

DNS Made Easy REST API Documentation v1.2

Overview

Making a request

REST requests can be made via HTTPS or HTTP by using the headers, authentication, data types, and methods specified below. The current endpoint for the API is available at [http\[s\]://api.dnsmadeeasy.com/V1.2/](http[s]://api.dnsmadeeasy.com/V1.2/)

Rate limiting

To prevent unwanted flooding of the API system, there is a maximum number of requests that can be sent in a given time period. This limit is tracked per API key and all requests count toward this limit. Refer to the *x-dnsme-requestLimit* and *x-dnsme-requestsRemaining* header fields for values related to this limit.

Data formats

The DNS Made Easy API supports both XML and JSON data formats, specified by using the *content-type* and *accept* HTTP header fields. If no format is specified, JSON will be used as a default.

Try it!

If you'd like to make an API request, just follow these steps. We've provided a sample Perl script that can be used to make requests.

- Download and install [cURL](#). Make sure that the cURL executable is part of your path.
- Download and install [Perl](#). Make sure that the Perl executable is part of your path.
- Download the necessary Perl extensions by running the following commands:

```
perl -MCPAN -e "install Digest::HMAC_SHA1"
perl -MCPAN -e "install HTTP::Date"
perl -MCPAN -e "install Config::Properties"
```

- Download the following DNS Made Easy files and save them into the same location:
 - [dnsmeapi.properties](#)
 - [dnsmeapi.pl](#)
- Put your API and Secret Keys into the dnsmeapi.properties file
- Make requests using the dnsmeapi.pl script as a wrapper around cURL! For example:

```
perl dnsmeapi.pl http://api.dnsmadeeasy.com/V1.2/domains/ -H
accept:application/xml
```

```
perl dnsmeapi.pl http://api.dnsmadeeasy.com/V1.2/domains/example.com -X PUT
```

```
perl dnsmeapi.pl http://api.dnsmadeeasy.com/V1.2/domains/example.com/records
-H accept:application/json
```

cURL accepts the **-X** option to set the HTTP method used for the request (GET is used if no method is specified). cURL accepts the **-H** option to set HTTP headers for the request.

Implementation

Authentication

Authentication with the DNS Made Easy API is done using the API and Secret keys, given on a per-account basis. The values for these keys can be found on the Advanced DNS settings page after you log in. To make an authenticated request, follow these steps:

- Create the string representation of the current date and time in HTTP format. Example:
Sat, 12 Feb 2011 20:59:04 GMT
- Calculate the hexadecimal HMAC SHA1 hash of that string using your Secret key as the hash key. Example:
b3502e6116a324f3cf4a8ed693d78bcee8d8fe3c
- Set the values for the request headers using your API key, the current date and time, and the HMAC hash that you calculated. This example was created using a Secret key of c9b5625f-9834-4ff8-baba-4ed5f32cae55:
 - x-dnsme-apiKey:1c1a3c91-4770-4ce7-96f4-54c0eb0e457a
 - x-dnsme-requestDate:Sat, 12 Feb 2011 20:59:04 GMT
 - x-dnsme-hmac:b3502e6116a324f3cf4a8ed693d78bcee8d8fe3c

Requests must be sent shortly after these headers are generated. If too much time has passed between when the *x-dnsme-requestDate* and *x-dnsme-hmac* strings were created and when the request is received by the DNS Made Easy API servers, then the request will be denied.

Requests made with invalid credentials or an invalid *x-dnsme-requestDate* value will receive an HTTP 403 – Forbidden response.

Common header fields

The DNS Made Easy API includes several custom HTTP header fields that contain information about the requests and responses that are sent. These headers fields are:

Request Header Fields	
Field name	Description
x-dnsme-apiKey	The API Key for your account. Refer to the Advanced DNS page after logging in to find the value for this field.
x-dnsme-requestDate	Standard HTTP-formatted date. This date is used to protect against falsified requests played back at a future time.
x-dnsme-hmac	HMAC hash of value of the x-dnsme-requestDate field. Refer to the

authentication section for more details on how to generate this value.

Response Header Fields

Field name	Description
x-dnsme-requestId	A unique identifier of the request that was sent. Use the value of this header to help identify your request for your own purposes or when contacting DNS Made Easy support.
x-dnsme-requestLimit	Maximum number of requests that can be sent before the rate limit is exceeded.
x-dnsme-requestsRemaining	Number of requests remaining before the rate limit is exceeded.

Data types

The following data types are used for requests and responses with the DNS Made Easy API:

Domain

The domain data type contains information about a domain or hosted zone. The following fields are available:

Field Name	Type	Description
name	string	The domain name.
nameServer	list of strings	Name servers associated with this domain.
gtdEnabled	boolean	Indicator of whether or not this domain uses the Global Traffic Director.

Example XML Representation:

```
<domain>
<name>myDomain.com</name>
<nameServer>ns10.dnsmadeeasy.com</nameServer>
<nameServer>ns11.dnsmadeeasy.com</nameServer>
<nameServer>ns12.dnsmadeeasy.com</nameServer>
<nameServer>ns13.dnsmadeeasy.com</nameServer>
<nameServer>ns14.dnsmadeeasy.com</nameServer>
<nameServer>ns15.dnsmadeeasy.com</nameServer>
<gtdEnabled>true</gtdEnabled>
</domain>
```

Example JSON Representation:

```
{
  "name": "myDomain.com",
  "nameServer": [
    "ns10.dnsmadeeasy.com",
    "ns11.dnsmadeeasy.com",
    "ns12.dnsmadeeasy.com",
    "ns13.dnsmadeeasy.com",
```

```
"ns14.dnsmadeeasy.com",
"ns15.dnsmadeeasy.com"
],
"gtdEnabled":true
}
```

Record

The record data type contains information about a record for a given domain. The following fields are available:

Field Name	Type	Description
id	number	Unique record identifier
name	string	Record name.
type	string	Record type. Values: A, AAAA, CNAME, HTTPRED, MX, NS, PTR, SRV, TXT
data	string	Record data. Content varies based on record type: A: <host IP> AAAA: <IPv6 host IP> CNAME: <target name> HTTPRED: <redirection URL> MX: <priority> <target name> NS: <name server> PTR: <target name> SRV: <priority> <weight> <port> <target name> TXT: <text value>
ttl	number	Time to live. The amount of time a record will be cached before being refreshed.
gtdLocation	string	Global Traffic Director location. Values: DEFAULT, US_EAST, US_WEST, EUROPE
password	string	For A records. Password used to authenticate for dynamic DNS.
description	string	For HTTPRED records. A description of the HTTPRED record.
keywords	string	For HTTPRED records. Keywords associated with the HTTPRED record.
title	string	For HTTPRED records. The title of the HTTPRED record.
redirectType	string	For HTTPRED records. Type of redirection performed. Values: Hidden Frame Masked, Standard – 302, Standard – 301
hardLink	boolean	For HTTPRED records.

Example XML Representation:

```
<record>
<data>208.94.147.96</data>
<gtdLocation>Default</gtdLocation>
<id>123456</id>
<name>dummy</name>
<ttl>1800</ttl>
<type>A</type>
</record>
```

```
<record>
<data>10 mail.myDomain.com.</data>
<gtdLocation>Default</gtdLocation>
<id>123456</id>
<name>dummy2</name>
<ttd>1800</ttd>
<type>MX</type>
</record>
```

Example JSON Representations:

```
{
  "name": "dummy",
  "id": 123456,
  "type": "A",
  "data": "208.94.147.96",
  "gtdLocation": "Default",
  "ttd": 1800
}

{
  "name": "dummy2",
  "id": 123456,
  "type": "MX",
  "data": "10 mail.myDomain.com.",
  "gtdLocation": "Default",
  "ttd": 1800
}
```

Secondary

The secondary data type contains information about a secondary domain entry. The following fields are available:

Field Name	Type	Description
name	string	The name of the secondary entry.
ip	list of strings	IP addresses for your master nameserver associated with this secondary entry.

Example XML Representation:

```
<secondary>
<name>myDomain.com</name>
<ip>10.10.10.10</ip>
<ip>10.10.10.11</ip>
<ip>10.10.10.12</ip>
</secondary>
```

Example JSON Representation:

```
{
  "name": "myDomain.com",
  "ip": [
    "10.10.10.10",
    "10.10.10.11",
```

```
"10.10.10.12"  
]  
}
```

Resources and methods:

The following resources and methods are available with the DNS Made Easy API:

/domains – operate on all domains for your account

Method	HTTP Status Codes	Description
GET	200 – OK	Returns a list of all domain names for your account.
DELETE	200 – OK	Deletes all domains for your account.

/domains/{domainName} – operate on the domain with the specified name

Method	HTTP Status Codes	Description
GET	200 – OK 404 – specified domain name is not found	Returns the domain object representation of the specified domain.
DELETE	200 – OK 404 – specified domain name is not found	Deletes the specified domain.
PUT	201 – domain successfully created 400 – domain name not valid, see errors in response content	Creates a domain entry with the specified name. Returns errors if name is not valid or conflicts with another domain.

/domains/{domainName}/records – operate on the full set of records for the specified name

Method	HTTP Status Codes	Description
GET	200 – OK 404 – specified domain name is not found	Returns a list of record objects containing all records for the specified domain. May use type and gtdLocation as query parameters. Example: <code>http://api.dnsmadeeasy.com/V1.2/domains/myDomain/records?gtdLocation=US_EAST&type=A</code>
POST	201 – record successfully created 400 – record not valid, see errors in response content	Creates a record with the representation specified in the request content. Returns errors if record is not valid. Note that a record ID will be generated by the system with this request and any ID that is sent will be ignored. Records are not modifiable for domains that are locked to a template.

/domains/{domainName}/records/{recordId} – operate on a record having the specified id for the specified domain name

Method	HTTP Status Codes	Description
GET	200 – OK 404 – specified domain name or record id is not found	Returns a record object representing the record with the specified id.
DELETE	200 – OK	Deletes the record with the specified id. Note that

404 – specified domain name or record id is not found	records are not modifiable for domains that are locked to a template.
---	---

/secondary – operate on all secondary entries for your account

Method	HTTP Status Codes	Description
GET	200 – OK	Returns a list of all secondary entry names for your account.
DELETE	200 – OK	Deletes all secondary entries for your account.

/secondary/{secondaryName} – operate on the secondary entry with the specified name

Method	HTTP Status Codes	Description
GET	200 – OK 404 – specified secondary entry name is not found	Returns the secondary entry object representation of the specified secondary entry.
DELETE	200 – OK 404 – specified secondary entry name is not found	Deletes the specified secondary entry.
PUT	201 – secondary entry successfully created or modified 400 – secondary entry name or IP addresses not valid, see errors in response content	Creates or modifies a secondary entry with the specified name. Returns errors if name is not valid or conflicts with another domain