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 a developer’s 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:

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

### Supported input parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>domainName</td>
<td><strong>required</strong>, the domain for which whois data is requested</td>
</tr>
<tr>
<td>username</td>
<td><strong>required unless</strong> ApiKey authentication is used, your account username</td>
</tr>
<tr>
<td>password</td>
<td><strong>required unless</strong> ApiKey authentication is used, your account password; <strong>make sure it is URL-encoded if it contains special characters (&amp;, %, * etc.)</strong></td>
</tr>
<tr>
<td>requestObject</td>
<td><strong>required unless</strong> password authentication is used, encoded user identifier for ApiKey authentication</td>
</tr>
<tr>
<td>digest</td>
<td><strong>required unless</strong> password authentication is used, ApiKey access token</td>
</tr>
<tr>
<td>outputFormat</td>
<td>XML</td>
</tr>
<tr>
<td>da</td>
<td>0</td>
</tr>
<tr>
<td>preferFresh</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip</td>
<td>0</td>
</tr>
<tr>
<td>checkProxyData</td>
<td>0</td>
</tr>
<tr>
<td>thinWhois</td>
<td>0</td>
</tr>
<tr>
<td>callback</td>
<td>a javascript function used when outputFormat is JSON; this is an implementation known asjsonp which invokes the callback on the returned response</td>
</tr>
<tr>
<td>_parse</td>
<td>0</td>
</tr>
<tr>
<td>registryRawText</td>
<td>a string representing the registry whois raw text to be parsed; works only when the _parse parameter is equal to 1</td>
</tr>
<tr>
<td>registrarRawText</td>
<td>a string representing the registrar whois raw text to be parsed; works only when the _parse parameter is equal to 1</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>servicetype</td>
<td>required, type of service; in this case, &quot;accountBalance&quot;</td>
</tr>
<tr>
<td>username</td>
<td>required, your account username</td>
</tr>
<tr>
<td>password</td>
<td>required, your account password; make sure it is URL-encoded if it contains special characters (&amp;, %, * etc.)</td>
</tr>
</tbody>
</table>

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

Supported input parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>servicetype</td>
<td>required, type of service; in this case, &quot;accountUpdate&quot;</td>
</tr>
<tr>
<td>username</td>
<td>required, your account username</td>
</tr>
<tr>
<td>password</td>
<td>required, your account password; make sure it is URL-encoded if it contains special characters (&amp;, %, * etc.)</td>
</tr>
<tr>
<td>warn_threshold</td>
<td>a positive number (defaults to 10)</td>
</tr>
<tr>
<td>warn_threshold_enabled</td>
<td>0</td>
</tr>
<tr>
<td>warn_empty_enabled</td>
<td>0</td>
</tr>
<tr>
<td>output_format</td>
<td>XML</td>
</tr>
</tbody>
</table>
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

```
<ErrorMessage>
  <errorCode>WHOIS_01</errorCode>
  <msg>
    The parameter domainName is missing.
  </msg>
<ErrorMessage>
```

```
{
  "ErrorMessage": {
    "errorCode": "WHOIS_01",
    "msg": "The parameter domainName is missing."
  }
}
```

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

```
<ErrorMessage>
  <errorCode>AUTHENTICATE_06</errorCode>
  <msg>
    You are limited to 50 queries per second. The request is rejected.
  </msg>
<ErrorMessage>
```

```
{
  "ErrorMessage": {
    "errorCode": "AUTHENTICATE_06",
    "msg": "You are limited to 50 queries per second..."
  }
}
```

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

```xml
<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>
  </registrant>
  <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>
```

### Output continued

```xml
<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>
```
<table>
<thead>
<tr>
<th>Schema</th>
<th>Schema continued</th>
</tr>
</thead>
</table>
| "WhoisRecord": { | "customField1Name": "String",
|   "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"
|     }
|   }
| },
| "customField1Value": "String",
| "customField2Name": "String",
| "customField2Value": "String",
| "customField3Name": "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",
|   "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" |
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 cannot 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