

## API Documentation

Last Updated: 8/25/2008

### Quick Overview

The UrlTrends API is designed to allow programmatic access to the all of the data that UrlTrends has gathered for any URL in our database, and in some cases URLs that are not in our database.

There are 9 different methods and 1 endpoint. The endpoints are as follows:

- GetInfo
- GetInfoLive (Formerly named: GetRankLive)
- GetHistory
- GetSearchTerms
- GetSearchTermsAndInfo
- AddKeywords
- GetCache
- GetWhoisLive
- GetWhoisHistory
- GetWhoisFromHistory

Each of these methods can be accessed at the following endpoint (location):

<http://apps.urltrends.com/api/2.0/>

**PLEASE NOTE:** The API endpoints that are/were available at <http://api.urltrends.com/> and at <http://beta.api.urltrends.com/> are now officially deprecated and will no longer be available after the changeover. After the changeover, all requests to the old API locations will be 301 redirected to the new location. If your application is not modified to use the new endpoint before the changeover period ends your application may start receiving errors or fail after the 301 redirects begin.

At this time the UrlTrends API is only available via a [REST](#)-like interface. Each endpoint available supports being called by HTTP POST or HTTP GET methods interchangeably. However, unless you have a specific reason to use HTTP POST, it is recommended that you use the HTTP GET method.

Because our API is made via REST calls you can use any programming language that allows you the ability to make HTTP calls over the internet.

## Requirements to use the UrlTrends API

To use the UrlTrends API you must be registered for a My UrlTrends account that has API capabilities.

For initial testing and implementation purposes, you can [register for our Basic Plan](#) (our free plan – not the API Basic plan) and [request an API key](#). Once you have completed testing, you must upgrade your account to our API Basic Plan, API Advanced Plan or the API Premium Plan prior to making your application public (internal or externally).

Each API plan that we offer have specific costs, features and limits associated with them. This information is outlined on our registration page as well as [here](#).

*If you expect that you will need more than 20,000 queries per day, please contact us for partnership opportunities or to arrange for a custom plan to be created.*

## Response Caching

Please note that we cache all API calls for 12 or 24 hours, depending on the endpoint. This includes the GetInfoLive call. It does not include the AddKeywords response.

Inside the details of each endpoint below there will be a section that explains what length of time applies to that endpoints cache.

## Rate Limiting / Throttling

The UrlTrends API currently has no limit on the number of requests applications can make concurrently. Please try to use the minimum number of requests needed to accomplish a task and avoid large amounts of simultaneous queries if possible.

If you need help figuring out how to efficiently perform a task or if you would like to determine if the number of simultaneous queries that you intend on performing will burden our API, please contact us.

You may use up to your daily limit each day. The counter resets at approximately 12:00 am PDT/PST (California time) each day. To see what the daily limit is for our API plans, please go [here](#).

## Cross Domain XML

Adobe Flash applications require a cross domain XML file to access the UrlTrends API. The cross domain XML file can be found at: <http://apps.urltrends.com/api/crossdomain.xml>

## Pass Through Variables

The UrlTrends API supports pass-through variables. To pass a variable through the API, simply append any variable to the end of our API's URL that is not a reserved variable. For example, below I appended "&testing=123" to the end of the URL to the API and the following was included in the XML result.

```
<Passthru>  
  <testing>123</testing>  
</Passthru>
```

In most cases you won't need this feature, but it's there if you need it. Most companies that use this feature use it to pass their internal ID's that are associated with the URL so that they can track it in the results.

Pass through variables work on any call that you make to the API. The variables are not cached by our caching system.

## Endpoint Variables

### REQUIRED Variables in all API Queries

The following arguments are required on all API calls:

#### ***command***

This should be set to one of the available methods that is outlined in the Quick Overview section. The section that is highlighted in yellow is the command variable.

```
http://apps.urltrends.com/api/2.0/GetInfo.xml?url=http%3A%2F%2Fwww.urltrends.com&api_key=[API_KEY]
```

#### ***format***

The API supports returning results in XML, JSON, or a serialized PHP array by setting the format similarly to a file name extension. If the value is not set to xml, php or json, XML will be returned.

```
http://apps.urltrends.com/api/2.0/GetInfo.xml?url=http%3A%2F%2Fwww.urltrends.com&api_key=[API_KEY]
```

#### ***api\_key***

When requesting information from the API you must use the API key that was assigned to you when you registered for your UrlTrends API account. The API key is an alphanumeric string that is 64 characters in length. E.g. `api_key=akie5vld8gm2n5lk4a1dsy53dvh8d2diau6mdygbcp08ncvrshyunwqad6hinkd3`

In the example below, you would replace the [API\_KEY] with your 64 character key.

```
http://apps.urltrends.com/api/2.0/GetInfo.xml?url=http%3A%2F%2Fwww.urltrends.com&api_key=[API_KEY]
```

**PLEASE NOTE:** There is now an underscore between "api" and "key". In previous versions of our API the underscore did not exist and you would have just used "apikey". This change was made so that existing developers using our API would be required to double check all of their API calls and results after the changeover, especially in the case that calls become 301 redirected due to not modifying their code before the changeover date. We apologize in advance for any inconvenience that this causes you.

## Endpoint Details

### GetInfo

The GetInfo endpoint is used to return the latest snapshot of information for an URL. It does not include historical information, but it does include important information like when we last monitored a URL, when the next time that it is scheduled for an update all of its latest search engine ranking information and more.

This API call is cached for 12 hours.

#### Example URL:

```
http://apps.urltrends.com/api/2.0/GetInfo.xml?url=http%3A%2F%2Fwww.urltrends.com&api_key=[API_KEY]
```

#### Optional Variables:

```
&update=0|1
```

Set the update variables value to 1 to add the URL to our systems if it does not already exist. Set it to 0 if you do not want to add the URL to our systems. By adding it to our systems we will start monitoring it monthly or weekly, depending on who is watching it. If it is not set it will default to 0.

```
&wait=0|1
```

Set the wait variable value to 1 if you want to wait for the results when we do not have the URL in our database. Set it to 0 if you don't want to wait. This option is only used when a URL is not in our database and the update value is set to 1. The default option if the variable is not set is 1.

This setting is the most useful when you have a large list of URLs to add to our database for monthly/weekly monitoring. By setting it to 0 you can quickly cycle through all of the URLs without waiting for our systems to return the information.

If this variable is set to 0, our API adds the URL to a queue to be processed later and instantly returns a message to your application so that it is not waiting for us to get all of the information on the URL. Typically within a minute the information will be available for you to access.

If set to 1 your application will wait for us to return the information. If we have the URL in our database our API will typically respond within a 10<sup>th</sup> of a second with the information requested. If the URL is NOT in our database your application will be waiting from 2 to 60 seconds while we gather the information. If you set this variable to 1 make sure that your application will wait up to 60 seconds before timing out.

```
&sendifavail=0|1
```

If this variable is set to 1 we will reply to your request with the search engine information for a URL. If set to 0, we will only reply with the date that the urls information was last updated. It defaults to 0 if not set.

This is most useful to see if a urls information has been updated. It should be set to 0 if the only purpose of the query is to see if the information has been updated. The extra information not sent will save both you and UrlTrends on unnecessary bandwidth usage.

#### **Required Variables:**

```
&url=[ URL encoded URL ]
```

### **GetInfoLive**

The GetInfoLive call responds in a format similar to the GetInfo call. The primary difference is that the data is collected live and you can control exactly what information that you want to receive. The information returned will not be stored in a URLs history, even if we are tracking it monthly or weekly.

This API call is cached for 12 hours.

#### **Example URL:**

```
http://apps.urltrends.com/api/2.0/GetInfoLive.xml?url=http%3A%2F%2Fwww.urltrends.com&api_key=[API_KEY]&type=Rank_PR:Rank_AR:Link_Google:Link_Yahoo:Site_Google:Site_Yahoo
```

This example URL gets the Google PageRank, Alexa Rank, Incoming Google links (backlinks), Incoming Yahoo links, Google site saturation for the domain and the Yahoo saturation for the domain.

### Required Variables:

```
&type=[ value ]
```

The type variable can have one or more text values. If more than one item is desired, separate the values with a colon.

Here are the possible values and a short description specifying what they will return:

Rank_PR	Returns the common Google PageRank and the average Google PageRank after checking 5 Google datacenters.  We check 5 Google datacenters and then return the value that is most common among them so that you have the most accurate result.  When you use this option we also return an additional element called "Rank_PR_AVG". This is the average rank returned after dividing the average PR by the number of datacenters that we checked.
Rank_AR	Returns the current Alexa Rank for this URL.
Cache_Archive	Returns the current number of items in the "Wayback Machine" at archive.org
Link_Yahoo	Returns the current number of backlinks at Yahoo!
Link_Google	Returns the current number of backlinks at Google.
Link_AllTheWeb	Returns the current number of backlinks at AllTheWeb!
Link_Altavista	Returns the current number of backlinks at AltaVista.
BM_Delicious	Returns the current number of times that this URL has been bookmarked on the social bookmarking service, del.icio.us.
BM_Furl	Returns the current number of times that this URL has been bookmarked on the social bookmarking service, Furl.
BM_YahooMyWeb	Returns the current number of times that this URL has been bookmarked on

	the new social bookmarking service, Yahoo My Web.
Site_Google	Returns the site saturation for the domain of this URL on Google.
Site_Yahoo	Returns the site saturation for the domain of this URL on Yahoo!
SV_Digg	Returns the highest number of times that this URL has been “Dugg” on Digg. It also returns the number of Digg stories that have been submitted for this URL.
SV_Twitturly	Returns the number of times that this URL has been mentioned on Twitter.
Blog_Technorati	Returns the number of blog posts that mention this URL on Technorati.
Blog_Icerocket	Returns the number of blog posts that mentions this URL on IceRocket.

&url=[ URL encoded URL ]

#### Optional Variables:

*There are no optional variables for this endpoint.*

**Note:** We request the information in this call in real-time (live). When requesting this information, we do it in parallel to each other so that it returns the results as quick as possible. However, the more items that you include, the longer your application *may* be waiting for the results. If one service takes a while to respond, it will delay the entire result set.

- Please do not request values that you do not intend to use.
- Please combine your call to get the items that you need in one call to save your API queries.

### GetHistory

The GetHistory endpoint is used to get all of the historical information that we have on the URL provided. The data returned in this call is the same data that we use to generate our charts.

This API call is cached for 24 hours.

#### Example URL:

```
http://apps.urltrends.com/api/2.0/GetHistory.xml?url=http%3A%2F%2Fwww.urltrends.com&api_key=[API_KEY]
```

**Required Variables:**

```
&url=[ URL encoded URL ]
```

**Optional Variables:**

*There are no optional variables for this endpoint.*

**GetSearchTerms**

The GetSearchTerms endpoint is used to get every search term that we have found the URL provided to be ranking in the top 30 positions at Google, Yahoo! or MSN. This response is similar to the GetSearchTermsAndInfo call, but it does not use return the extra ranking information that that call does. If you do not need the positions or engines that we found the term for, please use the call as it is much less resource intensive.

This API call is cached for 24 hours.

**Example URL:**

```
http://apps.urltrends.com/api/2.0/GetSearchTerms.xml?url=http%3A%2F%2Fwww.urltrends.com&api_key=[API_KEY]
```

**Required Variables:**

```
&url=[ URL encoded URL ]
```

**Optional Variables:**

*There are no optional variables for this endpoint.*

**GetSearchTermsAndInfo**

The GetSearchTermsAndInfo endpoint is used to get every search term that we have found the URL provided to be ranking in the top 30 positions at Google, Yahoo! or MSN. This response is similar to the GetSearchTerms call, but it also includes the position information as well as the number of sites found on each search engine for that term. If you do not need the additional information that this call provides over the GetSearchTerms call, please use the GetSearchTerms call as it is much less resource intensive.

This API call is cached for 24 hours.

**Example URL:**

```
http://apps.urltrends.com/api/2.0/GetSearchTermsAndInfo.xml?url=http%3A%2F%2Fwww.urltrends.com&api_key=[API_KEY]
```

**Required Variables:**

```
&url=[ URL encoded URL ]
```

**Optional Variables:**

*There are no optional variables for this endpoint.*

**AddKeywords**

The AddKeywords endpoint is used to add keywords into our keyword monitoring system. When a keyword or phrase is added to our systems via this call we will monitor the keyword for the top 30 sites on Google, Yahoo and MSN.

The keywords that you add via this call are used in the GetSearchTerms and GetSearchTermsAndInfo calls.

This should not be confused with a similar service that is provided to our My UrlTrends users. This system is the system that adds keyword to our publicly visible tools like our Trend Report and Reverse Keyword Lookup Tool.

This API call is not cached.

**Example URL:**

```
http://apps.urltrends.com/api/2.0/AddKeywords.xml?keywords=Keyword&api_key=[API_KEY]
```

Please note that the keyword variable should be URL encoded.

There is no response section on this API call. To verify that your call was successful, check the value of the ResponseCode in the API section to see if the result equals 200.

### GetCache

The GetCache endpoint is used to get a list of the UrlTrends cache items that we have for this URL. The results provide links to where the cached pages can be viewed.

This API call is cached for 24 hours.

#### Example URL:

```
http://apps.urltrends.com/api/2.0/GetCache.xml?url=http%3A%2F%2Fwww.urltrends.com&api_key=[API_KEY]
```

#### Required Variables:

```
&url=[ URL encoded URL ]
```

#### Optional Variables:

*There are no optional variables for this endpoint.*

### GetWhoisLive

The GetWhoisLive endpoint is used to get the latest publicly available whois record for the domain name provided.

This API call is cached for 24 hours.

#### Example URL:

```
http://apps.urltrends.com/api/2.0/GetWhoisLive.xml?cdn=urltrends.com&api_key=[API_KEY]
```

**Required Variables:**

```
&cdn=[ domain name ]
```

**Optional Variables:**

```
&type=quick
```

If you set the “type” variable to “quick”, we will get the whois information, and return the registration status, expiration, creation and last change information – essentially, everything but the raw whois output. If this variable is omitted, then we will return everything for the whois including the raw whois data. If you do not need the raw whois data, please set this option to help save bandwidth and resources.

**GetWhoisHistory**

UrlTrends regularly checks the whois record for all URLs that we monitor, and when there is a change to the record over the previous version that we have stored, we store the latest version too. The GetWhoisHistory endpoint is used to get a list of the whois records for the domain name provided that we have on file. The result provides the dates that the whois was changed on. This call is typically used in conjunction with the GetWhoisFromHistory endpoint.

This API call is cached for 24 hours.

**Example URL:**

```
http://apps.urltrends.com/api/2.0/GetWhoisHistory.xml?cdn=urltrends.com&api_key=[API_KEY]
```

**Required Variables:**

```
&cdn=[ domain name ]
```

**Optional Variables:**

*There are no optional variables for this endpoint.*

## GetWhoisFromHistory

The GetWhoisFromHistory endpoint is used to get the whois record for a specific date and domain name provided. The result provides the entire whois results as if you had just annually pulled it on the date provided. This call is typically used in conjunction with the GetWhoisHistory endpoint, which is used to get the dates that we have versions for.

This API call is cached for 24 hours.

### Example URL:

```
http://apps.urltrends.com/api/2.0/GetWhoisFromHistory.xml?cdn=urltrends.com&api_key=[API_KEY]
```

### Required Variables:

```
&cdn=[ domain name ]
```

### Optional Variables:

```
&updated=[YYYY-MM-DD]
```

The updated variable is used to specify which whois record to return. The valid dates available to be used can be obtained through the GetWhoisHistory endpoint. If this variable is omitted, the latest whois record will be returned.