

# Alexa Web Information Service

## 开发者指南

( API 版本 2005-07-11 )

## 译者序

Alexa Web Information Service（简称 AWIS）是 Alexa 提供的一个基于 Web 的 API 服务。通过此 API，你可以查询指定网站或网址的 Alexa 流量信息等内容。

由于工作上的需要，最近接触了 Alexa Web Information Service（AWIS），发现开发文档没有中文版，内容不多，我就顺手翻译了。

Alexa和Amazon对本文的原文保有版权（如果有的话）。本人对译文保有版权。如需转载请与本人联系（联系方式见此页：<http://www.naks.cn/blueven/>）。

本人才疏学浅，翻译中难免有错，欢迎大家指正。如果你对 AWIS 有问题的话，可以到<http://www.naks.cn/blueven>提出（这个页面同时是在线版《Alexa Web Information Service开发指南》的所在地），我将尽力回答。

感谢我的老板牛先生，他让我有机会接触 AWIS，并且借我 AWIS 的账号用。

最后为我的博客打个广告：布鲁文的蓝色奇想 <http://www.naks.cn/blueven/>

——by Ven

2008-4-15

# 介绍

Alexa Web Information Service (AWIS) 提供基于 Alexa 海量网站信息的创新性 Web 解决方案与服务，通过 web service API 进行访问。以下文档描述了如何使用 Alexa Web Information Service。

本文档包含以下部分：

- 发送请求  
本部分包含了如何向 Alexa Web Information Service 发送请求的信息。
- API 参考  
API 参考是 Alexa Web Information Service 的详细内容。如动作 (Action)，反馈组 (Response Group)，及其他组成该 API 的元素。

要想使用 Alexa Web Information Service，你必须先注册一个 Amazon Web Services 账号，然后订购 Alexa Web Information Service。以下是注册链接：

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

# 发送请求

本章介绍如何向 Alexa Web Search 服务进行认证和发送请求。

## 如何向 Alexa Web Information Service 发送请求

步骤如下：

- 1: 在<http://aws.amazon.com/awis>注册Alexa Web Information Service。
- 2: 在 AWS Access Identifiers 页面获取你的 Amazon Web Services 访问密钥。
- 3: 从资源中心下载一个示例代码。
- 4: 按照 README 文件的说明：
  - 把你的访问密钥贴入代码中。
  - 执行示例代码。

## 请求类型

本服务支持两种请求类型：

- Query 请求  
Query 请求是返回 XML 的 HTTP 请求。
- SOAP 请求  
SOAP 请求把参数传递进一个 SOAP envelope。

## Query 请求

Query 请求是返回 XML 的简单 HTTP GET 请求。

### Query 请求的示例

这是一个查询 yahoo.com 的流量等级数据的请求示例：

```
http://awis.amazonaws.com/?
  AWSAccessKeyId=9876543212345123
  &Timestamp=2007-01-26T01%3A15%3A38.000Z
  &Signature=oQkiPZUtQ9PITi2l4OTRA8fjYsM%3D
  &Version=2005-07-11
  &Action=UrlInfo
  &ResponseGroup=Rank
  &Url=yahoo.com
```

AWSAccessKeyId, Timestamp, SignatureVersion 和 Signature 参数用来认证和授权你的请求。Timestamp 和 Signature 的值在每次查询时必须重新生成。

Action, ResponseGroup 和 Url 参数用来控制返回什么样的信息。

### Query 请求的认证

AWS 使用请求认证来确认一个请求是由一个注册且进行过支付验证的开发者发出的。

#### 访问密钥识别码 (Access Key Identifiers)

创建一个 AWS 帐户时需要指定访问密钥 ID (Access Key ID) 和一个访问密钥 (Secret Access Key)。访问密钥 ID 通过 AWSAccessKeyId 参数传递给请求，用以验证该请求与对应账号的关联。为防止伪造，请求的发送者还需要用访问密钥进行 hash 计算，来为访问进行标记。Hash 码传递给 Signature 参数。永远不要把访问密钥告诉任何人。

#### 认证参数

以下参数用于对请求进行认证：

| 参数             | 描述   |
|----------------|--|
| AWSAccessKeyId | 你的 AWS 访问密钥 Id。  |
| Timestamp      | 用 UTC 格式表示的当前时间。如果时间戳与 Amazon 服务器上的时间相差超过 15 分钟，认证失败。Timestamp 必须是 URL 编码过的。 |
| Signature      | 查询签名通过将 Action 和 Timestamp 连起来，用访问密钥进行 RFC 2104-compliant HMAC-SHA1 哈希计算。    |

## 生成请求签名

注意：学习签署请求最快的方法是去资源中心查看示例代码。

如果资源中心没有你所用的编程语言的示例代码，进行以下步骤来生成 Timestamp 和 Signature：

- 1：Timestamp 值用 UTC 格式表示：yyyy-MM-ddTHH:mm:ss.fffZ。更多信息参看：<http://www.w3.org/TR/xmlschema-2/#dateTime>。
- 2：链接 Action 名称和 Timestamp 的值。
- 3：对 Action 加 Timestamp 字符串进行 RFC2104 兼容的 HMAC-SHA1 哈希计算，用访问密钥作为哈希计算的密钥。更多信息参见：<http://www.ietf.org/rfc/rfc2104.txt>
- 4：对哈希码进行 base64 必爱你妈以获得签名。
- 5：对签名进行 URL 编码。注意：签名中不能包含任何+号。比如在 C# 中，你必须用 `HttpUtility.UrlPathEncode` 来替代 `HttpUtility.UrlEncode`。
- 6：将计算出的签名通过 Signature 参数传递给你的请求。
- 7：对 Timestamp 进行 URL 编码，并传递给你的请求。

## 认证错误

如果你的请求无法通过认证，你将得到错误信息和错误原因。

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

如果认证失败，可能是以下原因：

- 1：你没有注册 Alexa Web Information service。即使你已经有了 Amazon Web Service 账号，仍需另外注册 Alexa Web Information service。
- 2：你的信用卡无效或过期。
- 3：你没有对 Signature 和 Timestamp 进行 URL 编码。如果你遇到的是间歇性的认证失败，那这就是最有可能的原因。比如在 URL 里出现了+号。
- 4：你没有正确的生成 Signature。
- 5：发送请求的电脑上的时间没有正确设置。

## SOAP 请求

你的 SOAP 工具包负责基于提供的 WSDL 来格式化和发送请求。本部分展示了什么是 SOAP 请求。在 SOAP 中，认证信息是放在 SOAP 请求中的如下元素里：

每个请求都必须包含认证信息，用以对请求进行确认。在 SOAP 中，认证信息放在<Request>元素中。需要的参数见下表：

| 参数             | 描述   |
|----------------|--|
| AWSAccessKeyId | 你的 AWS 访问密钥。   |
| Timestamp      | 必须是一个协调世界时（Coordinated Universal Time），例如 2005-01-31T23:59:59.183Z。如果时间戳与 Amazon 服务器上的时间相差超过 15 分钟，认证失败。Timestamp 必须是 URL 编码过的。  |
| Signature      | <p>连接[Action（例如“UrlInfo”）]+[Timestamp]，用 AWS 访问密钥作为算法密钥的 RFC 2104 HMAC-SHA1 摘要。参见“Query 请求”来了解如何生成签名。例如，见以下示例，签名元素必须包含“UrlInfo2005-01-31T23:59:59.183Z”的 HMAC-SHA1 摘要的值。</p> <p>SOAP 请求示例：</p> <pre>&lt;SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOA   &lt;SOAP-ENV:Body&gt;     &lt;m:UrlInfo xmlns:m="http://webservices.amazon.com/AWSAlexa/2005-07-11"&gt;       &lt;m:Request&gt;         &lt;m:Security&gt;            &lt;m:AWSAccessKeyId&gt;1234567890ABCDEFGHJI&lt;/m:AWSAc            &lt;m:Signature&gt;S1USmkAFnpQy6xCgKA Ea9RXITVA=&lt;/m:Si            &lt;m:Timestamp&gt;2006-09-08T22:02:46.000Z&lt;/m:Timest             &lt;/m:Security&gt;             &lt;m:Url&gt;amazon.com&lt;/m:Url&gt;             &lt;m:ResponseGroup&gt;Rank&lt;/m:ResponseGroup&gt;           &lt;/m:Request&gt;         &lt;/m:UrlInfo&gt;       &lt;/SOAP-ENV:Body&gt;     &lt;/SOAP-ENV:Envelope&gt;</pre> <p><b>URL 编码。</b> SHA-1 的哈希结果是二进制数据。需要在 SOAP 请求里指定一个编码。SOAP 请求必须是 Base64 编码的。</p> |

注意：由于对时间精度的取舍不同，.NET 用户必须注意，不要发送过于精确的时间戳。解决方法是手动创建精确到毫秒的日期-时间对象。

## SOAP 反馈示例

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <m:UrlInfoResponse xmlns:m="http://webservices.amazon.com/AWSAlexa/2005-07-11">
      <aws:Response xmlns:aws="http://alex.amazonaws.com/doc/2005-10-05/">
        <aws:OperationRequest>
          <aws:RequestId>3e894c2b-ee67-4de3-90f3-e9eeab516479</aws:RequestId>
        </aws:OperationRequest>
        <aws:UrlInfoResult>
          <aws:Alexa>
            <aws:TrafficData>
              <aws:DataUrl type="canonical">amazon.com/</aws:DataUrl>
              <aws:Asin>B0000690L4</aws:Asin>
              <aws:Rank>17</aws:Rank>
            </aws:TrafficData>
          </aws:Alexa>
        </aws:UrlInfoResult>
        <m:ResponseStatus>
          <m:StatusCode>Success</m:StatusCode>
        </m:ResponseStatus>
      </aws:Response>
    </m:UrlInfoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



## Batch 请求

Batch 请求与常规的 Query 请求类似，只是 Batch 请求允许在一个请求中调用多个服务。相对于发送多个独立的服务器请求，Batch 请求节省 HTTP 回转次数。除了 AWS 核心参数（AWSAccessKeyid, Timestamp, Signature, Action）以外的参数需按照以下规则来进行 batch：

- 如果查询请求包含至少一个[ActionName].[integer].[ParameterName]形式的参数，则它是一个 batch 请求。（例如：UrlInfo.1.Url）。
- 请求参数必须按组完整呈现（例如有 UrlInfo.3.Url，则 UrlInfo.2.Url 与 UrlInfo.1.Url 也必须有）。
- 为了方便键入，batch 请求允许有共享参数。共享参数是所有 batch 子查询中有相同的值的参数。共享参数必须是这种形式：[ActionName].Shared.[ParameterName]（例如：TrafficHistory.Shared.Start）。同时支持速记形式的.[ParameterName]。
- 单个请求最多调用 5 个服务。如果需要调用更多服务，则必须发送多个 batch 请求。
- 单个 batch 请求中的所有服务调用必须具有相同的 Action 类型。你不能在一个 batch 请求中混合不同的动作。

**Batch 请求示例 #1。** 以下是发送三个 URL，每个 URL 返回三个 UrlInfo 等级的 batch 请求。

```
http://awis.amazonaws.com?
  AWSAccessKeyId=
  [your AWSAccessKeyId]
  &Timestamp=
  [timestamp now, format as described in http://www.w3.org/TR/xmlschema-2/#dat
  &Signature=
  [signature calculated from request]
  &Action=
  UrlInfo
  &UrlInfo.Shared.ResponseGroup=
  Rank
  &UrlInfo.1.Url=
  [website #1]
  &UrlInfo.2.Url=
  [website #2]
  &UrlInfo.3.Url=
  [website #3]
```

**Batch 请求示例 #2。** 以下示例展示发送三个 URL,每个 URL 返回 5 条反向链接的的 batch 请求。

```
http://awis.amazonaws.com?
  AWSAccessKeyId=
  [your AWSAccessKeyId]
  &Timestamp=
  [timestamp now, format as described in http://www.w3.org/TR/xmlschema-2/#dat
  &Signature=
```

[signature calculated from request]

&Action=

SitesLinkingIn

&SitesLinkingIn.1.Url=

[website #1]

&SitesLinkingIn.2.Url=

[website #2]

&SitesLinkingIn.3.Url=

[website #3]

&SitesLinkingIn.Shared.Count=

5

&SitesLinkingIn.Shared.ResponseGroup=

SitesLinkingIn

## API 参考

本部分包含了 Alexa Web Information Service 的细节信息，包括包括动作（Action），反馈组（Response Group）及其他组成该 API 的元素。

为帮助你开始开发，每个动作（Action）列表包含至少一个示例请求。把示例请求当作开发你自己的请求的起点。记住，在使用前要把你的访问密钥 ID（AWSAccessKeyId）放入示例中。

### 我能用 AWIS 做什么？

- 获得 web 上页面与站点的信息，包括流量、内容和相关站点等：UrlInfo
- 获得按流量排序的某个分类下的站点列表，或者为网站创建可浏览的目录：CategoryBrowse，CategoryListings
- 获得历史流量等级：TrafficHistory
- 或者指定网站的反向链接列表：SitesLinkingIn
- 从 Alexa 爬虫处获得特定网页的链接及编码：Crawl

# CategoryBrowse

## 描述

CategoryBrowse 和 CategoryListings 提供了基于开放目录，[www.dmoz.org](http://www.dmoz.org) 和 Alexa 流量数据的目录服务。

对于任何指定的分类，CategoryBrowse 返回一个子分类的列表。在特定的分类中，你可以使用 Categorylistings 来获得按流量排序的该分类的文档。

## 参数

CategoryBrowse 动作有以下参数。必须提供“必要”参数，否则查询失败。

| 名称           | 描述   | 必要 |
|--------------|--|----|
| Action       | 把 Action 参数设置为 CategoryBrowse 来获得指定分类下的子分类     | 是  |
| RespinsGroup | 任何合法的 respinse group。参见下文中的 Response Group 部分。 | 是  |
| Path         | 合法的分类路径（Top/Arts, Top/Business/Automotive）     | 是  |
| Version      | 传递当前版本号（2005-04-11），确保将来版本中 API 变更后，请求仍然能成功发送。 | 否  |
| Descriptions | 用布尔值指定是否返回分类的描述（True   False）                  | 否  |

## 反馈组（Response Groups）

反馈组允许用户对返回数据进行更多的控制。在发送请求时指定一个或多个反馈组，将使你只获得希望得到的信息。

| 反馈组                | 描述                   |
|--------------------|----------------------|
| Categories         | 返回指定分类路径下的所有子分类。     |
| RelatedCategories  | 返回与指定分类路径相关的分类列表。    |
| LanguageCategories | 返回指定分类路径中可用的语言分类的列表。 |

## 示例请求

以下是一个 Query 请求的请求与反馈的示例。

## 请求

```
http://awis.amazonaws.com?
  Action=CategoryBrowse
  &AWSAccessKeyId=[Your AWS Access Key ID]
  &Signature=[signature]
  &Timestamp=[timestamp used in signature]
  &ResponseGroup=[Valid Response Group]
```

&Path=[Top/Arts, Top/Business/Automotive]

&Descriptions=[True | False]

## 反馈

```
<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>
```

## CategoryListings

### 描述

CategoryListings是基于开放目录，[www.dmoz.org](http://www.dmoz.org)的目录服务。对于任何指定的分类，它返回该分类下包含的网站的列表。

### 参数

CategoryListings 动作有以下参数。必须提供“必要”参数，否则查询失败。

| 名称            | 描述   | 必要 |
|---------------|--|----|
| Action        | 设置 Action 参数为 CategoryListings，以获得指定分类下的站点列表   | 是  |
| ResponseGroup | Listing 是唯一可用的 response group  | 是  |
| Path          | 合法的分类路径。注意，顶级分类不回返回任何列表，除非指定 Recursive=yes（见下文）。合法的值可以是：Top/Arts, Top/Business/Automotive<br>注意，必须对这个值进行 URL 编码。例如：<br>Path=Top%2FBusiness%2FConsumer_Goods_and_Services%2FElectronics | 是  |
| SortBy        | 指定如何排序返回的结果：（Popularity   Title   AverageReview）   | 否  |
| Recursive     | 指定为只返回当前分类下的列表，或者当前目录加所有子目录。True   False   | 否  |
| Start         | 指定返回结果的起始位置。注意，如果该值大于可用结果的总数，将返回一个空文档。   | 否  |
| Count         | 指定返回结果的数量，从 Start 处开始（最大 20）。  | 否  |
| Version       | 传递当前版本号（2005-04-11），确保将来版本中 API 变更后，请求仍然能成功发送。   |    |
| Description   | 用布尔值指定是否返回分类的描述（True   False）  |    |

### 查询示例

以下是一个 Query 请求的请求与反馈的示例。

### 请求

```
http://awis.amazonaws.com?
Action=CategoryListings
&AWSAccessKeyId=[Your AWS Access Key ID]
&Signature=[signature]
&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]
</aws:CategoryBrowseResponse>
```

## 反馈

```
<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>
```

# Crawl

## 描述

返回指定文档在最近一次 Alexa 抓取时的 meta 数据。Meta 数据中包含返回码 (return code)、页面大小、校验和 (checksum)、链接的 URL、图片、框架及其他。这些 meta 信息基于 Alexa 最近一次的 web 快照。每隔一个 Alexa 抓取周期 (两个月左右)，数据更新一次。

注意，该动作不返回任何流量信息。

## 参数

Crawl 动作有以下参数。必须提供“必要”参数，否则查询失败。

| 名称            | 描述   | 必要 |
|---------------|--|----|
| Action        | 设置 Action 参数为 Crawl，以获文档的 meta 数据。   | 是  |
| ResponseGroup | 唯一合法的值是 MetaData   | 是  |
| Url           | 任何合法的 URL。该参数指定你想获得的 URL，主机或域名的相关信息。   | 是  |
| Version       | 传递当前版本号 (2005-04-11)，确保将来版本中 API 变更后，请求仍然能成功发送。  | 否  |
| Start         | 指定返回结果的起始位置。注意，如果该值大于可用结果的总数，将返回一个空文档。   | 否  |
| Count         | 指定返回结果的数量，从 Start 处开始 (最大 20)。   | 否  |
| Purify        | 在发送请求前对 URL 进行规范化 (true   false)。默认是 true。   | 否  |
| ResponseCodes | Return metadata for entries that match one of this comma-separated list of HTTP response codes (200,302) |    |

## 查询示例

以下是一个 Query 请求的请求与反馈的示例。

## 请求

```
http://awis.amazonaws.com?
Action=Crawl
&AWSAccessKeyId=[Your AWS Access Key ID]
&Signature=[signature]
&Timestamp=[timestamp used in signature]
&ResponseGroup=MetaData
&Url=[Valid URL]
&Start=[number to start at]
&Count=[Number of results to return]
```



&Purify=[true | false]  
&ResponseCodes=[200,302]

## 反馈

```
<aws:CrawlResponse 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>608de633-e4a0-422e-ab7a-517209bc0df2</aws:RequestId>
    </aws:OperationRequest>
    <aws:CrawlResult>
      <aws:Alexa>
        <aws:CrawlData>
          <aws:MetaData>
            <aws:ResultNumber>1</aws:ResultNumber>
            <aws:RequestInfo>
              <aws:OriginalRequest>http://alexa.com:80/</aws:OriginalRequest>
              <aws:IPAddress>64.213.200.100</aws:IPAddress>
              <aws:RequestDate>20070502195602</aws:RequestDate>
              <aws:ContentType>text/html</aws:ContentType>
              <aws:ResponseCode>200</aws:ResponseCode>
              <aws:Length>58319</aws:Length>
              <aws:Language>en.utf-8 0.907 2829</aws:Language>
            </aws:RequestInfo>
            <aws:Checksums>
              <aws:AppearanceChecksum>db16a79395ad7a0774faf065aee9a794</aws:AppearanceChecksum>
              <aws:ContentChecksum>60e305f16781a67a585647efc158d193</aws:ContentChecksum>
            </aws:Checksums>
            <aws:OtherUrls>
              <aws:OtherUrl source="href">www.alexa.com/favicon.ico</aws:OtherUrl>
              <aws:OtherUrl source="src">purl.org/atom/ns</aws:OtherUrl>
            </aws:OtherUrls>
            <aws:Images>
              <aws:Image>client.alexa.com/common/images/alexa.gif</aws:Image>
              <aws:Image>client.alexa.com/common/images/button_search_arrow.gif</aws:Image>
            </aws:Images>
            <aws:Links>
              <aws:Link>
                <aws:LocationURI>www.alexa.com/</aws:LocationURI>
              </aws:Link>
              <aws:Link>
                <aws:Name>Traffic Rankings</aws:Name>
                <aws:LocationURI>alexa.com/site/ds/top_500?qterm=</aws:LocationURI>
              </aws:Link>
```

```
</aws:Links>
</aws:MetaData>
</aws:CrawlData>
</aws:Alexa>
</aws:CrawlResult>
<aws:ResponseStatus xmlns:aws="http://alexa.amazonaws.com/doc/2005-10-05/">
<aws:StatusCode>Success</aws:StatusCode>
</aws:ResponseStatus>
</aws:Response>
</aws:CrawlResponse>
```

# SitesLinkingIn

## 描述

SitesLinkingIn 返回指定网站的反向链接列表。在链接入该网站的域名中，只返回页面流量最高的链接。此数据每两个月更新一次。

## 参数

SitesLinkingIn 动作有以下参数。必须提供“必要”参数，否则查询失败。

| 名称            | 描述   | 必要 |
|---------------|--|----|
| Action        | 设置 Action 参数为 SitesLinkingIn，以获文档的 meta 数据。    | 是  |
| Url           | 任何合法的 URL。该参数指定你想获得的 URL，主机或域名的相关信息。           | 是  |
| ResponseGroup | 唯一合法的值是 SitesLinkingIn。                        | 是  |
| Count         | 指定返回结果的数量，从 Start 处开始（默认是 10，最大 20）。           | 否  |
| Start         | 指定返回结果的起始位置。默认是 0                              | 否  |
| Version       | 传递当前版本号（2005-07-11），确保将来版本中 API 变更后，请求仍然能成功发送。 |    |

## 查询示例

以下是一个 Query 请求的请求与反馈的示例。

### 请求

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

### 反馈

每个反向链接返回一个 aws:Site 元素。Title 和 url 为该网站流量等级最高的页面所属。

```
<aws:SitesLinkingInResponse 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>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://alex.amazonaws.com/doc/2005-10-05/">
<aws:StatusCode>Success</aws:StatusCode>
</aws:ResponseStatus>
</aws:Response>
</aws:SitesLinkingInResponse>
```

# TrafficHistory

## 描述

TrafficHistory 返回每日 Alexa 流量等级，每百万用户访问和每百万用户的唯一页面浏览。Alexa.com 网站上的流量图即用此数据生成。

## 参数

TrafficHistory 动作有以下参数。必须提供“必要”参数，否则查询失败。

| 名称            | 描述   | 必要 |
|---------------|--|----|
| Action        | 设置 Action 参数为 TrafficHistory 以获得一定时间内的流量等级。    | 是  |
| Url           | 任何合法的 URL。该参数指定你想获得的 URL，主机或域名的相关信息。           | 是  |
| ResponseGroup | 唯一可用的值是 History。                               | 是  |
| Range         | 返回的天数。注意，如果数据不可用，返回的文档将包含较少的结果。默认值：31。最大值：31。  | 否  |
| Start         | 结果的起始日期。第一个可用的日期是 20010901（2001 年 9 月 1 日）     | 否  |
| Version       | 传递当前版本号（2005-07-11），确保将来版本中 API 变更后，请求仍然能成功发送。 |    |

## 查询示例

以下是一个 Query 请求的请求与反馈的示例。

## 请求

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

## 反馈

每个指定的日期范围返回一个 aws:Date 元素。注意，排名在十万以后的网站不返回数据。

```
<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-ec1a-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>
```

## UrlInfo

### 描述

URL Information 提供关于网站的信息。示例信息包括站点的流行程度、相关站点和网站所有者的联系信息。

### 参数

TrafficHistory 动作有以下参数。必须提供“必要”参数，否则查询失败。

| 名称            | 描述   | 必要 |
|---------------|--|----|
| Action        | 设置 Action 参数为 UrlInfo 以获得特定 URL 的信息。           | 是  |
| Url           | 任何合法的 URL。该参数指定你想获得的 URL，主机或域名的相关信息。           | 是  |
| ResponseGroup | 任何合法的 respinse group。参见下文中的 Response Group 部分。 | 是  |
| Version       | 传递当前版本号（2005-07-11），确保将来版本中 API 变更后，请求仍然能成功发送。 | 否  |

### 反馈组（Response Groups）

反馈组允许用户对返回数据进行更多的控制。在发送请求时指定一个或多个反馈组，将使你只获得希望得到的信息。

| 反馈组           | 描述  |
|---------------|---|
| RelatedLinks  | 返回最多 11 个与当前网站或页面相关的链接。                         |
| Categories    | 返回最多 3 个当前站点在 DMOZ（开放目录）所属分类                    |
| Rank          | 返回指定站点在 Alexa 三个月的平均流量等级                        |
| RankByCountry | 显示基于该网站的访问用户所在国家的最高浏览者、页面浏览次数和流量等级。             |
| RankByCity    | 显示基于该网站的访问用户所在城市的最高浏览者、页面浏览次数和流量等级。             |
| UsageStats    | 返回指定网站的使用数据，例如 Reach 和 PV。                      |
| ContactInfo   | 返回网站所有者或注册者的信息。                                 |
| AdultContent  | 如果网站不包含成人内容，返回 no，否则返回 yes。                     |
| Speed         | 返回平均读取速度和得分。                                    |
| Language      | 返回指定站点的语言编码和字符编码。注意，这并不是所有页面的语言或字符编码，而只是大多数页面的。 |
| Keywords      | 返回用于辨识该网站的关键词列表。                                |
| OwnedDomains  | 返回该网站所有者拥有的其他域名。                                |
| LinksInCount  | 返回反向链接的数量。                                      |
| SiteData      | 返回为该站点创建的 Title，Description 和 Date。             |

### 查询示例

以下是一个 Query 请求的请求与反馈的示例。

## 请求

```
http://awis.amazonaws.com?  
Action=UrlInfo  
&AWSAccessKeyId=[Your AWS Access Key ID]  
&Signature=[signature]  
&Timestamp=[timestamp used in signature]  
&Url=[Valid URL]  
&ResponseGroup=[Valid Response Group]
```

## 反馈

```
<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,  
<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>
```