

# **DomainPlex API Documentation**

Copyright © 2018 DomainPlex Inc.

**IMPORTANT:** Please check back often for the latest API updates and to ensure that your application is compliant with our services.

We are not associated with any search engine or directory mentioned in this documentation. All data and services are provided AS IS.

## Table of Contents

Introduction.....	3
API Call Basics.....	4
API Data Request Format.....	5
API Data Output Format.....	7
Supported Extensions.....	8
Using the API.....	9
Sending Appraisal Queries.....	9
Sending Keyword Queries.....	10
Sending Whois Queries.....	11
Sending Lead Generator Queries.....	12
Sending Trademark Check Queries.....	14
Sending Word Parser / Language Detection Queries.....	15
Sending DNS Domain Availability Queries.....	15
Sending Registry-Level Domain Availability Queries.....	16
Sending Sold History Search Queries.....	17
Sending Reverse IP Lookup Queries.....	19
Sending SERP Queries.....	19
Downloading Domain Lists with the API.....	20
Downloading Raw Domain Lists.....	21
Downloading Processed Domain Lists.....	22
Downloading Zone Files.....	22
Advanced API Calls.....	24
Checking Session Progress.....	24
Stopping Queued Sessions.....	25
Downloading Completed Sessions.....	25
Creating Simple Notifications.....	25
Creating Post-back Notifications.....	26
Viewing Usage Statistics.....	27
API Command List.....	28
Aftermarket Domain Search.....	28
Cached Alexa queries.....	28
Appraisals.....	28
Cached appraisals.....	29
Access historical appraisal data for any domain.....	29
Combined metrics lookup for all major metrics.....	29
Live availability check.....	30
Domain Backlinks Report.....	30
Domain Backlinks Count.....	31
Domain Backlnks Raw.....	31
Backorder Bid Buy Domains.....	31
Domain Name Categorization System.....	31
Custom droplists.....	31
DNS based availability check.....	32
Domain Name Generator.....	32
Shows google trends.....	33
Domain idea generator.....	33
Domain end-user lead generator.....	33
Quick Lead Generator.....	34
Calculate lead score given a domain name. Custom tool.....	34
Domain name lists and zone files.....	35
Monitor domains, registrants, DNS records.....	35

Check moz domain authority score.....	35
Cached Overture queries.....	35
Pagerank Check.....	36
Pagerank History Check.....	36
Word parser and language detection.....	37
Raw PendingDelete file with drop order.....	37
Crawl sites, resolve IP & parse site data.....	37
Perform reverse Analytics / AdSense lookups.....	38
Reverse DNS lookup.....	38
Reverse IP lookups.....	38
Reverse whois for domain and email.....	39
Keyword search volume & CPC.....	39
Cached SERP queries.....	40
SERP Cache.....	41
Serp Summary (serp count and ad count) pulled from cache.....	42
Web crawl summary for a specific domain.....	42
Site crawler summary database for TLD and status lookups.....	42
Site Information.....	43
Shows the top keywords and their density that appear on any domain name or web site.....	43
Website Snapshot / Thumbnail.....	43
Website Snapshot History.....	44
Shows most recent snapshot for the site, as an image.....	45
Domain sold history searches.....	45
System Information (High Level Accounts).....	46
Trademark Analysis.....	46
Typo generator.....	46
Wayback Records Crawler.....	47
Whois queries.....	47
Historical Whois Lookup.....	47
Raw Archived Whois Data.....	48
Archived Whois Data.....	49
Check the most recent cached whois record.....	49
Whois history database.....	49
Get most recent raw historical whois record for a domain or a set of domains.....	50
Redundant whois queries.....	50
Reverse whois search email or registrant name showing all domains associated with either an email or registrant name.....	51
Cached reverse lookup using secondary DB.....	51
Reverse Whois Search.....	51
Cached Wordtracker queries.....	51
Archive of zone files.....	52
Zone Changes Database.....	52
Zone Adds/Drops.....	53
Zone Digger.....	53
Zone Domains.....	54
Zone Search.....	54
Sample HTTP POST Functions.....	55
Enterprise Services.....	59
Errors.....	60
Support.....	61
Glossary of Terms.....	62

## Introduction

The DomainPlex API was developed to allow instant access to the vast data and tools available on the DomainPlex network. DomainPlex stores archived search engine metrics (such as link popularity, PageRank and search results), web site snap shots (for over 100 million domains), whois data (for over 220 million domains), search volume and CPC information and much more. In addition to storing archived data, DomainPlex is also capable of querying various search engines, directories and data sources for **live** metrics. This means that metrics which do not exist in our archives or which are out of date are automatically refreshed.

Tools such as dynamic end-user lead generator, trademark and typo analyzer, reverse IP lookup, domain generator, typo generator and many others are also available via API calls.

Here are a few things you can do with the DomainPlex API:

- Instantly Appraise Domains
- Generate End-User Leads for Domains
- Parse Domains to Words and Determine Domain Language
- Check Domain Trademarks and Analyze Risk Level
- Check Whois for Domains with Registrant and E-mail parsing
- Check Search engine stats (Altavista, Bing, Google)
- Check Alexa, Wayback, PageRank
- Check Advertisers and Search Results for any keyword
- Check Monthly Search Volumes and CPC for any keyword
- Check Domains for DNS availability and live availability
- Check Archived Domain Sales Records (updated hourly)
- Check All domains hosted on a particular name server
- Reverse IP Lookups (lookup domains hosted on an IP)
- Crawl domains and get parsed title, metadata, server details, parking status and much more
- Download processed drop lists and zone files
- Generate typos, generate domain ideas from search data and much more

The API uses simple **GET/POST** requests to send and received data to and from our servers. Results are provided using non-standard (default) or standard XML format which can easily be parsed into variables using PHP, Perl, ASP and other programming languages.

Live queries are processed on our vast network of servers using multithreaded processes to ensure the quickest and most efficient data delivery.

API access is limited by daily query volume for each user and tool. Some users may not have access to all available tools and data sources.

## API Call Basics

An API call is a query to the DomainPlex API server which results in some data being processed and returned.

All API calls are treated as unique **sessions**. A session is synonymous to a job. With every API call, you tell the server to perform a certain task, such as an appraisal or whois query. Some jobs may take a fraction of a second to complete while others may take hours (such as a large-scale bulk appraisal). A method for keeping track of active and completed jobs is a unique session.

A session has certain data associated with it such as session name, session data as well as progress indicators and other internal attributes. Sessions are used to check the status of queued items, cancel active queues, download processed results and much more.

Sessions are also used to keep track of active scans for each user. To prevent service abuse and allow equal access for all users, every DomainPlex account is granted a limited number of active sessions. An active session is defined as a session that is currently processing data (such as a bulk appraisal). Most users are granted three sessions per tool (ex: whois, appraisal, resolve are all separate tools with three sessions each). A session can have a single domain or thousands of domains associated with it. Sessions are not limited by the quantity of domains or keywords they process, but by the number of separate active threads they occupy. In other words, you're allowed three active jobs to run at the same time, for each tool. Those active jobs could each be processing one domain or a thousand domains. The session threshold can be increased upon request.

Session name and other data is returned only on bulk scans and when the **release\_socket** command is issued as part of the API call.

By default, all API calls process a request and return results to the user within the same GET/POST response query. While this approach works for single queries, bulk queries may take anywhere from a few minutes to a few hours (depending on how many items were submitted into the queue) to finalize and the web server will generally close the connection to conserve resources. Sessions were introduced to allow for "background" scanning and for completion notification and instant data delivery when long scans complete.

**All recent results are stored in a temporary cache for a period of seven days.** In other words, you can retrieve the results immediately after the query is processed (for most queries such as appraisals, whois, etc), or a few hours or days later via a special API call (see "[Downloading Completed Sessions](#)" section). This ensures that data is never lost in the delivery stage due to browser or script timeout or other problems. You can also configure the API to post the results or notifications back to your server when a long scan is completed by specifying a post-back URL (see "[Creating Notifications](#)" section).

## API Data Request Format

Requests to the API are made using simple GET/POST HTTP queries compatible with all scripting languages such as PHP, Perl, Python as well as desktop based development environments such as .NET, ASP and many others. You can even paste an API call directly into your browser and see the results in XML format.

All requests are divided into at least three individual segments, but are part of a **single** URL.

- Authentication Segment
- Command Segment
- Data Segment
- Optional Segment
- Optional Format Segment

Below is a typical request URL to pull appraisal data for “example.com”:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=appraise&t=example.com&xml_standard=1
```

### Authentication Segment:

Calls to the DomainPlex API must include a form of authentication. This can be your account's API key or your account's username and password. For the security of your account, we **highly** recommend using key-based authentication. You can get or change your account's API key from your [Profile](#) page.

#### Key-based authentication - recommended method

Supply your account's API key with the `&api_key=KEY` parameter. Your key is a 32-character sequence of letters and numbers. Ensure that your key is passed exactly as it appears on your Profile page.

Example:

```
http://api.domainplex.com/?a=get&api_key=KEY ..
```

#### Username and password authentication - alternate method

Pass the `&u=USER` and `&p=PASS` parameters to authenticate with your account's username and password. The `u` parameter specifies your DomainPlex username while the `p` parameter specifies your DomainPlex password.

When implementing the API, store your account's API key and/or login information in a secure place that cannot be publicly accessed. Your API key should be treated with the same level of security as your username and password.

Example:

```
http://api.domainplex.com/?a=get&u=USERNAME&p=PASSWORD ..
```

### Command Segment:

The command segment refers to the `&d=` parameter which must be provided with every request. This parameter specifies the database or tool you wish to access via the API call. Please see the [API Command List](#) section for a complete list of valid `d` parameters.

### Data Segment:

The data segment refers to the `&t=` parameter which must be provided with every request. This parameter specifies the data you wish to process via the API. For example, with appraisal requests, you would specify a domain or a set of domains.

Most commands accept multiple items as part of the `t` parameter. The item separator is `>>`, meaning that you can submit multiple domains (or keywords) to the API by separating them with `>>` (no quotes).

Below is a typical request URL to pull appraisal data for `example.com` and `games.com` (notice the `>>` separator):

```
http://api.domainplex.com/?a=get&api_key=KEY&d=appraise&t=example.com>>games.com&xml_standard=1
```

### Optional Segment:

The optional segment refers to the optional `&o=` parameter. The optional parameter specifies extra data which may be provided with some queries (such as live search engine queries).

Individual tools may have other parameters that can be used to control behavior and output. Refer to the [API Command List](#) section for details on individual tools and their optional parameters.

### Optional Format Segment:

The optional format segment refers to the optional `&xml_standard=` parameter. The parameter specifies whether or not results should be provided in standard XML format (which can be parsed by any XML parser) or the smaller, but non-compliant DomainPlex XML format.

**We strongly recommend using a POST query even when submitting a single item.** A POST query will ensure that our web server properly receives and processes your data without the limitations of a GET query. For the sake of brevity, all of the example URLs in this document are shown using GET queries but can be sent with POST queries instead.

We do **not** recommend using PHP's `file_get_contents` function to perform API calls. Using `file_get_contents` will result in unexpected errors when processing bulk lists.



## API Data Output Format

Output format for all results is either non-standard XML (default) or, if the `&xml_standard=1` parameter is provided, standard XML.

Adding `&xml_standard=1` to the end of any API call will generate standardized XML format results which can be parsed by any XML parser or web browser. If the parameter is not supplied, DomainPlex will output the results in non-standard format to conserve bandwidth.

Typical (partial) output for an appraisal call is provided below (in standard XML):

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <appraisal domain="example.com">
    <domain_cc>Example.com</domain_cc>
    <sld>example</sld>
    <tld>com</tld>
    <words>example</words>
    <language>english</language>
    <language_probability>100.0</language_probability>
    <category>Other -- Generic Single Word</category>
    <appraised_value>325000</appraised_value>
    ...
  </appraisal>
</results>
```

The API uses **two** standard XML output formats based on the tool queried. The attributes are separated by either a new line (as seen in the image above for the appraisal API) or by listing multiple attributes on the same row (partial lead generator example below).

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <result domain="example.com" lead_domain="example1.com">
  <result domain="example.com" lead_domain="example2.com">
  <result domain="example.com" lead_domain="example3.com">
  ...
</results>
```

## Supported Extensions

We're constantly adding support for new gTLDs, ccTLDs and nTLDs. You can check the complete list of supported extensions using the API call below.

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois&o=supported_extensions
```

Please note that not all TLDs are supported by all tools. Contact Support if you have any questions.

## Using the API

Now that you've read and understand the query and response format you can begin sending API queries and parsing out the results.

---

### Sending Appraisal Queries

The appraiser has several modes of operation which determine the quality of the appraisal and the amount of data returned.

The deciding factor is always time, measured in either milliseconds or seconds. Based on the number of domains you want to appraise, the data you want to see and the maximum acceptable processing time, you should decide which mode to use.

Since the appraiser relies on various external metrics such as search volume and search popularity it may be necessary to perform live lookups at various data sources. Live lookups are time consuming, lasting anywhere from .25 seconds to 15 seconds and slow down the appraisal process. With large domain lists the slow down is definitely noticeable between the various modes of operation.

A standard appraisal call (with no optional parameters) will perform the following live lookups:

- Search Volume Lookup
- Search CPC Lookup
- Search Trend Lookup
- Search Results Lookup
- Search Ads Lookup
- Alexa Lookup
- PageRank Lookup
- DNS Lookup
- Overture Archive Lookup
- Word Tracker Archive Lookup
- Trademark Lookup
- Language Detection Lookup

Appraisal URL with no optimization parameters used to generate the above data:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=appraise&t=example.com&xml_standard=1
```

If one of the above metrics is not found in the database, or falls beyond the threshold of acceptable age tolerance (ex: metric is 30+ days old), it is checked live.

As you can see, there are quite a few metrics checked using the standard appraisal call. You may not be interested in seeing all of them, and many of them are not used by the appraisal algorithm. Therefore, you can make the appraiser run quicker and more efficiently by specifying the data you want returned.

### Making the Appraiser More Efficient

Several parameters are available that speed up the processing time and cut down on the amount of data returned, resulting in more efficient queries.

```
&efficient_mode=1
```

**Efficient mode**, which is the **recommended mode**, looks up only the metrics vital to the appraisal algorithm. Non-vital metrics are only checked in the cached database. If they do not exist in the archive, they are not looked up live. This ensures fast results and the most complete data whenever that data is available in cache.

The efficient mode ensures **high appraisal quality** and fast processing time. We strongly recommend adding `&efficient_mode=1` to all API appraisal queries.

```
&no_seo=1
```

When the **no\_seo mode** is enabled, the appraiser will not perform any non-vital search engine or metric queries. Metrics such as Alexa, Pagerank will not be looked up or returned (either cached or live).

The `no_seo` mode guarantees high appraisal quality and fairly fast processing time.

```
&fast_mode=1
```

When the **fast mode** is enabled, the appraiser performs no live queries. If the vital search data is not in the database, it will severely affect the appraiser. This is the fastest appraisal mode, usually generating appraisals in  $< 1/2$  seconds (.25 seconds on bulk scans), however it does not guarantee high quality results.

Use this scan mode when scanning large lists or when you do not need to have the highest quality appraisal.

**IMPORTANT:** This API call supports post-back and `release_socket`. This means that you can submit large lists and download the results when they are finished by applying the `&release_socket=1` parameter to the end of the API call. Additionally, by providing a post-back URL (see section “[Creating Post-back Notifications](#)”) you can have the results submitted back to your server when they are finished processing.

## Sending Keyword Queries

You can instantly check search volumes and CPC data for **billions** of keywords using the DomainPlex API.

The keyword data comes from an aggregate search feed that represents searches made on multiple engines including Google, Yahoo and Bing. We combine multiple feeds and data sources (such as toolbar queries, ISP logs and many other sources) into a normalized database that represents both US and global searches.

Searches are expressed as the number of **average searches performed per month on all search engines**. CPC is expressed as a dollar value of the average cost per click. The results include regional and global searches. Regional searches refer to US searches while global refer to world searches.

Calling the keyword API is quick and easy. All results are pulled from cache and returned instantly. If a particular keyword has less than 10 searches per month, it is stored in our database with a value of `-1`. The API supports looking up three types of keywords.

- **Broad** keywords (ex: if the keyword was “fast cars” then broad match would also include “fast”, “cars”, “fast cars photos”, etc.).
- **Phrase** keywords (ex: if the keyword was “fast cars” then broad match would also include “fast cars photos”, “very fast cars” and other keywords containing “fast cars” within them). Phrase keywords must be enclosed quotes, for example “KEYWORD”.
- **Exact** keywords (only searches matching the exact keyword are shown). Exact keywords must be enclosed with brackets, for example [KEYWORD].

Here is a sample API call to check broad, phrase and exact keywords (respectively) and return the results in standard XML format:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=search&t=fast+cars&xml_standard=1
```

```
http://api.domainplex.com/?a=get&api_key=KEY&d=search&t=%22fast+cars%22&xml_standard=1
```

```
http://api.domainplex.com/?a=get&api_key=KEY&d=search&t=[fast+cars]&xml_standard=1
```

If a record does not exist in cache, question marks are returned. A search volume of -1 indicates that the record was found in the database but has a search volume of < 10 searches per month associated with it.

**Important:** This API call does not support post-back or `release_socket`. All results are pulled from cache and no live queries are performed. Results are provided instantly.

## Sending Whois Queries

The API comes equipped with a powerful bulk whois tool capable of querying whois for many domain extensions as well as parsing out the vital whois information such as registration status, registrar, creation and expiration date, registrant, emails and much more.

The API also supports checking IP addresses and determining IP owners as well as geo-tracking them to a specific country / state / city.

Whois queries are generally divided into two separate queries (the API treats them as a single query however):

- **Registry Query** – Extracts registry level information such as creation date, expiration date, update date, registrar, name servers (com/net only)
- **Registrar Query** – Extracts registrant information such as contact details, emails and more.

Depending on what you need, and the type of domains you are scanning, you may be satisfied with the much faster registry level check (this only applies to .com/.net domains). To perform a registry level query add `&registry_only=1` to the end of your whois API call. Otherwise, a full registry and registrar level lookup will be performed.

The `registry_only=1` parameter is ignored on non com/net domains (ex: org, info, de, etc).

Querying the whois API is easy. Simply prepare one or more domains or IPs and issue the query. See example query below for `example.com` domain.

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois&t=example.com&xml_standard=1
```

API Whois query for `example.com` and `test.com` with `release_socket` indicating that results will be picked up later by using the `session_download` command:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois&t=example.com>>test.com&xml_standard=1&release_socket=1
```

The above two commands will perform a full whois query to extract the registry and registrar data for all domains.

**IMPORTANT:** This API call supports post-back and `release_socket`. This means that you can submit large lists and download the results when they are finished by applying the `&release_socket=1` parameter to the end of the API call. Additionally, by providing a post-back URL (see section “[Creating Post-back Notifications](#)”) you can have the results submitted back to your server when they are finished processing.

## Sending Lead Generator Queries

The lead generator is one of the most unique tools offered by the DomainPlex API. It is capable of generating end-user leads for any domain name, or a list of domains. It was designed to help locate enduser buyers, usually small businesses, for domains priced between \$150 and \$5,000. Our experience indicates that the highest sell through rate occurs at the \$250-\$350 range.

**It’s important to note that this tool is very resource intensive.** It is also slower than the other tools due to the amount of processing required to generate leads. As a result, **all lead generator queries are assumed to be socket\_release**, meaning that results will not be provided back with the API call but must be picked up later (1-7 minute processing time) or will be posted back to your server upon completion. You can check the status of an active lead generator session by using the `session_progress` API call (see “[Checking Queue Progress](#)” section).

Here is a quick overview of the steps taken to generate a set of leads for a single domain:

- **Zone Check** – The zone contains over 100 million domain names. The source domain is checked against the zone to find similar domain names which may be potential leads (ex: domains with hyphens that have the same keywords, or domains in inferior extensions).
- **Site Crawl** – The zone check method generates a wide range of a few hundred to a few thousand potential leads. All these leads must be crawled to ensure that the sites resolve and are not parked. Crawling thousands of sites in just a few minutes is very bandwidth intensive.
- **Whois Check** – Once potential leads have been identified, it’s necessary to perform whois checks to extract contact information such as emails and phone numbers. Leads associated with other domainers and domain holding companies are eliminated in this step.

As you can see, the process is very resource intensive but is still a lot more efficient then attempting to perform the same operation by hand or using multiple tools and services.

Consequently, API calls to this tool are limited and access is granted on an individual basis.

The lead generator has three modes for generating leads which can be specified by using the optional `&o=MODE` parameter.

- **Search** – Generates leads based on search engine results for the keyword within a domain name. For example, if your domain was “cheapcarinsurancequotes.com”, then the search approach will pull up a list of all sites that come up for “cheap car insurance quotes” at the various search engines.
- **Ads** – Generates leads based on sites that advertise for the domain keyword. For example, if your domain was “cheapcarinsurancequotes.com”, then the ad approach will pull up a list of all advertisers that advertise for “cheap car insurance quotes” (and related keywords) at the various search engines. Since these advertisers already pay for clicks on similar keywords, they make ideal candidates for the domain matching the exact keyword they advertise for. We automatically remove wildcard advertisers such as eBay and many others that would not be interested in buying domains.
- **DNS** – Generates leads based on similar registered and used domains. For example, if your domain was “cheapcarinsurancequotes.com” and someone owned and used “cheap-car-insurancequotes.com” or “cheapcarinsurancequotes.net”, then they are potential lead since in both cases your domain name is of higher quality. This approach usually generates hundreds or thousands of leads so they must all be crawled to narrow

down only those domains that are used by businesses.

- **All** – This is the default method for generating leads. It will combine the above methods into a single report and include only the leads with the highest probability of success.

Initiating a lead generator API call is easy. If you want to generate leads using all methods, simply use the following API call (remember, results will not be provided right away so you may want to specify a postback URL or download the results later).

```
http://api.domainplex.com/?a=get&api_key=KEY&d=leads&t=cheapcarinsurance.com&o=all&xml_standard=1
```

To pull DNS based leads use the following URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=leads&t=cheapcarinsurance.com&o=dns&xml_standard=1
```

To pull Search based leads use the following URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=leads&t=cheapcarinsurance.com&o=search&xml_standard=1
```

The following results are returned in standard XML format:

- **source\_domain** – The domain for which the leads are being generated.
- **lead\_domain** – The domain name associated with the lead.
- **lead\_reason** – The reason the specified lead was chosen (ex: domain is similar to source domain).
- **lead\_type** – The type of lead generated (DNS, ad, search).
- **lead\_source** – The source of the lead (if ad or search this will be the search engine used), otherwise left blank.
- **lead\_keyword** – The keyword used to find this lead (DNS, search and ad).
- **site\_title** – The title of the site that is associated with the specific lead.
- **meta\_keywords** – Meta keywords extracted from the site associated with the specific lead.
- **meta\_description** – Meta description extracted from the site associated with the specific lead.
- **ip** – The IP address the lead site is hosted on.
- **ip\_domains** – The number of domains hosted on the site's IP address.
- **whois\_registrant** – The registrant associated with the specified lead's domain, based on the whois record.
- **whois\_email** – The emails associated with the specified lead's domain, based on the whois record.
- **whois\_phone** – The phone numbers associated with the specified lead's domain, based on the whois record.
- **whois\_age** – The age of the domain name associated with the lead.
- **whois\_registrar** – The registrar associated with the specified lead's domain.
- **alexa\_traffic\_rank** – The Alexa traffic rank for the specified lead's domain.
- **alexa\_trend** – The Alexa traffic trend for the specified lead's domain.
- **alexa\_link\_popularity** – The Alexa link popularity for the specified lead's domain.
- **pagerank** – The Google PageRank associated with the specified lead's domain.
- **ad\_keyword** – Advertiser keyword used to find this lead (for ad leads).
- **ad\_search\_volume** – The average search volume for the ad keyword associated with this lead.
- **ad\_cpc** – The average CPC for the ad keyword associated with this lead.
- **ad\_daily\_cost** – The approximate daily ad cost associated with purchasing keyword advertisements for the specified lead.
- **ad\_monthly\_cost** – The approximate monthly ad cost associated with purchasing keyword advertisements for the specified lead.

**IMPORTANT:** This API call **requires** `release_socket` (it is assumed so you do not need to specify it in the URL). This means that results are never sent back with the API response. The processing time is usually 1-7 minutes. You can use the `session_progress` API call to check active lead generator session progress and download the results using

the `session_download` API call. Additionally, by providing a post-back URL (see section “[Creating Post-back Notifications](#)”) you can have the results submitted back to your server when they are finished processing.

---

## Sending Trademark Check Queries

The API comes equipped with a simple trademark analyzer and risk assessment tool.

Our trademark tool does NOT serve as a comprehensive trademark check. Please do not base any legal decision on the results generated by the trademark tool. It is provided to help you identify high risk domains but should never be relied on exclusively. Always seek legal advice for a professional risk analysis.

The trademark checker accepts a domain or a list of domains (separated using the standard `>>` delimiter) along with the strict match identifier (numeric).

Two types of matching algorithms are used to identify trademarks.

- **Exact Matching** – This algorithm uses exact matching with word based intelligence. This simply means that if a domain contains the exact trademark and the trademark does not overlap with other words (ex: `registrarfordomains.com` contains a trademark for “ford” but it really means “registrar for domains”, so it fails the word intelligence test), then a match is made.
- **Typo Matching** – Typo matching uses a letter probability algorithm to determine if a domain is a potential typo of a trademark. The probability thresholds can be set by the user by specifying a `&strict_level=X` parameter with the API call, replacing X with a numeric value between 10 and 100. The `strict_level` parameter determines how close the typo must come (percentage wise) to the actual trademark in order to consider it a trademark match. A value of 10 would make the algorithm extremely loose, while a value of 100 would require an exact match (ex: `MicrosoftGames.com => Microsoft™`). We strongly recommend a value of 65 (default) for this parameter.

Please keep in mind that setting a low `strict_level` parameter will result in many false-positives, or domains matched with trademarks not likely to be a trademark violation.

The following sample URL can be used to identify trademark violations for “mikrosoft.com”:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=trademark&t=mikrosoft.com
```

All results are provided using standard XML format, one domain per line and include the following properties:

- **Domain** – The actual domain queried for trademark analysis
- **Type** – The type of trademark found, if any. Valid fields include “exact” for exact matches, “typo” for typo matches and a blank field if no trademark match was made.
- **Word** – The actual trademark found (if any).
- **Company** – The name of the company that owns the trademark (if any).
- **Risk** – The risk associated with the particular trademark (0-10, with 10 being the highest risk). Some trademarks carry a lower risk while others a much higher risk. For example, a Wikipedia trademark carries a low risk while a Microsoft trademarks carries a high risk.
- **Score** – The probability that this is the matching trademark. If the probability falls below the `strict_level` setting, the domain is not considered a trademark.

In some cases, it may be counter-productive to perform typo trademark matches. Users who simply want to be alerted of exact trademark violations (ex: `MicrosoftGames.com => Microsoft`), should set the `strict_level` parameter to 100.

**IMPORTANT:** This API call supports post-back and `release_socket`. This means that you can submit large lists



and download the results when they are finished by applying the `&release_socket=1` parameter to the end of the API call. Additionally, by providing a post-back URL (see section “[Creating Post-back Notifications](#)”) you can have the results submitted back to your server when they are finished processing.

---

## Sending Word Parser / Language Detection Queries

The API features a powerful and extremely fast word parser that is capable of automatically detecting the language based on the words used in the domain.

At the time of this writing, the following languages are supported:

- English – Full support
- Polish – Full support
- Spanish – Good support, improving dictionary file
- German – Basic support, improving dictionary file
- French – Basic support, improving dictionary file

The word parser has two modes of operation. The default mode will use the English dictionary file **only** and will not perform language analysis. This mode is twice as fast as the language parsing mode. The language parsing mode is turned on by applying the `&multi_language=1` flag to the end of the API call.

Sample URL to query “fastcars.com” for words in the English dictionary only:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=parser&t=fastcars.com&xml_standard=1
```

A standard (English-only) word parsing query returns the following parameters via XML:

- Term – The original domain queried
- Words – The parsed words

Sample URL to query “fastcars.com” for words in all supported languages and return language details:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=parser&t=fastcars.com&xml_standard=1&multi_language=1
```

The multi-language word parsing query returns the following parameters via XML:

- Term – The original domain queried
- Words – The parsed words
- Language – The detected language based on the words
- Probability – The probability that this is the language

**IMPORTANT:** This API call supports post-back and `release_socket`. This means that you can submit large lists and download the results when they are finished by applying the `&release_socket=1` parameter to the end of the API call. Additionally, by providing a post-back URL (see section “[Creating Post-back Notifications](#)”) you can have the results submitted back to your server when they are finished processing.

---

## Sending DNS Domain Availability Queries

You can instantly check domain availability for a single domain or thousands of domains by using the DomainPlex API.

The API is also capable of generating multi-extension matches for a single domain query (ex: fastcars.com => fastcars.com, fastcars.net, fastcars.org, etc).

The DNS data comes from multiple registries and is updated daily. As a result, the DNS checker cannot be relied upon for the latest availability information (for example, it should not be used to monitor domain availability). Domains registered or deleted on the day of the query may not show up in the results or may show an invalid status.

For **live** availability queries we recommend using the slower live availability checker (see the next section). At the time of this writing, the following extensions are supported by the DNS availability checker:

- com
- net
- org
- info
- us
- biz
- mobi
- xxx
- Most nTLDs (see information regarding [retrieving a list of supported extensions](#))

Since the DNS availability mode is based on domains which exist in the daily zone file, domains which are in PendingDelete, Registrar-Hold or RedemptionPeriod status will show an invalid status of “available”. Furthermore, domains with no name servers (< 0.5% of all registrations) will show an available status as well even though they are not available.

The DNS checker can be queried in two ways. The primary mode performs a simple zone lookup based on the domain entered. The secondary mode will query the specified domains in all supported extensions (by supplying the `&multi_tld=1` parameter).

Sample URL to query “fastcars.com” for availability:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=zone&t=fastcars.com&xml_standard=1
```

The following results are returned:

- Domain – The original domain queried
- Status – The registration status: registered / available / unknown

If you wanted to check fastcars.com and all other supported extensions in one query you can do so by appending a `&multi_tld=1` parameter to the end of the API, for example:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=zone&t=fastcars.com&xml_standard=1&multi_tld=1
```

**IMPORTANT:** This API call processes data instantly and does not allow post-back or release\_socket queries. All queries are processed live and are not stored in the cache.

---

## Sending Registry Level Domain Availability Queries

Although much slower than the DNS level check, the live availability checker is capable of checking domain availability at the registry. This API call should only be used when the live availability of a domain must be known (ex: newly registered domains or newly dropped domains).

Please note that this API call cannot be used to drop catch or monitor domain availability. We have placed time limits on the queries for this reason. For faster responses and bulk scans we strongly recommend using the DNS availability checker (see the previous section).

At the time of this writing, the following extensions are supported by the DNS availability checker:

- com
- net
- org
- info
- us
- biz
- mobi
- xxx
- tel
- de
- co.uk
- Most nTLDs (see information regarding [retrieving a list of supported extensions](#))

The API call performs registry level availability checks, meaning that if the domain shows up as available, it is guaranteed to be available by the registry.

Sample URL to query “fastcars.com” for live availability:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=availability&o=availability&t=fastcars.com&xml_standard=1
```

The following results are returned:

- Domain – The original domain queried
- Status – The registration status: registered / available / unknown / error

**IMPORTANT:** This API call processes data instantly and does not allow post-back or release\_socket queries. All queries are processed live and are not stored in the cache. This API call cannot be used to backorder domains or monitor domain availability.

---

## Sending Sold History Search Queries

Our servers constantly crawl various domain sales aggregator sites and store the sales in a centralized database accessible and searchable by all API users.

The API allows for two search modes:

- **Simple Search Mode** – Instant results for domains matching the following criteria:
  - Exact Domains (&o=exact\_domains)
  - Exact Keywords (&o=exact\_keywords)
  - Flexible Domains (&o=flexible\_domains)
  - Flexible Keywords (&o=flexible\_domains)
- **Advanced Search Mode** – Instant results for domains matching any of the following criteria (&o=advanced must be supplied with the URL):
  - Keywords (&t parameter)
  - Extension (&extension parameter)

- Minimum Price (&min\_price parameter)
- Maximum Price (&max\_price parameter)
- Sold Date Range Start (&sold\_start parameter, format: YYYY-mm-dd)
- Sold Date Range End (&sold\_end parameter, format: YYYY-mm-dd)
- Minimum SLD Length (&min\_length parameter)
- Maximum SLD Length (&max\_length parameter)
- Sales Source (&source parameter)

Keywords or domains are not required. So, you can perform a search on a date range (ex: domains sold this month) to see a complete list of recent sales.

Results are always sorted by the most recent sales date. Please note that we show a maximum of 500 results per query.

Sample URL to query “cars.com” for exact domain match:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=sold_history&t=cars.com&o=exact_domains&xml_standard=1
```

Sample URL to query “cars” for exact SLD match:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=sold_history&t=cars&o=exact_keywords&xml_standard=1
```

Sample URL to query “cars” for wildcard keyword match:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=sold_history&t=cars&o=flexible_keywords&xml_standard=1
```

Sample URL to query domains sold between 2009-11-01 and 2009-11-30:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=sold_history&o=advanced/d_start=2009-11-01/d_end=2009-11-30&xml_standard=1
```

Sample URL to query domains in the .COM extension sold between \$100 and \$500:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=sold_history&o=advanced&min_price=100&max_price=500&extension=com&xml_standard=1
```

A standard query results in the following information being returned:

- term – The original domain or keyword queried
- domain name – The domain name sold
- sold\_date – The domain sales date
- price – The domain sales price
- source – The domain sales source (ex: Afternic, TRAFFIC)
- mediator – The sales mediator (only available on some sales)

**IMPORTANT:** This API call processes data instantly and does not allow post-back or release\_socket queries. All queries are processed live and are not stored in the cache.

## Sending Reverse IP Lookup Queries

The API is capable of performing instant reverse IP lookups. This means that you can specify an IP address and instantly see the domains that are hosted on it.

All data is pulled from our snap shot database of 110 million web sites. We crawl the entire web on a monthly basis. The results may be delayed by up to 30 days.

The reverse IP API functions in two modes (specified by the `&o=` parameter):

- **domains** – Returns a list of actual domains hosted on a particular IP
- **count** – Returns a count of domains hosted on a particular IP

Sample URL to see the number of domains hosted on “67.217.35.18”

```
http://api.domainplex.com/?a=get&api_key=KEY&d=reverse_ip&t=67.217.35.18&o=count&xml_standard=1
```

Sample URL to see domains hosted on “67.217.35.18”

```
http://api.domainplex.com/?a=get&api_key=KEY&d=reverse_ip&t=67.217.35.18&o=domains&xml_standard=1
```

**IMPORTANT:** This API call processes data instantly and does not allow post-back or release\_socket queries. All queries are pulled from cache and our database does not include results for all possible keywords.

---

## Sending SERP Queries

The API allows for pulling search results for specified keywords if they are already stored in our database.

The SERP API (SERP stands for Search Engine Results Page) does **not** perform a live lookup but relies on the cached data stored on our servers. Data quality is not guaranteed but it may be useful for various tools.

Only keywords can be used on this API call. Domains are not supported; neither is parsing words from domains. Please be advised that we only store “exact” search results, meaning that you have to put quotes around the keywords you submit (ex: “fast cars” and not fast cars).

SERP functions in two modes (specified by the `&o=` parameter):

- **search** – Returns search results for a given term
- **ads** – Returns ads for a given term

The following results are returned with the SERP API call:

- keyword – The original keyword queried
- source – The search engine from which the data is provided (this cannot be set)
- title – The title of the site in the results
- description – The description of the site
- url – The URL of the site (if it is not an ad lookup)
- domain – The domain of the site

Sample query to show the search results that come up on the search engines for “fast+cars”:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=serp&t="fast+cars"
```

Sample query to show the ad results that come up on the search engines for “fast+cars”:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=serp&t="fast+cars"&o=ads
```

**IMPORTANT:** This API call processes data instantly and does not allow post-back or release\_socket queries. All queries are pulled from cache and our database does not include results for all possible keywords.

## Downloading Domain Lists with the API

Many different domain name lists are available for download via the DomainPlex API in a standardized format. You no longer have to worry about loading lists from various sources and having to parse out specific data (such as bids from NameJet auctions, or drop dates from PendingDelete lists). All of our lists are updated at least once per day, with many lists being updated throughout the day (such as Available and Aftermarket auction lists).

Downloadable lists are divided into three categories:

- **Raw Domain Lists** – Lists containing domain names and information specific to the list type (ex: for PendingDelete lists, this would include the domain and drop date, for aftermarket auction lists, this would include the domain, auction start and end times, bids, links, etc). Data is provided via standardized XML format and generally updated throughout the day
- **Processed Domain Lists** – Raw domain lists scanned for metrics such as link popularity, term popularity, registrar, value, and much more.
- **Zone Files** – Both raw and clean zone files are available for download once per day for .com/.net/.org/.info/.us/.biz/.mobi/.xxx and nTLD extensions.

The following lists are available for download using both raw and processed methods:

- **Available** – Domains which became available today
  - Domain
  - &o=available&list\_date=YYYY-mm-dd
- **PendingDelete Domains** - Domains dropping within the next 6 days
  - Domain
  - Drop Date
  - &o=pendingdelete&list\_date=YYYY-mm-dd
- **Registrar-Hold** – Domains which have just expired and will drop within 30-120 days
  - Domain
  - Expire Date
  - Optional Registrar
  - Optional Creation Date
  - Optional Name servers
  - Optional Emails
  - &o=registrar-hold&list\_date=YYYY-mm-dd
- **Redemption Period** – Domains which have recently expired and will drop within 45-60 days
  - Domain
  - Expire Date
  - Optional Registrar
  - Optional Creation Date
  - Optional Name servers
  - Optional Emails
  - &o=redemptionperiod&list\_date=YYYY-mm-dd

- **SnapNames Exclusive Auctions**
  - Domain
  - Auction Join By Date
  - Auction End Date
  - Auction Price
  - Auction Bidders
  - Auction URL
  - `&o=snapnames+exclusive+drops&list_date=YYYY-mm-dd`
- **NameJet Pre-Release Auctions**
  - Domain
  - Auction Join By Date
  - Auction Price
  - Auction URL
  - `&o=namejet+pre-release&list_date=YYYY-mm-dd`
- **GoDaddy Auctions**
  - Domain
  - Auction End Date
  - Auction Price
  - Auction Bidders
  - Auction URL
  - Domain Traffic
  - Domain Revenue
  - Auction Title
  - Auction Description
  - `&o=godaddy+auctions&list_date=YYYY-mm-dd`
- **New Registrations** – Domains registered / added to the zone on the specified date
  - Domain
  - Optional Registrar
  - Optional Creation Date
  - Optional Expiration Date
  - Optional Name servers
  - `&o=raw_file&list_type=new_registrations&list_date=YYYY-mm-dd`
  - `&o=raw_file&list_type=new_registrations_registry&list_date=YYYY-mm-dd`
- **New Deletions** – Domains deleted from the zone on the specified date
  - Domain
  - `&o=raw_file&list_type=new_deletions&list_date=YYYY-mm-dd`
- **Expiring Soon** – Domains which are set to expire on the specified date
  - Domain
  - Registrar
  - Creation Date
  - Expiration Date
  - Update date
  - Name Server
  - Email
  - `&o=expiring_soon&list_date=YYYY-mm-dd`

---

## Downloading Raw Domain Lists

Once you have decided which list you want to download you can issue the following API command:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=lists&o=LISTCATEGORY&list_type=LISTTYPE&list_date=YY
YY-mmdd
```

Necessary parameters:

- `&o=LISTCATEGORY` – You can find the list category in the section above. Ex: `&o=expiring_soon`
- `&list_date=YYYY-mm-dd` – This is the date for which the list was PROCESSED (does not mean drop date or auction end date, but rather processing date, or the day it was entered into our system). Ex:  
`&list_date=2009-11-19`
- `&list_type=LISTTYPE` – Some queries (such as new registrations and new deletions) require a list type. Please see above section for details.

## Downloading Processed Domain Lists

Downloading processed domain lists is just as easy as downloading raw lists. Please note that due to the value of these lists and the resources required to generate them, they are only available to select clients.

Once you have decided which list you want to download you can issue the following API command:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=lists&o=LISTCATEGORY&list_type=LISTTYPE&list_date=YY
YY-mmdd?ocessed=1
```

Necessary parameters:

- `&o=LISTCATEGORY` – You can find the list category in the section above. Ex: `&o=expiring_soon`
- `&list_date=YYYY-mm-dd` – This is the date for which the list was PROCESSED (does not mean drop date or auction end date, but rather processing date, or the day it was entered into our system). Ex:  
`&list_date=2009-11-19`
- `&list_type=LISTTYPE` – Some queries (such as new registrations and new deletions) require a list type. Please see above section for details.
- `&processed=1` – Tells the API to show processed results.

## Downloading Zone Files

The API allows for instant download of daily updated zone files. Zone files are updated at 7 AM US Central time.

Due to the amount of bandwidth required to upload zone files this feature is only available upon request. The following zone files are currently supported:

- com
- net
- org
- info
- us
- biz
- mobi
- xxx
- Most nTLDs (see information regarding [retrieving a list of supported extensions](#))



Two types of downloads are available for each of the supported extensions:

- Raw – Raw zone files as provided by the registry (includes domains and name servers). Unsorted and about 6 GB uncompressed for .com (~300m records, ~80m unique domains). Unless you have a specific need for domains and name servers with the resources to load and parse this file, you should not download this list.
- Processed – Sorted & unique domain names. Approximately ~80m domains. Duplicates removed.

Sample URL to download **raw** com zone file:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=lists&o=zone_files&zone=com
```

Sample URL to download **processed** com zone file:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=lists&o=zone_files&zone=com?ocessed=1
```

## Advanced API Calls

---

### Checking Session Progress

Almost all API calls result in a new session being created to handle various aspects of the call. Each session stores information related to the query such the list of domains submitted, session start time, session end time, session progress, session status and much more.

While some queries return results instantly, other queries (such as bulk appraisal queries) may take from a few minutes to a few hours to complete, hence the need for sessions.

You can check the status of **all of your sessions** at any time by calling the API command below:

```
http://api.domainplex.com/?a=session_progress&api_key=KEY
```

The above command will return your last 100 sessions and the details associated with those sessions. If you have more than 100 sessions in your history you can specify a starting point with a `&start` parameter (remember, all results are sorted by newest to oldest).

```
http://api.domainplex.com/?a=session_progress&api_key=KEY?t=5000
```

The following information is returned with the session data:

- **session name** – A unique 45 character name assigned to this session.
- **session user** – The user associated with this session.
- **status** – The status of this session (active, processing, completed, canceled).
- **type** – The type of session this is (ex: appraisal, whois, word parser, lead).
- **stage** – The stage of the session (ex: appraisal, whois, word parser, lead).
- **stage\_description** – Description of what is being done (ex: checking whois for test.com).
- **stage\_progress** – The progress of the stage expressed as a percentage.
- **overall\_progress** – The overall session progress expressed as a percentage.
- **last\_updated** – Date and time this session was last updated (US Central time zone).
- **session\_download** – If the session is completed, this is the link to download the results.

Some API calls hand off work to daemons that specialize in processing a certain task.

For example, an appraisal query will split the work load to a word parsing daemon (which parses words), as well as to a search engine daemon (for checking search engine stats), a whois daemon (for pulling domain age and other relevant data) and so on. The “stage”, “stage\_description” and “stage\_progress” refer to the actual work being performed while the “type” attribute specifies the type of the master session (ex: appraisal, lead, whois).

You can also check the status of a single session once you have its name (from the bulk session URL above). This is ideal for quickly checking session progress via AJAX while a user waits for the results.

The following query can be used to check individual session progress (replace sessionname with the name of the session from the bulk list):

```
http://api.domainplex.com/?a=session_progress&api_key=KEY&session_name=SESSIONNAME
```

## Stopping Queued Sessions

All bulk sessions in “Active” status can be stopped and the resources assigned to the user released.

To cancel a session, you must know the session name. See the section above titled “Checking Session Progress” to obtain a session name for a session you wish to cancel.

Once you have obtained the session name run the API call below to cancel the session:

```
http://api.domainplex.com/?a=session_cancel&api_key=KEY&session_name=SESSIONNAME
```

Please note that it may take up to 1 minute for the session to completely cancel and for the resources to be released. Canceled sessions will NEVER return any results. All results are automatically purged when a session is canceled.

---

## Downloading Completed Sessions

You can download session results for up to seven days starting from the session completion date for all sessions with a “Completed” status.

To download session results, you must know the session name. See the section above titled “Checking Session Progress” to obtain a session name for a session you wish to download.

Once you have obtained the session name run the API call below to download the session results:

```
http://api.domainplex.com/?a=session_download&api_key=KEY&session_name=SESSIONNAME
```

All sessions older than seven days are automatically purged for privacy reasons and to conserve disk space.

---

## Creating Simple Notifications

The API server can notify your server when a session you’ve initiated completes. This is called a “Simple” notification because no results are sent back, only a notification indicating that the session completed and you can download the results at your convenience.

The following tools support simple notifications:

- appraiser
- lead generator
- whois checker
- trademark checker
- word parser

A notification is a URL to which our server will send a POST request when the session completes. The following requirements must be met:

- http URLs only (no https)
- URL length must be less than 250 characters
- URL must be valid and start with http://

- URL cannot include additional parameters such as ? or & (direct URL only)

For example, if you wanted to have your server listen for notification updates using `myscript.php`, you would add the following parameter to your API calls (ex: bulk appraisal queries):

```
-ify_url=http://www.MYDOMAIN.com/myscript.php
```

Keep in mind that PHP and other programming languages may require that you encode the URL prior to sending it with the request (in PHP you would call the `urlencode` function on the actual URL you want to notify to).

Adding the above parameter to your API query will result in your server receiving a notification upon completion of the session.

The notification contains the following data (in PHP they are accessible using the `$_POST[ 'PARAMNAME' ]` variable, where `PARAMNAME` is one of parameters below):

- **api\_session** – The API session for which this notification is sent
- **api\_user** – The API user for which this notification is sent
- **api\_duration** – The session processing duration, in milliseconds

---

## Creating Post-back Notifications

In addition to sending simple notifications, the API server can also post-back the results of the session back to your server when the session completes via a post-back notification.

The following tools support Post-back notifications:

- appraiser
- lead generator
- whois checker
- trademark checker
- word parser

A post-back URL should point to a script on your server that will receive a POST request from our server with session details and session results. A post-back URL must meet the following requirements:

- http URLs only (no https)
- URL length must be less than 250 characters
- URL must be valid and start with `http://`
- Your script must be able to process POST methods
- Results must be under 50 megabytes
- URL cannot include additional parameters such as ? or & (direct URL only)

For example, if you wanted to have your server listen for notification updates using `myscript.php`, you would add the following parameter to your API calls (ex: bulk appraisal queries):

```
&postback_url=http://www.MYDOMAIN.com/myscript.php
```

Keep in mind that PHP and other programming languages may require that you encode the URL prior to sending it with the request (in PHP you would call the `urlencode` function on the actual URL you want to post back to).

Adding the above parameter to your API query will result in your server receiving a notification and results data upon completion of the session.

The post-back notification contains the following data parameters (in PHP they are accessible using the `$_POST [ 'PARAMNAME' ]` variable, where `PARAMNAME` is one of parameters below):

- `api_session` – The API session for which this notification is sent
- `api_user` – The API user for which this notification is sent
- `api_duration` – The session processing duration, in milliseconds
- `data` – The actual results of the session (ex: appraisal or whois results)

---

## Viewing Usage Statistics

You can view the query log for your account at any time by using an appropriate API call.

Please keep in mind that we increment query counts based on the total amount of data processed and not based on query count.

In other words, if you submitted an appraisal query with one domain, the query count would be incremented by one. However, if you sent an appraisal query with 100 domains then the query count would be incremented by 100 because 100 items will be *processed* in a single query.

The following query logs are available:

- Daily Global Query Summary – The total number of queries processed for all tools and services.
- Daily Tool Query Summary – The total number of queries processed for each tool and service.
- Daily IP Query Summary – The total number of queries processed for each of your client's remote IPs (if specified using the `&remote_ip` parameter).

**All logs are updated live with no delay.**

Sample URL to pull daily global query summary:

```
http://api.domainplex.com/?a=stats&api_key=KEY&o=global_summary
```

Sample URL to pull daily tool query summary:

```
http://api.domainplex.com/?a=stats&api_key=KEY&o=tool_summary
```

Sample URL to pull daily IP query summary:

```
http://api.domainplex.com/?a=stats&api_key=KEY&o=ip_summary
```

By adding a `&remote_ip=IPADDRESS` parameter to the end of API calls you'll be able to track your clients' usage and place limits based on query counts and IPs.

## API Command List

The `commands` listed below are those configured for your account.

**aftermarket** Aftermarket Domain Search.

Search various archived domain name aftermarket data for one or more domain names.

- `t` - one or more domain names, separated with `>>`

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=aftermarket&t=alkamshle.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <aftermarket domain="alkamshle.com" source="NameJet Pre-Release" action_date="2011-12-30
00:00:00" bid="69.00" bin="0.00" bidders="0" url="http://www.namejet.com/Pages/Auctions/
BackorderDetails.aspx?domainname=alkamshle.com" title="" mod_date="2011-12-11 03:36:44" />
</results>
```

**alexa** Check Cached Alexa Data.

Retrieve cached Alexa traffic data for one or more domains, including traffic rank, link popularity and other stats.

- `t` - one or more domains, separated with `>>`

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=alexa&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="utf-8"?>
<results>
  <alexa domain="example.com" link_popularity="35222" traffic_rank="9128" trend="6600"
owner="internet assigned numbers authority" email="res-dom@iana.org" city="marina del rey"
state="CA" country="US" language="EN" code="" created="1995-08-14"
phone="+13108239358,fax:+13108238649" />
</results>
```

**appraise** Generate Automated Appraisals and Extract Domain Analytics.

Process domain name appraisals and extract domain analytics. Returns appraised value and a wide variety of analytic data.

Please refer to the [Sending Appraisal Queries](#) portion of this document for complete details.

- `t` - one or more domains, separated with `>>`
- `efficient_mode=1` - optional, recommended - will query only the most vital metrics necessary for an appraisal, while pulling archived non-vital metrics from the database (e.g. PageRank).
- `fast_mode=1` - optional - will appraise domains faster, without querying live metrics.
- `no_seo=1` - optional - will not provide extra SEO data (such as Alexa, Pagerank, etc), thus improving the response time.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=appraise&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <appraisal domain="example.com">
    <domain_cc>Example.com</domain_cc>
    <sld>example</sld>
    <tld>com</tld>
    <words>example</words>
    <language>english</language>
    <language_probability>100.0</language_probability>
    <category>Other -- Generic Single Word</category>
    <appraised_value>325000</appraised_value>
    ...
  </appraisal>
</results>
```

---

**appraise\_cache**    Cached appraisals.

Example URL:

```
http://api.domainplex.com/?a=get&u=%username%&p=%password%&d=
```

---

**appraise\_history**    Access historical appraisal data for any domain.

Access historical appraisal data for any domain

Example URL:

```
http://api.domainplex.com/?a=get&u=%username%&p=%password%&d=
```

---

**archive**    Check Cached Domain Metrics.

Retrieve cached metrics for one or more domain names, including Alexa data, keyword search rank and link popularity, Wayback stats and past auction details.

- t - one or more domains, separated with >>

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=archive&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <domain>example.com</domain>
  <alexa>
    <traffic_rank>9128</traffic_rank>
    <link_popularity>35222</link_popularity>
    <trend>6600</trend>
    ...
  </alexa>
  <google>
    <link_popularity>2300</link_popularity>
    <link_saturation?</link_saturation>
    <sld_term_popularity>0</sld_term_popularity>
    <sld_ads>1</sld_ads>
    <keyword_term_popularity>0</keyword_term_popularity>
    <keyword_ads?</keyword_ads>
  </google>
  ...
</result>
```

**availability** Check Live Domain Availability.

Perform live availability checks on one or more domain names to determine if they are available for registration.

Slower than the dns availability service but can detect domains that are registered but not resolving in DNS. See the [Sending Registry-Level Domain Availability Queries](#) section for complete details.

- t - one or more domains, separated with >>

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=availability&o=availability&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <domain>example.com</domain>
  <status>registered</status>
</result>
```

**backlinks** Domain Backlinks Report.



Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=backlinks&t=msn.com&xml_standard=1
```

---

**backlinks\_count** Domain Backlinks Count.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=backlinks_count&t=msn.com&xml_standard=1
```

---

**backlinks\_raw** Domain Backlinks Raw.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=backlinks_raw&t=msn.com&xml_standard=1
```

---

**backorder\_bid\_buy** Backorder Bid Buy Domains at major drop catchers.

Backorder, bid or purchase domain names at an auction provider.

This is an advanced tool. Please contact us for details regarding implementation.

---

**category** Domain Name Categorization System.

Analyze one or more domain names and attempt to categorize them into the DomainPlex category tree.

- t - one or more domain names, separated with >>

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=category&t=fastcars.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <term domain="fastcars.com" category_root="Vehicles -- Automobiles" category_sub="Other --
Common Adjectives" />
</results>
```

---

**custom\_droplist** Retrieve Custom Droplists.

Retrieve a custom daily droplist for the specified TLD.

Example URL:

```
http://api.domainplex.com/?a=get&u=%username%&p=%password%&d=custom_droplist&tld=tld
```

### **dns** Check DNS Based Domain Availability.

Quickly check to see if one or more domain names are registered based on DNS lookup. Faster than a live availability lookup and can be used on large quantities of domain names to pre-screen for availability. Please see the [Sending DNS Domain Availability Queries](#) section for complete details.

- **t** - one or more domains, separated with >>

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=dns&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <domain>example.com</domain>
  <status>registered</status>
</result>
```

### **generate** Domain Name Generator.

Accept a list of keywords and a list of extensions and produce a list of all possible combinations of the two.

- **t** - one or more keywords, separated with >>
- **e** - one or more TLDs, beginning with ., separated with >>
- **o=simple**

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=generate&t=example>>first>>second&e=.com>>.net>>.org&o=simple&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="utf-8"?>
<results>
  <domain name="example.com" \>
  <domain name="example.net" \>
  <domain name="example.org" \>
  <domain name="first.com" \>
  <domain name="first.net" \>
  <domain name="first.org" \>
```

```
<domain name="second.com" \>
<domain name="second.net" \>
<domain name="second.org" \>
</results>
```

## google\_trend

 Shows google trends.

Retrieve Google search trends for the specified keywords.

- `t` - one or more keywords, separated with `>>`
- `all=1` - optional - perform broad keyword matching instead of exact.
- `json=1` - optional - return results in JSON format. If omitted, results will be returned in a serialized PHP array.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=google_trend&t=science
```

## ideas

 Generate Domain Ideas from Search Keywords.

Produces a list of keyword phrases based on one or more supplied keywords. Includes search CPC and other statistics for each generated keyword.

- `t` - one or more keywords, separated with `>>`

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=ideas&t=example&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <term>raoults law examples</term>
  <search_cpc>0.00</search_cpc>
  <gsr>277</gsr>
  <ads>0</ads>
  <overture>0</overture>
  <word_tracker>0</word_tracker>
</result>
...
```

## leads

 Generate End-User Leads for Domains.

Given one or more domain names, produce a list of similar domains with ownership information, analytic data and more.

See the [Sending Lead Generator Queries](#) section for complete details.

- `t` - one or more domain names, separated with `>>`
- `o=MODE` - optional - specify the lead generation mode. Valid modes include:
  - `search` - search engine results for the keyword within a domain name.
  - `ads` - sites that advertise for the domain keyword.
  - `dns` - similar registered and used domains.
  - `all` (default) - all of the above combined.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=leads&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <result source_domain="Example.com" lead_domain="trythisforexample.com" lead_reason="Searched
under a related keyword" lead_type="search" lead_source="google" lead_keyword="example" ... />
  <result source_domain="Example.com" lead_domain="analysisexample.com" lead_reason="Searched
under a related keyword" lead_type="search" lead_source="msn" lead_keyword="example" ... />
  <result source_domain="Example.com" lead_domain="exampledepot.com" lead_reason="Searched
under a related keyword" lead_type="search" lead_source="msn" lead_keyword="example" ... />
  ...
</results>
```

## **leads\_quick** Quick Lead Generator.

Quickly generate leads for one or more domain names. Uses static data and returns less analytic data, but may be faster than the `leads` tool as a result.

See the [Sending Lead Generator Queries](#) section for complete details.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=leads_quick&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <domain name="EXAMPLE.ORG" type="Similar Domain" ... registrant_domains="66"
email_domains="148" />
  <domain name="PERSONALEXAMPLE.COM" type="Similar Domain" ... registrant_domains="661978"
email_domains="713830" />
  <domain name="ERLANGEXAMPLES.COM" type="Similar Domain" ... registrant_domains="2"
email_domains="2" />
  ...
</results>
```

## **leads\_score** Calculate lead score given a domain name. Custom tool.

Example URL:

```
http://api.domainplex.com/?a=get&u=%username%&p=%password%&d=
```

---

## **lists** Download Domain Lists & Zone Files.

Many different domain name lists are available for download via the DomainPlex API in a standardized format.

For complete details, see the [Downloading Domain Lists with the API](#) section.

Parameters:

- `o` - the type of list to retrieve; see the section linked above.
- `list_date` - date for data retrieval in `yyyy-mm-dd` format.
- `list_type` - only required for `raw_file` mode.
- `processed` - return processed results instead of raw results.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=lists&o=hold&list_date=2009-10-31&xml_standard=1
```

---

## **monitor** Monitor domains, registrants, DNS records.

Functions for processing domain, DNS, registrant, email and keyword monitoring.

While it is possible to make advanced calls using this API tool, we recommend using one of our other services for monitoring and simpler API integration, such as [domainIQ](#). Please contact us for details regarding a domainIQ account or for specific details regarding monitoring queries directly through the DomainPlex API.

---

## **moz** Check moz domain authority score.

Check moz domain authority score.

Example URL:

```
http://api.domainplex.com/?a=get&u=%username%&p=%password%&d=
```

---

## **overture** Check Cached Overture Data.

Fetch archived Overture keyword data for the supplied domain names or keywords.

- `t` - one or more domain names or keywords. Domain names will be converted to keywords (e.g. "sample.com" becomes "sample com").

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=overture&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <overture term="example.com" search_volume="1466" />
</results>
```

---

**pagerank**    Pagerank Check.

Fetch Pagerank data for the specified domain(s).

- t - one or more domain names, separated with >>

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=pagerank&t=msn.com
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <pagerank domain="msn.com" pr="8" verified_pr="8" site_language="en" redirect="No" />
</results>
```

---

**pagerank\_history**    Pagerank History Check.

Fetch historical Pagerank data for the specified domain.

- t - one or more domain names, separated with >>

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=pagerank_history&t=msn.com
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <pagerank_history id="41950303" domain="info.com" pr="6" verified_pr="6" ... />
  <pagerank_history id="133787624" domain="info.com" pr="5" verified_pr="5" ... />
  <pagerank_history id="185673863" domain="info.com" pr="5" verified_pr="5" ... />
  ...
</results>
```

**parser** Parse Words from Domains and Identify Domain Language.

Splits domain names into words and attempts to detect domain name language.

See the [Sending Word Parser / Language Detection Queries](#) section for complete details.

- `t` - one or more domain names, separated with `>>`
- `multi_language=1` - optional - if specified, attempt to parse the language of the domain name(s). If omitted, only the English dictionary will be used and language analysis will be skipped.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=parser&t=example.com&multi_language=1&xml_standard=1
```

**pendingdelete\_raw** Raw PendingDelete file with drop order.

Fetch the latest raw pendingdelete list.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=pendingdelete_raw
```

**resolve** Crawl Domain, Parse Data and Resolve IP.

Crawl the provided domain name(s) and return site IP, HTTP status code, page title, number of links and more data.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=resolve&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <resolve domain="example.com">
    <ip>93.184.216.34</ip>
    <ip_count>1</ip_count>
    <status_code>200</status_code>
    <title_xml>Example Domain</title_xml>
    <meta_keywords />
    <resolve_time>317</resolve_time>
    <page_size>1269</page_size>
    <links>1</links>
    <images>0</images>
    ...
  </resolve>
</results>
```

**reverse\_adsense\_analytics** Perform reverse Analytics / AdSense lookups.

Look up domains using a Google AdSense or Analytics ID. Return either the count of domains associated with the given ID, or a list of domains.

- `type` - the type of lookup to perform. Values include `adsense` or `analytics`. Required.
- `t` - an Analytics or AdSense ID (typically beginning with CA- or UA-).
- `count=1` - optional - if supplied, return a count of the domains associated with the provided ID. If omitted, a list of domains is returned.

Returns results in XML format.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=reverse_adsense_analytics&type=analytics&t=UA-123123&count=1
```

Example Output:

```
<?xml version="1.0" encoding="utf-8"?>
<results>
  <reverse_adsense_analytics type="analytics" id="UA-123123" count="8005" />
</results>
```

**reverse\_dns** Reverse DNS lookup.

Perform a reverse DNS lookup to find domains hosted on the specified nameserver.

- `t` - one or more nameserver domains, separated with `>>`.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=reverse_dns&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <reverse_dns root_dns="example.com." domain="abc.guru" parked="0" />
  <reverse_dns root_dns="example.com." domain="andreearetstdns3.camera" parked="0" />
  <reverse_dns root_dns="example.com." domain="anlass.mobi" parked="0" />
  <reverse_dns root_dns="example.com." domain="anthonyeden.sexy" parked="0" />
  <reverse_dns root_dns="example.com." domain="any.clothing" parked="0" />
  <reverse_dns root_dns="example.com." domain="austinfraley.us" parked="0" />
</results>
```

**reverse\_ip** Reverse IP Lookup.

Look up domain names that are pointed at the specified IP address.



- t - one or more IP addresses.
- o=domains - optional - if provided, return a list of domains on the IP address. Otherwise, just a summary will be returned.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=reverse_ip&t=74.125.45.100&o=domains&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="utf-8"?>
<results>
  <reverse_ip ip="74.125.45.100" domain="92288.net" />
  <reverse_ip ip="74.125.45.100" domain="battarville.com" />
  <reverse_ip ip="74.125.45.100" domain="bmonkey.com" />
  ...
</results>
```

---

**reverse\_whois\_domain** Reverse whois for domain and email.

Examine a domain name and find the number of other domains associated with it by email.

- t - one or more domain names or email addresses, separated by >>
- type - specify either domain or email, depending on what is supplied for t.
- 'whois' - specify 0 or 1 for whether or not you wish to show whois data such as registrar, registrant, contact details, etc.

Returns results in JSON format.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=reverse_whois_domain&t=sample.com&type=domain
```

Example Output:

```
[
  {
    "item": "sample.com",
    "type": "domain",
    "domains_found": "1",
    "domains": [
      "sample.com"
    ],
    "email": "njrybtcnj@whoisprivacyprotect.com"
  }
]
```

---

**search** Perform Search Volume & CPC Lookups.

Retrieve statistics for one or more keywords, including global search volume, average CPC and ad competition level.

See the [Sending Keyword Queries](#) section for complete details.

- `t` - one or more keywords, separated with `>>`. Domains will be converted to keyword phrases if provided (e.g. `example.com` becomes `example com`).

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=search&t=example&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <keyword value="example" type="broad">
    <global_search_volume>25500000</global_search_volume>
    <avg_cpc>0.9523</avg_cpc>
    <ad_competition>low</ad_competition>
  </keyword>
</results>
```

---

## search\_volume

Look up search volume data for the specified keywords.

- `t` - one or more keywords, separated with `>>`
- `type` - optional - can be `exact` (default), `broad` or `phrase`.

Returns results in JSON format.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=search_volume&t=science&type=exact
```

Example Output:

```
[
  {
    "keyword": "[science]",
    "avg_search": "450000",
    "avg_cpc": "1.220",
    "avg_competition": "7"
  }
]
```

---

## serp

**Cached SERP Lookups.**

Fetch archived SERP (search engine result page) data for the supplied domain names or keywords.

See the [Sending SERP Queries](#) section for complete details.

- `t` - keyword to look up.
- `o=search` or `o=ads` - return search results for the given term, or ads for the given term.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=serp&t="fast+cars"&o=search&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <serp keyword="'fast cars'" source="google" ... domain="fastcars.net" />
  <serp keyword="'fast cars'" source="google" ... domain="popularmechanics.com" />
  <serp keyword="'fast cars'" source="google" ... domain="topspeed.com" />
  <serp keyword="'fast cars'" source="google" ... domain="thefastcars.freesevers.com" />
  ...
</results>
```

---

## **serp\_cache** SERP Cache.

Get cached SERP data for one or more domain names.

- `t` - one or more domain names, separated with `>>`

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=serp_cache&t=example.com
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <serp keyword="example.com" type="serp" domain="example.org" page="1" position="1" />
  <serp keyword="example.com" type="serp" domain="en.wikipedia.org" page="1" position="2" />
  <serp keyword="example.com" type="serp" domain="tools.ietf.org" page="1" position="3" />
  <serp keyword="example.com" type="serp" domain="onlaine.tumblr.com" page="1" position="4" />
  <serp keyword="example.com" type="serp" domain="tech.slashdot.org" page="1" position="5" />
  ...
</results>
```

---

## **serp\_detail** .

Retrieve specific SERP listings and ad listings for the provided keywords.

- `t` - one or more keywords, separated with `>>`

Returns results in JSON format.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=serp_detail&t=science
```

**Example Output:**

```
[
  {
    "keyword": "[science]",
    "avg_search": "450000",
    "avg_cpc": "1.220",
    "avg_competition": "7"
  }
]
```

**serp\_summary\_cache** Serp Summary (serp count and ad count) pulled from cache.

Retrieve archived SERP (search engine results phrase) lookups and related ad lookups for one or more keywords or domain names.

- t - one or more domain names or keywords, separated with >>

**Example URL:**

```
http://api.domainplex.com/?a=get&api_key=KEY&d=serp_summary_cache&t=sample&xml_standard=1
```

**Example Output:**

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <serp keyword=""sample"" serp="906000000" ads="0" />
</results>
```

**site** Web crawl summary for a specific domain.

Web crawl summary for a specific domain.

**Example URL:**

```
http://api.domainplex.com/?a=get&api_key=%api_key%&d=site&t=test.com
```

**site\_crawl\_summary\_status** Site crawler summary database for TLD and status lookups.

Given a domain extension and a desired status (Developed, Parked, Placeholder, Redirect) this API returns a list of all domain names and their site crawl information (title, description, page count, etc).

**Example URL:**

```
http://api.domainplex.com/?a=get&u=%username%&p=%password%&d=
```

## **site\_info** Site Information.

Fetch basic information about the site on the provided domain name.

- `t` - one or more domain names, separated with `>>`.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=site_info&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <site_info domain="example.com" parked="?" dns="?" response_code="302"
  response_details="redirect" redirect_domain="iana.org" site_pages="1" site_size="0"
  crawl_time="5386" unique_domain_links="9428" total_links="165039" links_ips="8712"
  links_unique_ips="6848" links_subnets="8708" links_unique_subnets="3375" links_internal="0"
  links_external_unique="0" pagerank="-" alexa_traffic_rank="52938" ip="192.0.43.10" />
</results>
```

## **site\_keyword\_cloud** Shows the top keywords and their density that appear on any domain name or web site.

Shows the top keywords and their density that appear on any domain name or web site.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=%api_key%&d=site_keyword_cloud
```

## **snapshot** Website Snapshot / Thumbnail.

Capture a snapshot of the website on the specified domain name.

Unlike other API calls, this call returns a raw image in PNG format.

- `domain` - the domain name to capture.
- `full=1` - optional - return the entire page instead of a cropped version (the default).
- `nocache=1` - optional - force the page to be refetched even if a recent version exists in the cache.
- `max_cache_age=X` - optional - specify the maximum acceptable age for a cached snapshot, in seconds. Default is one day (86400 seconds).
- `width` and `height` - optional - width and height to which the snapshot will be cropped and/or resized.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=snapshot&domain=yahoo.com
```

**Example Output:**

```
Raw PNG data.
```

**snapshot\_history** Website Snapshot History.

Retrieve archived snapshots for the specified domain name.

- `domain` - the domain name for archived snapshots.
- `full=1` - optional - if specified, return complete full-size images instead of thumbnails.
- `base64=1` - optional - if specified, return images in base64 format. Default behavior is URLs referencing images.
- `force_image=1` - optional - return raw image instead of URL or base64-encoded data.
- `most_recent=1` - optional - if specified, only return the most recently-archived snapshot.
- `limit=X` - optional - if specified, limit the results to X snapshots in descending archive date order.
- `width=X, height=X` - optional - specify the size of the thumbnail when `full=1` is omitted.

Note that this API call returns output in JSON format.

**Example URL:**

```
http://api.domainplex.com/?a=get&api_key=KEY&d=snapshot_history&domain=msn.com
```

**Example Output:**

```
[
  {
    "id": "3370250",
    "domain": "MSN.COM",
    "format": "jpeg",
    "img": null,
    "img_thumb": null,
    "mod_date": "2012-08-22 16:42:57",
    "ucc": "1345671777",
    "table_name": "snapshot",
    "db_hostname": "67.217.45.249",
    "dns_clean": "",
    "img_url_thumb": "http://snapshots.domainiq.com/img.php?domain=MSN.COM&width=250&height=125&img=...",
    "img_url_full": "http://snapshots.domainiq.com/img.php?domain=MSN.COM&full=1&img=..."
  },
  {
    "id": "591623",
    "domain": "MSN.COM",
    "format": "png",
    "img": null,
    "ip": "65.55.206.228",
    "dns": "NS5.MSFT.NET.",
    "mod_date": "2013-04-26 00:32:41",
    "ucc": "1366954361",
    "table_name": "snapshot6",
  }
]
```

```

        "db_hostname":"67.217.45.249",
        "dns_clean":"msft.net",
        "img_thumb":null,

"img_url_thumb":"http://snapshots.domainiq.com/img.php?domain=MSN.COM&width=250&height=125&img=
...",
        "img_url_full":"http://snapshots.domainiq.com/img.php?domain=MSN.COM&full=1&img=..."
    },
    ...
]

```

---

### **snapshot\_history\_image** Shows most recent snapshot for the site, as an image.

Fetch the most recently cached image for the specified domain name. Use the `snapshot` function to fetch a current snapshot.

- `t` - domain name to look up.

Returns data in raw PNG format.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=snapshot_history_image&t=domainiq.com
```

Example Output:

```
Raw PNG data.
```

---

### **sold\_history** Search Domain Sales Database.

Retrieve the archived sales history for one or more domain names or domains containing the specified keyword.

- `t` - one or more domain names or keywords.
- `o=flexible_keywords` - optional - expand keyword matching criteria.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=sold_history&t=cars&o=flexible_keywords&xml_standard=1
```

Example Output:

```

<?xml version="1.0" encoding="UTF-8"?>
<results>
  <sold_search term="cars" words="cars">
    <domain name="cars.lv" sold_date="2012-02-08" price="1995" source="Sedo" mediator="" />
    <domain name="cars.net" sold_date="2011-06-22" price="170000" source="UpMarketDNs"
mediator="" />
    <domain name="cars.asia" sold_date="2008-02-24" price="3350" source="Private"
mediator="dotAsia" />
  </sold_search>
</results>

```

```

        <domain name="cars.net.au" sold_date="2011-02-21" price="5101" source="netfleet.com.au"
mediator="" />
        ...
    </sold_search>
</results>

```

---

## **systems** System Information.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=systems
```

---

## **trademark** Perform Trademark Analysis and Risk Assessment.

Process trademark matching and risk analysis for one or more domains. Also, detects typo trademarks and matches them with the possible trademark. This tool is NOT a comprehensive trademark check. Do not use this tool to make legal decisions.

Please refer to the [Sending Trademark Check Queries](#) section for complete details.

- `t` - one or more domain names, separated with `>>`
- `strict_level=X` - optional - a numeric value between 10 and 100 specifying how close the typo must come (percentage-wise) to the actual trademark for a match. Higher level will match fewer trademarks. Default is 65.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=trademark&t=mikrosoft.com&xml_standard=1
```

Example Output:

```

<?xml version="1.0" encoding="UTF-8"?>
<results>
  <trademark domain="mikrosoft.com" type="typo" word="microsoft" company="Microsoft" risk="10"
score="67.0" domains_matched="9079" />
</results>

```

---

## **typo** Generate Domain Typos.

Generate possible typos of one or more supplied domain names.

- `t` - one or more domain names, separated with `>>`

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=typo&t=example.com&xml_standard=1
```



**Example Output:**

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <domain name="example.com">
    <typo domain="wexample.com" />
    <typo domain="sexample.com" />
    <typo domain="dexample.com" />
    <typo domain="rexample.com" />
    <typo domain="3example.com" />
    <typo domain="4example.com" />
    <typo domain="ezexample.com" />
    ...
  </domain>
</results>
```

**wayback** Wayback Records Crawler.

Fetch Wayback archive records for the supplied domain(s).

- `t` - one or more domain names, separated with `>>`

**Example URL:**

```
http://api.domainplex.com/?a=get&api_key=KEY&d=wayback&t=example.com
```

**Example Output:**

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <domain name="example.com" records="0" age="2015" last_active_date="" last_parked_date=""
  last_parked_provider="" keyword_summary="" title="" language="" error="0" sub_errors="0" />
</results>
```

**whois** Perform Live and Cached Whois Lookups.

Perform live whois data lookups for one or more domain names.

Refer to the [Sending Whois Queries](#) section for complete details.

Add optional parameter `registry_only=1` to limit lookups to registry-only data for COM/NET domains, which will omit registrar-level ownership information but will result in faster queries.

**Example URL:**

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois&t=example.com&xml_standard=1
```

**whois\_archive** Historical Whois Lookup.

Fetch the most recent archived whois data for the supplied domain name(s).

- `t` - one or more domain names, separated with `>>`

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois_archive&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <whois domain="example.com">
    <domain>example.com</domain>
    <status>clientTransferProhibited,ClientDeleteProhibited,ClientUpdateProhibited</status>
    <registrar>RESERVED-INTERNET ASSIGNED NUMBERS AUTHORITY</registrar>
    <registrar_url>http://res-dom.iana.org</registrar_url>
    <ns_1>A.IANA-SERVERS.NET</ns_1>
    <ns_2>B.IANA-SERVERS.NET</ns_2>
    <creation_date>1995-08-14</creation_date>
    <expiration_date>2015-08-13</expiration_date>
    <update_date>2014-08-14</update_date>
    ...
  </results>
```

## **whois\_archive\_raw** Raw Archived Whois Data.

Fetch raw archived whois records for the supplied domain name(s).

- `t` - one or more domain names, separated with `>>`

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois_archive_raw&t=example.com
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <whois domain="example.com">
    <domain>example.com</domain>
    <mod_date>2015-04-21</mod_date>
    <utt>1429645686</utt>
    <raw>Domain Name: EXAMPLE.COM
    Registrar: RESERVED-INTERNET ASSIGNED NUMBERS AUTHORITY
    Sponsoring Registrar IANA ID: 376
    Whois Server: whois.iana.org
    Referral URL: http://res-dom.iana.org
    Name Server: A.IANA-SERVERS.NET
    Name Server: B.IANA-SERVERS.NET
    ...
  </raw>
</whois>
  <whois domain="example.com">
    <domain>example.com</domain>
```

```

    <mod_date>2015-04-14</mod_date>
    <utt>1429044950</utt>
    <raw>Domain Name: EXAMPLE.COM
Registrar: RESERVED-INTERNET ASSIGNED NUMBERS AUTHORITY
Sponsoring Registrar IANA ID: 376
Whois Server: whois.iana.org
Referral URL: http://res-dom.iana.org
Name Server: A.IANA-SERVERS.NET
Name Server: B.IANA-SERVERS.NET
    ...
    </raw>
</whois>
    ...
</results>

```

---

### **whois\_archive\_raw\_custom** Archived Whois Data.

Fetch a list of available archived raw records for a domain name, or fetch one of those records based on its ID.

- `t` - the domain name for archived whois lookup.
- `dates_only=1` - return a list of the dates of available archived whois records for the domain name.
- `record_id=X` - return the record specified by record ID X. Get a list of available records with a `dates_only=1` call first.

Note that this tool only returns results in PHP serialized format.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois_archive_raw_custom&t=sample.com&dates_only=1
```

Example Output:

```
a:36:{i:0;o:8:"stdClass":2:{s:2:"id";s:9:"719836267";s:8:"mod_date";s:10:"2015-01-07";}i:1;o:8:"stdClass":2:{s:2:"id";s:9:"713572586";s:8:"mod_date";s:10:"2015-02-13";}i:2;o:8:"stdClass":2:{s:2:"id";s:9:"702956690";s:8:"mod_date";s:10:"2014-07-18";}i:3;o:8:"stdClass":2:{s:2:"id";s:9:"696940876";s:8:"mod_date";s:10:"2014-09-19";}i:4;o:8:"stdClass":2:{s:2:"id";s:9:"696826040";s:8:"mod_date";s:10:"2014-09-18";}}
```

---

### **whois\_cache** Check the most recent cached whois record.

Check the most recent cached whois record.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=%api_key%&d=whois_cache&t=domainplex.com
```

---

### **whois\_history** Whois history database.

Retrieve archived whois history records for one or more domain names.

- `t` - one or more domain names, specified by `>>`
- `limit=X` - optional - limit the number of returned results to `X`. Default is 100.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois_history&t=example.com
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <whois domain="example.com">
    <domain>example.com</domain>
    <status>clientTransferProhibited,ClientDeleteProhibited,ClientUpdateProhibited</status>
    <registrar>RESERVED-INTERNET ASSIGNED NUMBERS AUTHORITY</registrar>
    <registrar_url>http://res-dom.iana.org</registrar_url>
    <ns_1>A.IANA-SERVERS.NET</ns_1>
    <ns_2>B.IANA-SERVERS.NET</ns_2>
    <creation_date>1995-08-14</creation_date>
    <expiration_date>2013-08-13</expiration_date>
    <update_date>2012-08-14</update_date>
    <email />
    <registrant>Internet Assigned Numbers Authority</registrant>
    <domains_registrant_name>66</domains_registrant_name>
    <domains_registrant_email>1</domains_registrant_email>
    <whois_record_date>2013-08-13 07:57:26</whois_record_date>
  </whois>
  <whois domain="example.com">
    ...
  </whois>
  <whois domain="example.com">
    ...
  </whois>
  ...
</results>
```

---

**whois\_history\_raw\_vm** Get most recent raw historical whois record for a domain or a set of domains.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=%api_key%d=whois_history_raw_vm&t=test.com
```

---

**whois\_redundant** Perform Live Redundant Whois Lookups.

Perform live whois data lookups for one or more domain names. This function is essentially the same as the `whois` function, but performs whois lookups at multiple servers to attempt to get the quickest and/or most complete result.

Options are the same as for the `whois` function, except that `release_socket` and notifications are not supported. Refer to the [Sending Whois Queries](#) section for complete details.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois_redundant&t=example.com
```

**whois\_reverse** Reverse whois search email or registrant name showing all domains associated with either an email or registrant name.

Reverse whois search email or registrant name showing all domains associated with either an email or registrant name.

Example URL:

```
http://api.domainplex.com/?a=get&u=%username%&p=%password%&d=
```

**whois\_reverse\_cache** Cached reverse lookup using secondary DB.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=%api_key%&d=whois_reverse_cache
```

**whois\_reverse\_search** Perform reverse whois lookups on registrants or emails.

Look up one or more email addresses or registrant names and return a count of associated domains for each.

Specify `email` or `registrant` for the `o` parameter, and supply email addresses or registrant names via the `t` parameter. Optionally provide the `include_domains=1` parameter to include each resulting domain name in the output.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=whois_reverse_search&o=email&t=domains@yahoo.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <reverse_whois status="ok" query="domains@yahoo.com" count="4" />
</results>
```

**wordtracker** Check Cached Word Tracker Data.

Fetch cached Word Tracker search volume data for the supplied keywords.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=wordtracker&t=example&xml_standard=1
```

**Example Output:**

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <wordtracker term="example" search_volume="136" />
</results>
```

**zone\_archive** Archive of zone files.

Fetch archived zone files.

- `type` - the type of query to perform. Options include:
  - `index` - return a list of available zone files. This is the default mode and will return **many** results (over 100,000). For optimal performance on the client side, please cache the results of this call and use it as minimally as possible.
  - `get_file=X` - retrieve a zone file based on its file ID (X).

**Example URL:**

```
http://api.domainplex.com/?a=get&api_key=KEY&d=zone_archive&type=index
```

**Example Output:**

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <file id="1" basename="2011-07-23-full.biz.gz" date="2011-07-23" type="biz"
  filesize="30087064" />
  <file id="2" basename="2011-07-23-full.com.gz" date="2011-07-23" type="com"
  filesize="1801202610" />
  <file id="3" basename="2011-07-23-full.info.gz" date="2011-07-23" type="info"
  filesize="121119928" />
  <file id="4" basename="2011-07-23-full.net.gz" date="2011-07-23" type="net"
  filesize="259428020" />
  <file id="5" basename="2011-07-23-full.org.gz" date="2011-07-23" type="org"
  filesize="142607754" />
  <file id="6" basename="2011-07-23-full.sk.gz" date="2011-07-23" type="sk" filesize="25474476"
  />
  <file id="7" basename="2011-07-23-full.us.gz" date="2011-07-23" type="us" filesize="22241870"
  />
</results>
```

**zone\_changes** Download reports of newly registered or newly deleted domains with whois data.

Fetch a list of changes made to the zone file such as new registrations or domains removed from the zone for a specific date.

`type` can be `create_date` or `expire_date`. Supply a date via the `date` parameter in `yyyy-mm-dd` format. Note that the date must be between the current date and -4 days.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=zone_changes&type=create_date&date=2014-01-01&xml_standard=1
```

## zone\_diff -.

-

Example URL:

```
http://api.domainplex.com/?a=get&u=%username%&p=%password%&d=zone_diff
```

## zone\_digger Zone Digger.

Perform wildcard searches on the zone, optionally returning site and whois information.

- keyword1 - the first keyword to search for.
- keyword2 - optional - the second keyword to search for.
- logic1 and logic2 - optional - the kind of match to perform. Valid options:
  - contain
  - start
  - end
  - not
- exclude\_parked=1 - optional - if specified, exclude domains that are detected as parked.
- exclude\_undeveloped=1 - optional - if specified, exclude domains that are detected as undeveloped.
- site\_stats=1 - optional - if specified, include stats for the returned domains.
- whois\_stats=1 - optional - if specified, include whois details for the returned domains.

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=zone_digger&t=example.com&xml_standard=1
```

Example Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <zone_info domain="ALTISSIMAGROUPINGS.COM" dns="HOST4U.NET" parked="0" />
  <zone_info domain="ANIMALGROUPINGS.COM" dns="DOMAINCONTROL.COM" parked="0" />
  <zone_info domain="ARTGROUPINGS.COM" dns="DOMAINCONTROL.COM" parked="0" />
  <zone_info domain="BEACONGROUPINGS.COM" dns="REGISTRAR-SERVERS.COM" parked="0" />
  <zone_info domain="CRESCENTMOONGROUPINGS.COM" dns="NAME-SERVICES.COM" parked="0" />
  <zone_info domain="FOREVERGROUPINGS.COM" dns="REGISTRAR-SERVERS.COM" parked="0" />
  <zone_info domain="FURNITUREGROUPINGS.COM" dns="INTERNETTRAFFIC.COM" parked="1" />
  <zone_info domain="GOLFGROUPINGS.COM" dns="STABLEHOST.COM" parked="0" />
  <zone_info domain="GROUPINGS.COM" dns="POWWEB.COM" parked="0" />
  <zone_info domain="GROUPINGS.INFO" dns="START-DNS.DE" parked="0" />
  <zone_info domain="GROUPINGS.NET" dns="NAME.COM" parked="0" />
</results>
```

**zone\_domains** Download a list of domains in a zone.

Download a plain-text list of all domains in the zone for the specified TLD. Most gTLDs and nTLDs are supported.

Example URL:

```
http://api.domainplex.com/?a=get&u=%username%&p=%password%&d=zone_domains&t=cool
```

---

**zone\_search** Zone Search.

Quickly search the zone for one or more keywords.

Returns results in plain text format.

- t - one or more keywords, separated with >>

Example URL:

```
http://api.domainplex.com/?a=get&api_key=KEY&d=zone_search&t=example.com&xml_standard=1
```

---

Example Output:

```
SEMEXAMPLE.COM  
BUSINESSREPORTEEXAMPLE.COM  
ASFEXAMPLE.COM  
DOMAINEXAMPLE.COM  
RESIGNATIONLETTEREXAMPLE.COM  
EASYEXAMPLE.COM  
SAY4EXAMPLE.COM  
BETHEEXAMPLE.COM  
GOODEEXAMPLE.COM  
AGOOEXAMPLE.COM
```

---



## Sample HTTP POST Functions

### PHP

```
public function post_data($url, $data, $optional_headers = null) {  
  
    $params = array('http' => array(  
        'method' => 'POST',  
        'content' => $data,  
        'timeout' => 600  
    ));  
  
    if ($optional_headers !== null) {  
        $params['http']['header'] = $optional_headers;  
    }  
  
    $ctx = stream_context_create($params);  
    $fp = fopen($url, 'rb', false, $ctx);  
  
    $response = stream_get_contents($fp);  
  
    return $response;  
  
}
```

**Java**

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.URL;

import javax.net.ssl.HttpURLConnection;

public class ExampleClass {

    public String postData(String url, String data) throws Exception {

        String url = "https://api.domainplex.com/";
        URL obj = new URL(url);
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();

        con.setRequestMethod("POST");
        con.setDoOutput(true);

        DataOutputStream wr = new DataOutputStream(con.getOutputStream());
        wr.writeBytes(data);
        wr.flush();
        wr.close();

        BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();

        return response.toString();
    }
}
```

**Perl**

```
sub postData {  
    my ($url, $data) = (@_);  
    use LWP::UserAgent;  
  
    my $ua = LWP::UserAgent->new;  
    my $req = HTTP::Request->new(POST => $url);  
    $req->content($data);  
  
    my $resp = $ua->request($req);  
    return $resp->decoded_content;  
}
```

## Python

```
import urllib
import urllib2

def postData(url, data):

    encodedData = urllib.urlencode(data)
    req = urllib2.Request(url, encodedData)

    response = urllib2.urlopen(req)
    return response.read()
```

## Enterprise Services

Looking to perform high volume appraisals and other queries?

The same technology that powers DomainPlex is now available to anyone at an unbeatable price.

The following technologies can be licensed and used on **your hardware or ours** with no per-query costs:

- Appraisal Technology
- Search Engine Scanning Technology
- Keyword CPC Scanning Technology
- Whois Scanning Technology
- Web Site Scanning Technology
- Domain Trademark Scanning Technology

Licensing any one of our scanning technologies also gives you **unlimited access** to our archived databases which contain billions of historical metrics (search engine, search popularity, whois, resolve status, hosting, etc).

### Advantages of using our Enterprise Services:

- Affordable. One time monthly fee. No query limits (based on your hardware and IP space).
- Quick setup, on our hardware or yours.
- No contracts. Cancel any time.
- Easy to use web interface accessible from EstiBot.com.
- Free updates. We make sure your scanners are up to date and functioning perfectly.
- Centralized data storage on our vast network. Instantly access all of your scan results.
- Help in data management.
- Significant discounts for multiple licenses.
- 24/7 support team to assist with any problems you may have.

If you are interested in licensing our technology please contact [api@domainplex.com](mailto:api@domainplex.com) and describe your needs so we can design a solution to fit your business and budget.

## Errors

Occasionally, the API will return an error message instead of the anticipated query response. Errors are formatted using a standardized title/message XML format for easy parsing and interpretation.

Errors assume the following format:

```
<api_error>
  <error_title>ERROR TITLE</error_title>
  <error_description>ERROR DESCRIPTION</error_description>
</api_error>
```

Here are some error codes and their meanings:

- `<error_title>limit reached</error_title>` – You have reached your daily global query limit. Limits reset at midnight US Central time.
- `<error_title>tool limit reached</error_title>` – You have reached the daily limit for the specified tool (ex: appraiser). You can however use other tools (ex: whois). Limits reset at midnight US Central time.
- `<error_title>ip limit reached</error_title>` – Your user has reached the specified maximum number of queries per IP. This limit is set by YOU to prevent abuse. You should alert the user that they have reached their daily limit. Limits reset at midnight US Central time.
- `<error_title>daemon error</error_title>` - There was an error processing your request. Please try again. If this problem persists, please contact [api@domainplex.com](mailto:api@domainplex.com)
- `<error_title>system maintenance</error_title>` - We are performing system maintenance. The error description will usually indicate the duration.

If you experience errors other than those described above and need assistance please contact [api@domainplex.com](mailto:api@domainplex.com).

## Support

Our experienced staff is here to assist you with any questions or problems you may have during regular business hours.

We strongly recommend reading through this entire API guide prior to contacting us with API questions.

You may use the following contact information to get in touch with us:

**E-mail:** [api@domainplex.com](mailto:api@domainplex.com)

**Phone:** 847-380-2733

**Fax:** 847-897-2054

**Business Hours:**

Mon – Sat: 7 AM – 5 PM US Central Time

Sunday: Closed

## Glossary of Terms

- **API** - Application Programming Interface. A backend server application that processes queries and returns standardized results.
- **API Call** – A query to the DomainPlex API system sent via HTTP GET or POST method.
- **Aftermarket** – The general term applied to “second hand” domains, or domains which are not registered by the end user but are purchased for the purpose of resale by speculators and investors.
- **ccTLD** – A country code top level domain (ex: .co.uk, .de, .pl, .nl, .fi, .se, etc).
- **Domain Auction** – Many domains that reach their expiration date are auctioned off and only enter the drop pool if they are not sold during the aftermarket auction.
- **Domain Term / Words** – The actual words that appear within a domain. For example, if the domain was “fastcars.com” then the parsed words would be “fast cars”.
- **Drop Date** – The date a specific domain will become available for registration.
- **Drop Pool** – One or more domains scheduled to become available at a specific date and time.
- **Expiration Date** – The date a specific domain will expire. Typically, a domain “drops” or becomes available within 0-120 days of expiring, depending on the registrar.
- **Link Popularity** – The number of web sites that link to a specific domain.
- **Link Saturation** – The amount of pages that appear in a search engine for a specific domain.
- **Registrar** – An organization that provides domain registrations to individuals and businesses (ex: GoDaddy).
- **Registry** – An organization that handles and maintains domain name records as well as registrar accreditations. For example, Verisign is the registry that handles .com/.net TLDs (Top Level Domains).
- **TLD** – Top Level Domain, ex: .com, .net, .org
- **Term Popularity** – The number of search results for a domain term. For example, if our domain was “fastcars.com” then the term would be “fast cars” and the term popularity would be the number of results that show up for the term “fast cars”.
- **Term Ads** - The number of ad results for a domain term. For example, if our domain was “fastcars.com” then the term would be “fast cars” and the term ads would be the number of ads that show up for the term “fast cars”.
- **SLD** – Second Level Domain, or the domain name minus the extension, ex: domain.com has domain as SLD.
- **Zone File** – A list of domain names and name servers associated with them. Zone files are provided by the registry for the purpose of resolving domains.