Sites is a web service that provides lists of web sites, ordered by Alexa Traffic Rank. Using this web service, developers can page through lists of top sites and incorporate traffic data into their applications. The following documentation describes how to use Alexa Top Sites.

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

The Making Requests Guide contains information about how to make requests to Alexa Top Sites.

- [API Reference \(p. 8\)](#)

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

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

- Sign up for an Amazon Web Services account at <http://aws.amazon.com>.
- Sign up for Alexa Top Sites at <http://aws.amazon.com/alexatopsites>.

Making Requests

This chapter describes how to authenticate and send requests to Alexa Top Sites.



Important

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

How to Make Requests to Alexa Top Sites

1. Sign up for Alexa Top Sites at <http://aws.amazon.com/alexatopsites>.
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 for sites ranked 301 - 400 in Brazil:

```
http://ats.amazonaws.com/?
    AWSAccessKeyId=9876543212345123
    &Action=TopSites
    &Count=100
    &CountryCode=BR
    &ResponseGroup=Country
    &SignatureMethod=HmacSHA1
    &SignatureVersion=2
    &Start=301
    &Timestamp=2011-05-06T17%3A58%3A49.463Z
    &Url=yahoo.com
    &Signature=Wz2UT%2BtCYZghLBmqtkfEpg%2FqrK8%3D
```

See [Common Request Parameters \(p. 12\)](#) and [TopSites \(p. 9\)](#) 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 \(p. 5\)](#). For the sample request above, the string to sign is:

```
GET\n
ats.amazonaws.com\n
/>\n
AWSAccessKeyId=9876543212345123&Action=TopSites&Count=100&CountryCode=br
&ResponseGroup=Country&SignatureMethod=HmacSHA256&SignatureVersion=2
&Start=1001&Timestamp=2011-05-12T16%3A50%3A15.065Z
```

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 (*AWSAccessKeyId*, *Timestamp*, *Signature*, *SignatureMethod*, *SignatureVersion*, and *Action*) 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: `TopSites.1.CountryCode`).
- The numbers for the request params must be contiguous (i.e. if `TopSites.3.CountryCode` is present, `TopSites.1.CountryCode` and `TopSites.2.CountryCode` are required as well).
- To save on typing, batched requests are allowed to have shared parameters. These are the parameters that have the same value for all the batch sub-requests. Shared parameters have the form `[ActionName].Shared.[ParameterName]` (example: `TopSites.Shared.ResponseGroup`). Short-hand form of `Shared.[ParameterName]` is supported as well.
- Up to 5 service calls can be batched at a time 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

```
http://ats.amazonaws.com/?
  AWSAccessKeyId=[your AWSAccessKeyId]
  &Action=TopSites
  &Signature=[signature calculated from request]
  &SignatureMethod=[HmacSha1 or HmacSha256]
  &SignatureVersion=2
  &Timestamp=[time now, format described in Common Request
Parameters (p. 12)]
  &TopSites.Shared.ResponseGroup=[Country]
  &TopSites.1.CountryCode=[query #1]
  &TopSites.2.CountryCode=[query #2]
  &TopSites.3.CountryCode=[query #3]
```

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, please check whether:

- You are not signed up for Alexa Top Sites. Even if you already have an Amazon Web Services account, you must separately sign up for Alexa Top Sites at <http://aws.amazon.com/alexatopsites>
- 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 using the Alexa Top Sites API.

Each action listed contains a 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

- [TopSites \(p. 9\)](#) provides lists of web sites ordered by Alexa Traffic Rank.

TopSites

Description

The TopSites action provides lists of web sites ordered by Alexa Traffic Rank. A global list is available, as well as lists for individual countries.

Request Parameters

In addition to the [common request parameters](#) (p. 12), the TopSites action takes the following parameters. Required parameters must be provided for the request to succeed.

Name	Description	Type	Value
<i>CountryCode</i>	Valid country code (a list of country codes is available from ListCountries ResponseGroup). A global list of top sites is returned by default.	Optional	
<i>CityCode</i>	Valid city code (a list of city codes is available from ListCities ResponseGroup). A global list of top sites is returned by default.	Optional	
<i>Start</i>	Number of result at which to start. Used for paging through results. Default value is '1'.	Optional	Number to start at
<i>Count</i>	Number of results (maximum) per page to return. Default value is '100'. Maximum is '100'.	Optional	Number from 1 to 100
<i>ResponseGroup</i>	Any valid response group. See the Response Group section for valid options.	Required	Comma-separated list of response groups.

Response Groups

Response groups allow the user more control over what data is returned. 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
Country	Top sites for a given country
City	Top sites for a given city
ListCountries	Countries that have top sites
ListCities	Cities that have top sites

Sample Request

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

Request

```
http://ats.amazonaws.com/?
  AWSAccessKeyId=1234567890ABCDEFGHIJ
  &Action=TopSites
  &Count=1
  &CountryCode=CN
  &ResponseGroup=Country
  &Signature=1234567890abcdefghijABCDEFGH
  &SignatureMethod=[HmacSHA256 or HmacSHA1]
  &SignatureVersion=2
  &Timestamp=2006-01-01T00:00:00.000Z
```

For more information on signing requests, see [Calculating Signatures \(p. 5\)](#).

Response


For a description of rank, reach, and page views, see <http://www.alexametrics.com/help/traffic-learn-more>.

```
<?xml version="1.0"?>
<aws:TopSitesResponse xmlns:aws="http://alexametrics.com/doc/2005-10-05/">
  <aws:Response>
    <aws:OperationRequest>
      <aws:RequestId>9ffc5e13-175e-4c7e-b33b-0efe3501d1f3</aws:RequestId>
    </aws:OperationRequest>
    <aws:TopSitesResult>
      <aws:Alexa>
        <aws:TopSites>
          <aws:List>
            <aws:CountryName>China</aws:CountryName>
            <aws:CountryCode>CN</aws:CountryCode>
            <aws:TotalSites>671496</aws:TotalSites>
            <aws:Sites>
              <aws:Site>
                <aws:DataUrl>baidu.com</aws:DataUrl>
                <aws:Country>
                  <aws:Rank>1</aws:Rank>
                  <aws:Reach>
                    <aws:PerMillion>358000</aws:PerMillion>
                  </aws:Reach>
                  <aws:PageViews>
                    <aws:PerMillion>77410</aws:PerMillion>
                    <aws:PerUser>11.5</aws:PerUser>
                  </aws:PageViews>
                </aws:Country>
                <aws:Global>
                  <aws:Rank>4</aws:Rank>
                </aws:Global>
              </aws:Site>
            </aws:Sites>
          </aws:List>
        </aws:TopSites>
      </aws:Alexa>
    </aws:TopSitesResult>
  <aws:ResponseStatus>
```

```
<aws:StatusCode>Success</aws:StatusCode>  
</aws:ResponseStatus>  
</aws:Response>  
</aws:TopSitesResponse>
```

Common Request Parameters

Each action in the Alexa Top Sites 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 Type: String Valid Value: TopSites	Yes
<i>AWSAccessKeyId</i>	A string, distributed by Amazon FPS 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>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 as described in Calculating Signatures (p. 5) .	Yes
<i>SignatureMethod</i>	The hashing algorithm method used to create the signature. Type: String Valid Values: HmacSHA256 (preferred) and HmacSHA1.	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, 2005-11-21, to ensure that requests succeed even if the API changes in future versions. Type: Date Valid Value: 2005-11-21	No