

DNS Lookup API User Guide

A domain has a number of records associated with it, a DNS server can be queried to determine the IP address of the primary domain (A record), mail servers (MX records), DNS servers (NS nameservers) and other items such as SPF records (TXT records).

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 DNS Lookup 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 DNS Lookup API Web Service

Use the following URL:

<https://www.whoisxmlapi.com/whoisserver/DNSService?domainName=google.com&type=1&username=xxxxx&password=xxxxx>

Supported input parameters

Parameters	Values
domainName	required , the domain for which 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

Parameters	Values
digest	required unless password authentication is used, ApiKey access token
type	required , dns type: A, NS, SOA, MX, etc.; you can specify multiple comma-separated values, e.g., A,SOA,TXT; all records can be retrieved with type= _all
outputFormat	XML JSON (defaults to XML)
callback	a javascript function used when outputFormat is JSON; this is an implementation known as JSONP which invokes the callback on the returned response

Supported DNS Types

We support around fifty DNS types. You can use type codes or names from the table below:

Name		Name		Name		Name	
A	1	MG	8	MX	15	NSAP	22
NS	2	MR	9	TXT	16	NSAP_PTR	23
MD	3	NULL	10	RP	17	SIG	24
MF	4	WKS	11	AFSDB	18	KEY	25
CNAME	5	PTR	12	X25	19	PX	26
SOA	6	HINFO	13	ISDN	20	GPOS	27
MB	7	MINFO	14	RT	21	AAAA	28

Name		Name		Name		Name	
LOC	29	KX	36	IPSECKEY	45	TLSA	52
NXT	30	CERT	37	RRSIG	46	ANY	255
EID	31	A6	38	NSEC	47	DLV	32769
NIMLOC	32	DNAME	39	DNSKEY	48		
SRV	33	APL	42	DHCID	49		
ATMA	34	DS	43	NSEC3	50		
NAPTR	35	SSHFP	44	NSEC3PARAM	51		

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 DNS Lookup 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.)

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](#).

Sample Code for Making a Query to DNS Lookup API Web Service

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

User password authentication

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

Usage Limits

The maximum number of requests per second is 1. In case that the limit is breached (either from single IP address or from multiple IPs), you will be penalized with the 30 second ban. That ban includes IPs that were used at the time of the offence and includes all following IPs (based on username from additional requests) used during the ban time.

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>
---	--

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 1 query 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*

DNS_01: *Invalid type: \$type*

DNS_02: *\$errorMessage*

Sample Output (XML)

Output	Output continued
<pre> <DNSData> <domainName> google.com </domainAvailability > <types> <int> 1 </int> </types> <dnsTypes> A </dnsTypes> <audit> <createdDate> 2017-04-05 22:34:50.197 UTC </createdDate> <updatedDate> 2017-04-05 22:34:50.197 UTC </updatedDate> </audit> </pre>	<pre> <dnsRecords> <ARecord> <type> 1 </type> <dnsType> A </dnsType> <name> google.com </name> <ttl> 299 </ttl> <rRsetType> 1 </rRsetType> <rawText> google.com. 299 IN A 216.58.193.206 </rawText> <address> 216.58.193.206 </address> </ARecord> </dnsRecords> </DNSData> </pre>

Sample Output (JSON)

Output	Output continued
<pre> "DNSData": { "domainName": "google.com", "types": [1, 6, 16], "dnsTypes": "A,SOA,TXT", "audit": { "createdDate": "2017-04-05 22:37:36.822 UTC", "updatedDate": "2017-04-05 22:37:36.822 UTC" }, "dnsRecords": [{ "type": 16, "dnsType": "TXT", "name": "google.com", "ttl": 3599, "rRsetType": 16, "rawText": "google.com 3599 IN TXT \"v=spf1 include:_spf.google.com ~all\"", "strings": ["v=spf1 include:_spf.google.com ~all"] }, </pre>	<pre> { "type": 1, "dnsType": "A", "name": "google.com", "ttl": 299, "rRsetType": 1, "rawText": "google.com 299 IN A 216.58.193.206", "address": "216.58.193.206" }, { "type": 6, "dnsType": "SOA", "name": "google.com", "ttl": 59, "rRsetType": 6, "rawText": "google.com 59 IN SOA ns3.google.com", "admin": "dns-admin.google.com", "host": "ns3.google.com", "expire": 1800, "minimum": 60, "refresh": 900, "retry": 900, "serial": 152292289 }] } </pre>