

API Documentation

- Introduction
- Getting Started Using the NetDNA API
 - Authentication
 - Generate API Key and API User ID
- Request Parameters
 - authString
 - Sample authString generation code
- Sample Code
- Data Types
 - Common Data Types
 - Common Structs
- API Methods
 - List of all methods
 - PULL ZONES
 - PUSH ZONES
 - VOD ZONES
 - CACHE
 - USERS
 - Reporting
- Pull Zones
 - **API URL**
 - **pullzone.create**
 - Description
 - Method Signature
 - Parameters
 - Return Values
 - **pullzone.update**
 - Description
 - Method Signature
 - Parameters
 - Return Values
 - **pullzone.delete**
 - Description
 - Method Signature
 - Parameters
 - Return Values
 - **pullzone.listZones**
 - Description
 - Method Signature
 - Parameters
 - Return Values
- Push Zones
 - **API URL**
 - **pushzone.create**
 - Description
 - Method Signature
 - Parameters
 - **Return Values**
 - **pushzone.update**
 - Description
 - Method Signature
 - Parameters
 - **Return Values**
 - **pushzone.delete**
 - Description
 - Method Signature
 - Parameters
 - **Return Values**
 - **pushzone.listZones**
 - Description
 - Method Signature
 - Parameters
 - Return Values
- VOD Zones
 - **API URL**
 - **vodzone.create**
 - Description

- Method Signature
 - Parameters
- **vodzone.update**
 - Description
 - Method Signature
 - Parameters
- **vodzone.delete**
 - Description
 - Method Signature
 - Parameters
- **vodzone.listZones**
 - Description
 - Method Signature
 - Parameters
 - Return Values
- Cache
 - **API URL**
 - **cache.purge**
 - Description
 - Method Signature
 - Parameters
 - Return Values
 - **cache.preload**
 - Description
 - Method Signature
 - Parameters
 - Return Values
- User
 - **API URL**
 - **user.listUsers**
 - Description
 - Method Signature
 - Parameters
 - Return Values
 - **user.update**
 - Description
 - Method Signature
 - Parameters
 - Return Values
- Report
 - **API URL**
 - **report.getTotalTransfer**
 - Description
 - Method Signature
 - Parameters
 - Return Values
 - **report.getTotalHits**
 - **Description**
 - **Method Signature**
 - **Parameters**
 - **Return Values**
 - **report.getTotalTransferStats**
 - **Description**
 - **Method Signature**
 - **Parameters**
 - **Return Values**
 - **report.getCacheHitStats**
 - **Description**
 - **Method Signature**
 - **Parameters**
 - **Return Values**
 - **report.getPopularFiles**
 - **Description**
 - **Method Signature**
 - **Parameters**
 - **Return Values**
 - **report.getUsagePerDay**
 - **Description**
 - **Method Signature**
 - **Parameters**
 - **Return Values**
 - **report.getNodeHits**
 - **Description**
 - **Method Signature**

- [Parameters](#)
- [Return Values](#)

Introduction

The NetDNA [API](#) is an [XML-RPC](#) based programming interface to many of the most common functions found in the NetDNA Control Panel.

The API includes important functions to create, modify and delete zones, prime and purge cache files, list and update users and create all of the various reports.

To use the NetDNA API, you must have a NetDNA account, an API Key, an API user id and the xmlrpc library installed. These are all described in more detail below along with complete documentation of all the API methods and their parameters. Sample scripts are included later in this document to make using the API as easy as possible.

Getting Started Using the NetDNA API

If you don't already have a NetDNA CDN account visit the NetDNA website and sign up for a limited time free trial account at <https://signup.netdna.com>

Authentication

The functionality offered through the XML-RPC API requires NetDNA API user authentication.

Generate API Key and API User ID

In order to use the NetDNA Public API, developers must first generate a unique API Key and API User ID. To generate these, login to the NetDNA Control Panel, navigate to the Manage Account > API section and press the button labeled "Add API key" and a API Key and API User ID will be generated for you.

Request Parameters

All requests to the XML-RPC web services require authorization. Developers may gain access to the web services by generating an API Key and API User Id as detailed in the previous section. An `apiUserId`, `authString` and `currentDate` must be sent as a parameter with every XML-RPC method call.

- int **apiUserId** - A unique key required for each developer to access any XML-RPC method
- string **authString** - A unique authentication string required for each developer to access any XML-RPC method
- string **currentDate** - An ISO 8601 formatted date/time (e.g. 2010-02-12T15:19:21+00:00)

authString

An `authString` is a unique authentication string required for each developer to access any XML-RPC method call. It is an sha256 hash of the following string format:

```
currentDate:apiKey:method
```

where:

- **currentDate** (string) – is an ISO 8601 formatted string of the current date/time in PST
- **apiKey** (string) – is the API User Key generated through the control panel
- **method** (string) – is the XML-RPC method to invoke (without the namespace)

Sample authString generation code

```
$currentDate = date('c');

$apiKey = '1234asrx2sajsadedaxmisakaopqdfme';

$method = 'update';

$authString = hash('sha256', "$currentDate:$apiKey:$method");
```

Sample Code

To make it easy to get started, we have included a sample API script.

4.1 PHP Sample Code

```
<?php
date_default_timezone_set('America/Los_Angeles');
include("xmlrpc/lib/xmlrpc.inc");
$cur = date('c');
$apiKey = '1234ajfkepq23zxvmaoepflawqmxlawq';
$apiUserId = 199;
$method = 'purge';
$authString = hash('sha256', $cur . ':' . $apiKey . ':' . $method);

// this is the url to purge
$url= 'http://zonename.alias.netdna-cdn.com/testing/test.png';
$f=new xmlrpcmsg("cache.$method", array(php_xmlrpc_encode($apiUserId),
php_xmlrpc_encode($authString), php_xmlrpc_encode($cur),
php_xmlrpc_encode($url)));
$c=new xmlrpc_client("/xmlrpc/cache", "api.netdna.com", 80,'http11');
$r=&$c->send($f);
print_r($r);
?>
```

Data Types

Common Data Types

- array – Array of values, storing no keys
- base64 – base64 encoded binary data
- boolean – logical value (0 or 1)
- date/time – Date and time in ISO 8601 format
- double – double precision floating point number
- integer – Whole number, integer
- string – string of characters, must follow XML encoding
- struct – associative array
- nil – discriminative null value

Common Structs

1. struct **pullzone** (

- int **id**
- string **name**
- string **origin**
- string **vhost**
- string **ip**

- string **compression**
- string **vanity_domain**
- string **vanity_ip**

)

2. struct **pushzone** (

- int **id**
- string **name**
- string **vanity_domain**
- string **vanity_ip**
- string **storage_location**

)

3. struct **vodzone** (

- int **id**
- string **name**
- string **storage_location**

)

API Methods

List of all methods

Here is a simple list of the available XML-RPC API methods. A complete description is included in the following sections.

PULL ZONES

- pullzone.create
- pullzone.update
- pullzone.delete
- pullzone.listZones

PUSH ZONES

- pushzone.create
- pushzone.update
- pushzone.delete
- pushzone.listZones

VOD ZONES

- vodzone.create
- vodzone.update
- vodzone.delete
- vodzone.listZones

CACHE

- cache.purge
- cache.prime

USERS

- user.listUsers
- user.update

Reporting

- report.getTotalTransfer
 - report.getTotalHits
 - report.getTotalTransferStats
 - report.getCacheHitStats
 - report.getPopularFiles
 - report.getUsagePerDay
 - report.getNodeHits
-

Pull Zones

API URL

All XML-RPC requests should go to the following URL: <http://api.netdna.com/xmlrpc/pullzone>

pullzone.create

Description

pullzone.create - Creates a new Pull Zone on the edge servers. This type of zone caches content on the edge servers and pulls from the customer's origin server as needed.

Method Signature

```
array pullzone.create ( int apiUserId, string authString, string currentDate, struct values )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- struct **values** -- A pullzone struct
 - string **name**
 - string **origin**
 - string **vhost**
 - string **ip**
 - string **compression**
 - string **vanity_domain**

Return Values

On Success

```
array (0, struct (

- int id
- string vanity_ip

))
```

On Error

```
array ( int errorCode, string errorString )
```

- **errorCode**: 2001
 - **errorString**: Pull Zone creation failed
- **errorCode**: 2002
 - **errorString**: <name> is already in use
- **errorCode**: 2003

- errorString: <origin> does not resolve

Examples

pullzone.update

Description

pullzone.update - Updates an existing Pull Zone

Method Signature

```
int pullzone.update ( int apiUserId, string authString, string currentDate, int id, struct values )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- int **id** -- the id of the zone to update
- struct **values** -- A key/value pair that contains the new zone settings
 - string **name** (optional)
 - string **origin** (optional)
 - string **vanity_domain** (optional)
 - string **vhost** (optional)
 - string **ip** (optional)

Return Values

Returns **0** if the update is a success, **1** if otherwise

Examples

pullzone.delete

Description

pullzone.delete – Deletes an existing Pull Zone

Method Signature

```
int pullzone.delete (int apiUserId, string authString, string currentDate, int id)
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- int **id** -- the id of the zone to delete

Return Values

Returns **0** if delete is a success, **1** if otherwise.

Examples

pullzone.listZones

Description

pullzone.listZones - Lists all Pull Zones from an account

Method Signature

```
array pullzone.listZones ( int apiUserId, string authString, string currentDate )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date

Return Values

An array of pullzones

Examples

Push Zones

API URL

All XML-RPC requests should go to the following URL: <http://api.netdna.com/xmlrpc/pushzone>

pushzone.create

Description

pushzone.create - Creates a new Push Zone.

Method Signature

```
array pushzone.create ( int apiUserId, string authString, string currentDate )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- struct **values** -- a pushzone struct
 - string **name**
 - string **password**
 - string **vanity_domain**

Return Values

On Success

array (0, struct (

- int **id**
- string **vanity_ip**
- string **storage_location**

)

)

On Error

array (int **errorCode**, string **errorMessage**)

- errorCode: 3001
 - errorString: Push Zone creation failed
- errorCode: 3002
 - errorString: <name> is already in use

Examples

pushzone.update

Description

pushzone.update – Updates an existing Push Zone

Method Signature

```
int pushzone.update ( int apiUserId, string authString, string currentDate, struct values )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- int **id** -- the id of the zone to update
- struct **values**
 - string **name** (optional)
 - string **password** (optional)
 - string **vanity_domain** (optional)

Return Values

Returns **0** if the update is a success, **1** if otherwise

Examples

pushzone.delete

Description

pushzone.delete – Deletes an existing Push Zone

Method Signature

```
int pushzone.delete ( int apiUserId, string authString, string currentDate, int id )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- int **id** -- the id of the zone to delete

Return Values

Returns **0** if delete is a success, **1** if otherwise

Examples

pushzone.listZones

Description

pushzone.listZones – Lists all Push Zones from an account

Method Signature

```
array pushzone.listZones ( int apiUserId, string authString, string currentDate )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date

Return Values

An array of pushzones

Examples

VOD Zones

API URL

All XML-RPC requests should go to the following URL: <http://api.netdna.com/xmlrpc/vodzone>

vodzone.create

Description

vodzone.create - Create a new VOD Zone.

Method Signature

```
array vodzone.create ( int apiUserId, string authString, string currentDate, struct values )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- struct **values**
 - string **name**
 - string **password**

Return Values

On Success

```
array (0, struct (

- int id
- int vanity_ip
- string storage_location

)
```

On Error

```
array ( int errorCode, string errorMessage )

- errorCode: 4001
  - errorMessage: VOD Zone creation failed
- errorCode: 4002
  - errorMessage: <name> is already in use

```

Examples

vodzone.update

Description

vodzone.update – Updates an existing Vod Zone

Method Signature

```
int vodzone.update ( int apiUserId, string authString, string currentDate, int id, struct values )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- int **id** - the id of the zone to update
- struct **values**
 - string **name** (optional)
 - string **password** (optional)

Return Values

Returns **0** if the update is a success, **1** if otherwise

Examples

vodzone.delete

Description

vodzone.delete – Deletes an existing Vod Zone

Method Signature

```
int vodzone.delete ( int apiUserId, string authString, string currentDate, int id )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- int **id** -- the id of the zone to delete

Return Values

Returns **0** if delete is a success, **1** if otherwise

Examples

vodzone.listZones

Description

vodzone.listZones – Lists all Vod Zones from an account

Method Signature

```
array vodzone.listZones ( int apiUserId, string authString, string currentDate )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date

Return Values

An array of vodzones

Examples

Cache

The methods to purge and preload cache are listed in this section. These methods use the following API url:

API URL

All XML-RPC requests should go to the following URL: <http://api.netdna.com/xmlrpc/cache>

cache.purge

Description

cache.purge - Purges a file from cache so it is pulled from the origin server the next time it is requested.

Method Signature

```
boolean cache.purge ( int apiUserId, string authString, string currentDate, string url )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- string **url** -- url of the file to purge from cache

Return Values

Returns **TRUE** if purging is successful, **FALSE** if otherwise.

Examples

cache.preload

Description

cache.preload - Loads a file into cache so this version is retrieved from cache and not from the origin server the next time it's requested.

Method Signature

```
array cache.preload ( int apiUserId, string authString, string currentDate, string url )
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date
- string **url** -- url of file to preload into cache

Return Values

array (statusCode, statusString)

- int **statusCode** – 0 if preload is a success, 1 if otherwise
- string **statusString** – “**Success**” if preload is a success, “**Failed**” if otherwise

Examples

User

API URL

All XML-RPC requests should go to the following URL: <http://api.netdna.com/xmlrpc/user>

user.listUsers

Description

user.listUsers - Lists all users on an account

Method Signature

```
array user.listUsers ( int apiUserId, string authString, string currentDate)
```

Parameters

- int **apiUserId** -- the API user id
- string **authString** -- used for server validation
- string **currentDate** -- ISO 8601 formatted date

Return Values

Array of users

Examples

user.update

Description

user.update - Modifies user information

Method Signature

```
array user.update ( int userId, string authString, string currentDate, int id, struct values )
```

Parameters

- int **apiUserId** - the API user id
- string **authString** - used for server validation
- string **currentDate** - ISO 8601 formatted date
- int **id** - the id of the user to update
- struct values - The key/value pair for the new user settings

Return Values

array

Examples

Report

API URL

All XML-RPC requests should go to the following URL: <http://api.netdna.com/xmlrpc/report>

report.getTotalTransfer

Description

report.getTotalTransfer - Returns the total bandwidth transfer for a given Zone ID and date range

Method Signature

```
double report.getTotalTransfer ( int apiUserId, string authString, string currentDate, int zoneId, int type = null, string from = null, string to = null )
```

Parameters

- int **apiUserId**
- string **authString**
- string **currentDate**
- int **zoneId** - the Zone ID
- int **type** - 1=today, 2=current hour, 3=date range
- string **from** - starting date
- string **to** - end date

Return Values

array

report.getTotalHits

Description

report.getTotalHits - Get total hits for a given Zone ID and date range

Method Signature

```
int report.getTotalHits ( int apiUserId, string authString, string currentDate, int zoneId, int type = null, string from = null, string to = null )
```

Parameters

- int **apiUserId**
- string **authString**
- string **currentDate**
- int **zoneId** - the Zone ID
- int **type** - 1=current date, 2=current hour, 3=date range
- string **from** - start date
- string **to** - end date

Return Values

array

report.getTotalTransferStats

Description

report.getTotalTransferStats - Returns transfer stats for a given company/zone and date range

Method Signature

```
array report.getTotalTransferStats ( int apiUserId, string authString, string currentDate, int companyId, string dateFrom, string dateTo, int zoneId = NULL, array sortBy = array(), string viewby = "daily", int maxReturn = NULL, string offset = NULL, string timezone = "GMT" )
```

Parameters

- int **apiUserId**
- string **authString**
- string **currentDate**
- int **companyId**
- string **dateFrom** - start date
- string **dateTo** - end date
- int **zoneId**
- array **sortBy**
- string **viewby**
- int **maxReturn**
- string **offset**
- string **timezone**

Return Values

array

report.getCacheHitStats

Description

report.getCacheHits - Returns the total cache hits for a given company/zone and date range

Method Signature

```
int report.getCacheHitStats ( int apiUserId, string authString, string currentDate, int companyId,
string dateFrom, string dateTo, int zoneId = NULL, array sortBy = array(), int maxReturn = NULL, int
offset = NULL, string timezone = "GMT" )
```

Parameters

- int **apiUserId**
- string **authString**
- string **currentDate**
- int **companyId**
- string **dateFrom** - start date
- string **dateTo** - end date
- int **zoneId** - the Zone ID
- array **sortBy**
- int **maxReturn**
- int **offset**
- string **timezone**

Return Values

int

report.getPopularFiles

array report.getPopularFiles - Returns a list of popular files for a given company/zone and date range

Description

Method Signature

```
array report.getPopularFiles ( int apiUserId, string authString, string currentDate, int companyId,
string dateFrom, string dateTo, int zoneId = NULL, array sortBy = array('hits DESC'), int maxReturn =
NULL, int offset = NULL )
```

Parameters

- int **apiUserId**
- string **authString**
- string **currentDate**
- int **companyId**
- string **dateFrom** - start date
- string **dateTo** - end date
- int **zoneId** - the Zone ID
- array **sortBy**
- int **maxReturn**
- int **offset**

Return Values

array

report.getUsagePerDay

Description

report.getUsagePerDay - Returns usage stats for a give company/zone and date range

Method Signature

```
int report.getUsagePerDay ( int apiUserId, string authString, string currentDate, int companyId,
string dateFrom, string dateTo, int zoneId, array sortBy = array(), int maxReturn = NULL, int offset =
NULL )
```

Parameters

- int **apiUserId**
- string **authString**
- string **currentDate**
- int **companyId**
- string **dateFrom** - start date
- string **dateTo** - end date
- int **zoneId** - the Zone ID
- array **sortBy**
- int **maxReturn**
- int **offset**

Return Values

int

report.getNodeHits

Description

report.getNodeHits - returns a list of node hits for a given company/zone and date range

Method Signature

```
array report.getNodeHits ( int apiUserId, string authString, string currentDate, int companyId, string dateFrom, string dateTo, int zoneId = NULL, array sortBy = array(), int maxReturn = NULL, int offset = NULL )
```

Parameters

- int **apiUserId**
- string **authString**
- string **currentDate**
- int **companyId**
- string **dateFrom** - start date
- string **dateTo** - end date
- int **zoneId** - the Zone ID
- array **sortBy**
- int **maxReturn**
- int **offset**

Return Values

array