

Whois API User Guide

Our Hosted Whois Web Service provides registration details, also known as Whois Records, of domain names or IP addresses.

WhoisXmlApi.com provides RESTful APIs. It is designed for server-to-server communication between your system and the WhoisXmlApi.com network using standard HTTP or HTTPS protocols. The query responses are delivered in either XML or JSON formats based on your requests which are made through GET/POST methods.

To get started you need an account with WhoisXmlApi.com. Authentication is required each time in order to use the API. The first 500 Whois API calls are complimentary when you register for a free developer account.

API Status

You can check the API's current performance and availability status on the [API Status Page](#).

Making a Call to Whois API Web Service

Use the following URL:

<https://www.whoisxmlapi.com/whoisserver/WhoisService?domainName=google.com&username=x&password=x>

Supported input parameters

Parameters	Values
domainName	required , the domain for which whois data is requested
username	required unless ApiKey authentication is used, your account username
password	required unless ApiKey authentication is used, your account password; make sure it is URL-encoded if it contains special characters (&, %, * etc.)
requestObject	required unless password authentication is used, encoded user identifier for ApiKey authentication
digest	required unless password authentication is used, ApiKey access token
outputFormat	XML JSON (defaults to XML)
da	0 1 2 (defaults to 0) 1 results in a quick check on domain availability, 2 is slower but more accurate
preferFresh	0 1 (defaults to 0) 1 results in getting the latest Whois record even if it's
ip	0 1 (defaults to 0) 1 results in returning ips for the domain name incomplete

Parameters	Values
ipWhois	0 1 (defaults to 0) 1 results in returning the whois record for the hosting ip if the whois record for the tld of the input domain is not supported
checkProxyData	0 1 (defaults to 0) 1 results in fetching proxy/whois guard data, if it exists, in the <i>WhoisRecord</i> → <i>privateWhoisProxy</i> schema element
thinWhois	0 1 (defaults to 0) 1 results in returning whois data from registry only, without fetching data from registrar; returned registry data corresponds to the <i>WhoisRecord</i> → <i>registryData</i> schema element
ignoreRawTexts	0 1 (defaults to 0) 1 results in stripping all raw text from the output
callback	a javascript function used when outputFormat is JSON; this is an implementation known as JSONP which invokes the callback on the returned response
_parse	0 1 (defaults to 0) 1 provides parsing for input whois raw texts described below
registryRawText	a string representing the registry whois raw text to be parsed; works only when the _parse parameter is equal to 1
registrarRawText	a string representing the registrar whois raw text to be parsed; works only when the _parse parameter is equal to 1

Authentication

Password authentication

This method uses your account's username and password to authenticate with the web service. Just pass your credentials as query string parameters (*username, password*).

API key authentication

Generate access tokens providing your API and secret keys to make Whois API calls without sending your username and password. These tokens will need to be included as part of the request URL (*requestObject, digest*). Please, note that all tokens expire after 5 minutes and will have to be regenerated. Access the [API Key Management](#) section of your account to create your keys (make sure the key you're using is marked as active).

Query Account Balance Information

In order to query your account balance, use the following URL:

<https://www.whoisxmlapi.com/accountServices.php?servicetype=accountbalance&username=x&password=x>

Supported input parameters

Parameters	Values
servicetype	required , type of service; in this case, "accountBalance"
username	required , your account username
password	required , your account password; make sure it is URL-encoded if it contains special characters (&, %, * etc.)

Account Balance Low or Zero Notification

When your account balance falls below certain level (default is 100, but customizable as per your need) you will receive an email alert. Another email will be sent when your account balance reaches 0.

To customize the warning level, use the URL:

[whoisxmlapi.com/accountServices.php?servicetype=accountUpdate&username=x&password=x&warn_threshold=30](https://www.whoisxmlapi.com/accountServices.php?servicetype=accountUpdate&username=x&password=x&warn_threshold=30)

Supported input parameters

Parameters	Values
servicetype	required , type of service; in this case, "accountUpdate"
username	required , your account username
password	required , your account password; make sure it is URL-encoded if it contains special characters (&, %, * etc.)
warn_threshold	a positive number (defaults to 10) when your account balance reaches this value a warning email will be sent
warn_threshold_enabled	0 1 (defaults to 1) indicates whether the low balance warning notification will be sent at all
warn_empty_enabled	0 1 (defaults to 1) indicates whether the zero balance email notification will be sent to you
output_format	XML JSON (defaults to XML)

Usage of HTTP & HTTPS

You can use both http and https to make API queries. With HTTPS the connection will be more secure but slightly slower.

TLDs Supported

We support thousands of TLDs, both gTLDs and ccTLDs. You can see the entire list [here](#).

XML Schema / Whois Query Result Documentation

You can download the following for your reference:

[xml schema](#)

[wsdl schema](#)

[documentation on the query result](#)

[sample xml result](#)

An XML Schema describes the structure of an XML document. The XML Schema language is also referred to as XML Schema Definition (XSD).

An WSDL (Web Services Description Language) document is written in XML and describes a web service by specifying its location and methods.

More info:

https://www.w3schools.com/xml/schema_intro.asp

https://www.w3schools.com/xml/xml_wsdl.asp

Sample Code for Making a Query to Whois API Web Service

You may view examples of querying Whois API by clicking on the following links or visiting our [GitHub page](#).

User password authentication

- C#: [example](#), [full project](#)
- Java: [example](#), [full project](#)
- Javascript: [using the callback parameter](#)
- Node.js: [example](#)
- Perl: [example](#)
- PHP: [example](#), [XML & JSON parsing](#)
- PowerShell: [example](#)
- Python: [example](#)
- Ruby: [example](#)

API key authentication

- C#: [example](#), [full project](#)
- Java: [example](#), [full project](#)
- Javascript: [JSONP](#)
- Node.js: [example](#)
- Perl: [example](#)
- PHP: [example](#)
- PowerShell: [example](#)
- Python: [example](#)
- Ruby: [example](#)

Frequently Asked Questions

How old is the Whois record returned?

The data age ranges between 0 to 24 hours. Typical data age is 0 hours (real time) since we don't get many duplicate domains from user queries in a day. We cache the Whois record for up to 24 hours. When you send us a query, we first check to see if the Whois record in cache is older than 24 hours, if it is, then we will fetch it in real time. You may also use `_hardRefresh=1` to force a real-time fetch, but this would cost **5 times the credits**.

Why do I get inconsistent Whois record response? Sometimes registrant information shows up under WhoisRecord and sometimes it shows up under WhoisRecord→registryData.

For each domain name there are at most 2 Whois records, one from registry (e.g. Verisign) and one for registrar (e.g. Godaddy). We return the registry Whois record under `WhoisRecord→registryData` and registrar whois record under `WhoisRecord`. For most tlds other than com/net, only registry Whois record exists.

Are there any TLDs that you don't support?

There are some TLDs for which their registry/registrar doesn't publish Whois records. For these TLDs we only provide a simple Whois record with name server information. [Click here](#) for a list of such TLDs.

Error Handling

Error information containing the [error code](#) and text description is returned in the following format.

Sample output

<pre><ErrorMessage> <errorCode>WHOIS_01</errorCode> <msg> The parameter domainName is missing. </msg> </ErrorMessage></pre>	<pre>{ "ErrorMessage": { "errorCode": "WHOIS_01" "msg": "The parameter domainName is missing." } }</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------

Usage Limits & Requests Throttling

The maximum number of requests per second is 50. In case that the limit is breached, your subsequent requests will be rejected until the next second.

Sample output

<pre><ErrorMessage> <errorCode>AUTHENTICATE_06</errorCode> <msg> You are limited to 50 queries per second. The request is rejected. </msg> </ErrorMessage></pre>	<pre>{ "ErrorMessage": { "errorCode": "AUTHENTICATE_06" "msg": "You are limited to 50 queries per second..." } }</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------

Terms of Service or SLA for Using Whois API Web Service

Please click on the link to review the [Terms of Service](#).

Sample Output (XML)

Output	Output continued
<pre><WhoisRecord> <createdDate> 1997-09-15T00:00:00-0700 </createdDate> <updatedDate> 2015-06-12T10:38:52-0700 </updatedDate> <expiresDate> 2020-09-13T21:00:00-0700 </expiresDate> <registrant> <name> Dns Admin </name> <organization> Google Inc. </organization> <street1> Please contact contact-admin@google.com, 1600 Amphitheatre Parkway </street1> <city> Mountain View </city> <state> CA </state> <postalCode> 94043 </postalCode> <country> UNITED STATES </country> <countryCode> US </countryCode> <email> dns-admin@google.com </email> <telephone> 16502530000 </telephone> <fax> 16506188571 </fax> <rawText> Registrant Name: Dns Admin Registrant Organization: Google Inc. Registrant Street: Please contact contact- admin@google.com, 1600 Amphitheatre Parkway Registrant City: Mountain View Registrant State/Province: CA Registrant Postal Code: 94043 Registrant Country: US Registrant Phone: +1.6502530000 Registrant Fax: +1.6506188571 Registrant Email: dns-admin@google.com </rawText> </registrant></pre>	<pre><administrativeContact> <name> DNS Admin </name> <organization> Google Inc. </organization> <street1> 1600 Amphitheatre Parkway </street1> <city> Mountain View </city> <state> CA </state> <postalCode> 94043 </postalCode> <country> UNITED STATES </country> <countryCode> US </countryCode> <email> dns-admin@google.com </email> <telephone> 16506234000 </telephone> <fax> 16506188571 </fax> <rawText> Admin Name: DNS Admin Admin Organization: Google Inc. Admin Street: 1600 Amphitheatre Parkway Admin City: Mountain View Admin State/Province: CA Admin Postal Code: 94043 Admin Country: US Admin Phone: +1.6506234000 Admin Fax: +1.6506188571 Admin Email: dns-admin@google.com </rawText> </administrativeContact> </WhoisRecord></pre>

Sample Output (JSON)

```
{
  "WhoisRecord": {
    "createdDate": "1997-09-15T00:00:00-0700",
    "updatedDate": "2015-06-12T10:38:52-0700",
    "expiresDate": "2020-09-13T21:00:00-0700",
    "registrant": {
      "name": "Dns Admin",
      "organization": "Google Inc.",
      "street1": "Please contact contact-admin@google.com, 1600 Amphitheatre Parkway",
      "city": "Mountain View",
      "state": "CA",
      "postalCode": "94043",
      "country": "UNITED STATES",
      "countryCode": "US",
      "email": "dns-admin@google.com",
      "telephone": "16502530000",
      "fax": "16506188571",
      "rawText": "Registrant Name: Dns Admin\nRegistrant Organization: Google Inc.\nRegistrant Street:
        Please contact contact-admin@google.com, 1600 Amphitheatre Parkway\nRegistrant City:
        Mountain View\nRegistrant State/Province: CA\nRegistrant Postal Code: 94043\nRegistrant
        Country: US\nRegistrant Phone: +1.6502530000\nRegistrant Fax:
        +1.6506188571\nRegistrant Email: dns-admin@google.com"
    },
    "administrativeContact": {
      "name": "DNS Admin",
      "organization": "Google Inc.",
      "street1": "1600 Amphitheatre Parkway",
      "city": "Mountain View",
      "state": "CA",
      "postalCode": "94043",
      "country": "UNITED STATES",
      "countryCode": "US",
      "email": "dns-admin@google.com",
      "telephone": "16506234000",
      "fax": "16506188571",
      "rawText": "Admin Name: DNS Admin\nAdmin Organization: Google Inc.\nAdmin Street: 1600
        Amphitheatre Parkway\nAdmin City: Mountain View\nAdmin State/Province:
        CA\nAdmin Postal Code: 94043\nAdmin Country: US\nAdmin Phone:
        +1.6506234000\nAdmin Fax: +1.6506188571\nAdmin Email: dns-admin@google.com"
    }
  }
}
```

Output Schema

Schema	Schema continued
<pre> "WhoisRecord": { "createdDate": "Date", "updatedDate": "Date", "expiresDate": "Date", "registrant": { "name": "String", "organization": "String", "street1": "String", "city": "String", "state": "String", "postalCode": "String", "country": "String", "countryCode": "String", "email": "String", "telephone": "String", "fax": "String", "rawText": "String" }, "administrativeContact": { "name": "String", "organization": "String", "street1": "String", "city": "String", "state": "String", "postalCode": "String", "country": "String", "countryCode": "String", "email": "String", "telephone": "String", "fax": "String", "rawText": "String" }, "domainName": "String", "nameServers": { "rawText": "String", "hostNames": "String", "ips": "String" }, "status": "String", "rawText": "String", "parseCode": "String", "header": "String", "strippedText": "String", "audit": { "createdDate": { "@class": "String", "\$": "Date" }, "updatedDate": { "@class": "String", "\$": "Date" } } }, </pre>	<pre> "customField1Name": "String", "customField1Value": "String", "registrarName": "String", "registrarIANAID": "Number", "customField2Name": "String", "customField3Name": "String", "customField2Value": "String", "customField3Value": "String", "registryData": { "createdDate": "String", "updatedDate": "String", "expiresDate": "String", "domainName": "String", "nameServers": { "rawText": "String", "hostNames": "Array", "ips": "Array" }, "status": "String", "rawText": "String", "parseCode": "String", "header": "String", "strippedText": "String", "footer": "String", "audit": { "createdDate": { "@class": "String", "\$": "Date" }, "updatedDate": { "@class": "String", "\$": "Date" } } "registrarName": "String", "registrarIANAID": "Number", "createdDateNormalized": "Date", "updatedDateNormalized": "Date", "expiresDateNormalized": "Date", "whoisServer": "String", "referralURL": "Date" }, "contactEmail": "Date", "domainNameExt": "Date", "estimatedDomainAge": "Number" } </pre>

Error Codes

WHOIS_01: *\$errorMessage*

WHOIS_02: *User is not logged in.*

WHOIS_03: *Unable to retrieve whois record for \$domainName*

WHOIS_04: *\$errorMessage*

DB_01: *Authentication can not complete due to database error.*

DB_02: *Exception in getUser : \$errorMessage*

DB_03: *ApiKey is missing*

DB_04: *Api Key is not found.*

DB_05: *Exception in getApiKey() while getting Entity:
\$errorMessage*

DB_06: *Api Key is not found.*

DB_07: *Exception in getApiKey() while getting Entity:
\$errorMessage*

AUTHENTICATE_01: *Queries available for \$id is
\$balance/\$reserve # monthly:
\$monthlyBalance:\$monthlyReserve, please refill*

AUTHENTICATE_02: *Queries available for \$ip is
\$balance/\$reserve, please refill*

AUTHENTICATE_03: *Username/password is missing*

AUTHENTICATE_04: *ApiKey parameters missing*

AUTHENTICATE_05: *Access restricted due to the subscription
limitation.*

AUTHENTICATE_06: *You are limited to 50 queries per second. The
request is rejected.*

AUTHENTICATE_07: *Request timeout*

AUTHENTICATE_08: *Timestamp in the future*

AUTHENTICATE_09: *Unknown error!*

AUTHENTICATE_10: *User account \$userName is \$status*

AUTHENTICATE_11: *Token missing*

AUTHENTICATE_12: *Decoding token failed*

AUTHENTICATE_13: *Token has expired*

AUTHENTICATE_14: *Captcha authentication failed*

AUTHENTICATE_15: *Cannot get IPQuota*

AUTHENTICATE_16: *Username is missing*

AUTHENTICATE_17: *Password is missing*

AUTHENTICATE_18: *Invalid username/password.*

API_KEY_01: *ApiKey is disabled.*

API_KEY_02: *Timestamp in the future*

API_KEY_03: *Request timeout*

API_KEY_04: *You are not authorized*

API_KEY_05: *ApiKey authenticate failed: \$errorMessage*