

Table of Contents

1 Dns-zone:GET	1
2 GET /dns/{zone}	2
2.1 Contents	2
2.2 Request	2
2.3 Response	2
2.4 Examples	2
2.5 See also	3
3 Dns-zone-records:GET	4
4 GET /dns/{zone}/records	5
4.1 Contents	5
4.2 Request	5
4.3 Response	6
4.4 Examples	8
4.5 See also	9
5 Dns-zone-records:POST	10
6 POST /dns/{zone}/records	11
6.1 Contents	11
6.2 Request	11
6.3 Response	13
6.4 Examples	13
6.5 See also	14
7 Dns-zone-records-recordId:PUT	15
8 PUT /dns/{zone}/records/{recordId}	16
8.1 Contents	16
8.2 Request	16
8.3 Response	17
8.4 Examples	17
8.5 See also	18
9 Dns-zone-records-recordId:DELETE	19
10 DELETE /dns/{zone}/records/{recordId}	20
10.1 Contents	20
10.2 Request	20
10.3 Request Headers	20
10.4 Response	20
10.5 Examples	20
10.6 See also	20
11 Dns-heartbeat:GET	21
12 GET /dns/heartbeat	22
12.1 Contents	22
12.2 Request	22
12.3 Response	22
12.4 See also	22
13 Dns-healthcheck:GET	23
14 GET /dns/healthcheck	24
14.1 Contents	24
14.2 Request	24
14.3 Response	24
14.4 See also	24
15 Common Features	25

1 Dns-zone:GET

2 GET /dns/{zone}

Returns the details of a DNS zone.

2.1 Contents

- 1 GET /dns/{zone}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Parameters
 - ◇ 1.1.2 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - ◆ 1.3 Examples
 - ◆ 1.4 See also

2.2 Request

GET /dns/{zone}

2.2.1 Parameters

zone - *string*
The name of the DNS zone.

2.2.2 Request Headers

Authorization
Base64-encoded username:password string

2.3 Response

2.3.1 Status Code

- 200 OK
Success
- 401 Unauthorized
The server could not verify the credentials from the authorization header.
- 403 Forbidden
The server denies the access to the resource.
- 404 Not Found
The specified zone does not exist.

2.3.2 Response Body

```
{
  "minimumTTL": {minimumTTL},
  "retry": {retry},
  "name": "{zone}",
  "primaryNS": "{primaryNS}",
  "refresh": {refresh},
  "expire": {expire},
  "emailAddress": "{emailAddress}",
  "ttl": {ttl},
  "serial": {serial},
  "links": [
    {"href": "https://api.hostway.com/dns/{zone}", "rel": "self"},
    {"href": "https://api.hostway.com/dns/{zone}/records", "rel": "related"}
  ]
}
```

2.4 Examples

Request:

GET /dns/test.com

Response:

401 Unauthorized

Request:

```
GET /dns/unknown-zone.com
Authorization: Basic {base64_encoded(username:password)}
```

Response:

403 Forbidden

```
{
  "forbidden": {"message": "Unauthorized: Dns failed permission check", "code": 403, "details": ""}
}
```

Request:

```
GET /dns/test.com
Authorization: Basic {base64_encoded(username:password)}
```

Response:

200 OK

```
{
  "minimumTTL": 300,
  "retry": 7200,
  "name": "test.com.",
  "links": [
    {"href": "http://127.0.0.1/dns/test.com/", "rel": "self"},
    {"href": "http://127.0.0.1/dns/test.com/records", "rel": "related"}
  ],
  "primaryNS": "a.dns.hostway.net.",
  "refresh": 28800,
  "expire": 604800,
  "emailAddress": "hostmaster.siteprotect.com.",
  "ttl": 14400,
  "serial": 1359151749
}
```

2.5 See also

- [DNS API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

3 Dns-zone-records:GET

4 GET /dns/{zone}/records

Returns the DNS zone records.

4.1 Contents

- 1 GET /dns/{zone}/records
 - ◆ 1.1 Request
 - ◇ 1.1.1 Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - 1.2.2.2 Record Structures
 - 1.2.2.2.1 A record
 - 1.2.2.2.2 AAAA record
 - 1.2.2.2.3 NS record
 - 1.2.2.2.4 TXT and SPF records
 - 1.2.2.2.5 SRV record
 - 1.2.2.2.6 MX record
 - 1.2.2.2.7 CNAME record
 - ◆ 1.3 Examples
 - ◇ 1.3.1 Get a list of DNS records
 - ◇ 1.3.2 Get a paginated list of DNS records
 - ◇ 1.3.3 Get a filtered list of DNS records
 - ◆ 1.4 See also

4.2 Request

GET /dns/{zone}/records

4.2.1 Parameters

zone - *string*
The name of the DNS zone.

4.2.1.1 URI Parameters

filterType
Optional. Specifies the type of records to return: *A*, *AAAA*, *NS*, *TXT*, *SRV*, *MX*, *CNAME*

page - *number*
Optional. Specifies the page to return in paginated response. Default is 1.

pageSize - *number*
Optional. Specifies the number of records per page. Default 10, Max 100.

4.2.2 Request Headers

Authorization
Base64-encoded username:password string

4.3 Response

4.3.1 Status Code

200 OK
Success

401 Unauthorized
The server could not verify the credentials from the authorization header.

404 Not Found
The specified zone does not exist.

4.3.2 Response Body

```
{
  "list": [
    {
      "id": {id},
      "name": "{domain}",
      "type": "AAAA",
      "ttl": {ttl},
      "address": "{ipv6_address}",
      "links": [
        { "href": "https://api.hostway.com/dns/{zone}/records/{id}/", "rel": "self" }
      ]
    },
    {
      "id": {id},
      "name": "{domain}",
      "type": "A",
      "ttl": {ttl},
      "address": "{ipv4_address}",
      "links": [
        { "href": "https://api.hostway.com/dns/{zone}/records/{id}/", "rel": "self" }
      ]
    }
  ],
  "links": [
    { "href": "https://api.hostway.com/dns/{zone}/records/", "rel": "self" }
  ]
}
```

4.3.2.1 Parameters

list - list
list of DNS records

id - integer
The record's ID

id - string
The record's type. Possible values are *A*, *AAA*, *NS*, *CNAME*, *MX*, *SVR* or *TXT*. Both SPF and TXT records will appear as *TXT* and share the same record structure.

ttl - integer
DNS record's TTL in seconds

links - list
[Hypermedia](#) to a particular DNS record

other type-specific parameters (optional)
contains type-specific data. See [Record Structures](#) for more information.

links - list
[Hypermedia](#) for the current DNS records list resource. May include links with information for the pagination.

4.3.2.2 Record Structures

The goal of this section is to show the presentation of each DNS record in the response body based on the the record's type. The structure of each DNS record depends only on the record's type i.e. **A**, **AAA**, **NS**, **CNAME**, **MX**, **SVR** or **TXT**.

4.3.2.2.1 A record

```
{
  "id": {record_id},
  "name": "{domain}",
  "type": "A",
  "address": "{ip_address}",
  "ttl": 3600,
  "links": [
    {
      "href": "{server_name}/dns/{zone}/records/{record_id}",
      "rel": "self"
    }
  ]
}
```

4.3.2.2.2 AAAA record

```
{
  "id": {record_id},
  "name": "{domain}",
  "type": "AAAA",
  "address": "{ipv6_address}",
  "ttl": 3600,
  "links": [
    {
      "href": "{server_name}/dns/{zone}/records/{record_id}",
      "rel": "self"
    }
  ]
}
```

4.3.2.2.3 NS record

```
{
  "id": {record_id},
  "name": "{domain}",
  "type": "NS",
  "nsdname": "{ns_domain}",
  "ttl": 3600,
  "links": [
    {
      "href": "{server_name}/dns/{zone}/records/{record_id}",
      "rel": "self"
    }
  ]
}
```

4.3.2.2.4 TXT and SPF records

```
{
  "id": {record_id},
  "name": "{domain}",
  "type": "TXT",
  "data": "{data}",
  "ttl": 3600,
  "links": [
    {
      "href": "{server_name}/dns/{zone}/records/{record_id}",
      "rel": "self"
    }
  ]
}
```

Note: SPF records also appear as TXT records.

4.3.2.2.5 SRV record

```
{
  "id": {record_id},
  "name": "{domain}",
  "type": "SRV",
  "priority": {priority},
  "weight": {weight},
  "port": {port},
  "target": "{target}",
  "ttl": 3600,
  "links": [
    {
      "href": "{server_name}/dns/{zone}/records/{record_id}",
      "rel": "self"
    }
  ]
}
```

4.3.2.2.6 MX record

```
{
  "id": {record_id},
  "name": "{domain}",
  "type": "MX",
  "preference": {preference},
  "exchange": "{mx_domain}",
  "ttl": 3600,
  "links": [
    {
      "href": "{server_name}/dns/{zone}/records/{record_id}",
      "rel": "self"
    }
  ]
}
```

4.3.2.2.7 CNAME record

```
{
  "id": {record_id},
  "name": "{name}",
  "type": "CNAME",
  "cname": "{domain}",
  "ttl": 3600,
  "links": [
    {
      "href": "{server_name}/dns/{zone}/records/{record_id}",
      "rel": "self"
    }
  ]
}
```


4.4 Examples

4.4.1 Get a list of DNS records

By default, the DNS records list is paginated to 10 records per page. If the page is not specified, first page will be returned.

GET <http://servername/dns/ng-test-hwus-petre-101202.com/records>

```
{
  "list" : [ {
    "name" : "ftp.ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/10865/",
      "rel" : "self"
    } ],
    "ttl" : 14400,
    "address" : "10.10.100.60",
    "type" : "A",
    "id" : 10865
  }, {
    "name" : "ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/10866/",
      "rel" : "self"
    } ],
    "ttl" : 14400,
    "address" : "10.10.100.60",
    "type" : "A",
    "id" : 10866
  }, {
    "name" : "www.ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/10864/",
      "rel" : "self"
    } ],
    "ttl" : 14400,
    "address" : "10.10.100.60",
    "type" : "A",
    "id" : 10864
  }, {
    "name" : "test-ipv6.ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/179100/",
      "rel" : "self"
    } ],
    "ttl" : 3000,
    "address" : "2001:0db8:85a3:0042:1000:8a2e:0370:7334",
    "type" : "AAAA",
    "id" : 179100
  }, {
    "name" : "sourcename.ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/179042/",
      "rel" : "self"
    } ],
    "cname" : "destinationdomain.com",
    "ttl" : 14400,
    "type" : "CNAME",
    "id" : 179042
  }, {
    "name" : "mail.ng-test-hwus-petre-101202.com.ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/179043/",
      "rel" : "self"
    } ],
    "exchange" : "mail.destination.com.ng-test-hwus-petre-101202.com",
    "preference" : 3,
    "ttl" : 14400,
    "type" : "MX",
    "id" : 179043
  }, {
    "name" : "ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/10860/",
      "rel" : "self"
    } ],
    "nsdname" : "vpdns.debian.hps.hostway",
    "ttl" : 14400,
    "type" : "NS",
    "id" : 10860
  }, {
    "name" : "ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/10861/",
      "rel" : "self"
    } ],
    "nsdname" : "vpdns2.debian.hps.hostway",
    "ttl" : 14400,
    "type" : "NS",
    "id" : 10861
  }, {
    "name" : "_service1._tcp.ng-test-hwus-petre-101202.com",
    "weight" : 3,
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/179041/",
      "rel" : "self"
    } ],
    "port" : 80,
    "priority" : 2,
    "ttl" : 14400,
    "type" : "SRV",
    "id" : 179041,
    "target" : "svc-dest.ng-test-hwus-petre-101202.com"
  }, {
```

```

"name" : "spf-test.ng-test-hwus-petre-101202.com",
"links" : [ {
  "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/179065/",
  "rel" : "self"
} ],
"data" : "v=spf1 exists:example.com -all",
"ttl" : 14400,
"type" : "TXT",
"id" : 179065
} ],
"links" : [ {
  "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/",
  "rel" : "self"
}, {
  "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records?page=2",
  "rel" : "next"
}, {
  "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records?page=2",
  "rel" : "last"
} ]
}

```

4.4.2 Get a paginated list of DNS records

By default, the DNS records list is paginated to 10 records per page, if the **pageSize** parameter is not specified.

GET <http://servername/dns/ng-test-hwus-petre-101202.com/records?page=2>

```

{
  "list" : [ {
    "name" : "test-text.ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/179044/",
      "rel" : "self"
    } ],
    "data" : "test text",
    "ttl" : 14400,
    "type" : "TXT",
    "id" : 179044
  } ],
  "links" : [ {
    "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records?page=2",
    "rel" : "self"
  }, {
    "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records?page=1",
    "rel" : "first"
  }, {
    "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records?page=1",
    "rel" : "previous"
  } ]
}

```

4.4.3 Get a filtered list of DNS records

Filtered by type.

GET <http://servername/dns/ng-test-hwus-petre-101202.com/records?filterType=TXT>

```

{
  "list" : [ {
    "name" : "spf-test.ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/179065/",
      "rel" : "self"
    } ],
    "data" : "v=spf1 exists:example.com -all",
    "ttl" : 14400,
    "type" : "TXT",
    "id" : 179065
  }, {
    "name" : "test-text.ng-test-hwus-petre-101202.com",
    "links" : [ {
      "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records/179044/",
      "rel" : "self"
    } ],
    "data" : "test text",
    "ttl" : 14400,
    "type" : "TXT",
    "id" : 179044
  } ],
  "links" : [ {
    "href" : "http://servername/dns/ng-test-hwus-petre-101202.com/records?filterType=TXT",
    "rel" : "self"
  } ]
}

```

4.5 See also

- [DNS API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

5 Dns-zone-records:POST

6 POST /dns/{zone}/records

Creates new DNS zone records.

6.1 Contents

- 1 POST /dns/{zone}/records
 - ◆ 1.1 Request
 - ◇ 1.1.1 Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 A record
 - 1.1.4.2 AAAA record
 - 1.1.4.3 NS record
 - 1.1.4.4 TXT record
 - 1.1.4.5 SRV record
 - 1.1.4.6 MX record
 - 1.1.4.7 CNAME record
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Header
 - ◆ 1.3 Examples
 - ◇ 1.3.1 Successful Request
 - ◇ 1.3.2 Unauthorized Request
 - ◇ 1.3.3 Attempt to create duplicate record
 - ◆ 1.4 See also

6.2 Request

POST /dns/{zone}/records

6.2.1 Parameters

zone - *string*
The name of the DNS zone.

6.2.1.1 URI Parameters

None.

6.2.2 Request Headers

Authorization
Base64-encoded username:password string

Content-Type
application/json

6.2.3 Request Body

Added support for wildcard domain names.
Ex:
*
*.somedomain.com
The wildcard record name is stored in the database as it is supplied.
If it's missing the zone origin(i.e. '') the API will return it with the zone

appended, i.e. '*.domain.com'.

Added support for @ records, which represent the origin itself.

Ex:

@

The API will return the record with the zone origin appended to the name, i.e. '@.domain.com'

6.2.3.1 A record

```
{
  "name": "{domain}",
  "type": "A",
  "address": "{ip_address}",
  "ttl": 3600
}
```

name Fully qualified domain name (FQDN) of the record

address IPv4 address

ttl Time to live (TTL) for the record in seconds

6.2.3.2 AAAA record

```
{
  "name": "{domain}",
  "type": "AAAA",
  "address": "{ipv6_address}",
  "ttl": 3600
}
```

name Fully qualified domain name (FQDN) of the record

address IPv6 address

ttl Time to live (TTL) for the record in seconds

6.2.3.3 NS record

```
{
  "name": "{domain}",
  "type": "NS",
  "nsdname": "{ns_domain}",
  "ttl": 3600
}
```

name Fully qualified domain name (FQDN) of the record

nsdname FQDN of the host which should be authoritative for the domain

ttl Time to live (TTL) for the record in seconds

6.2.3.4 TXT record

```
{
  "name": "{domain}",
  "type": "TXT",
  "data": "{data}",
  "ttl": 3600
}
```

name Fully qualified domain name (FQDN) of the record

data TXT record data

ttl Time to live (TTL) for the record in seconds

6.2.3.5 SRV record

```
{
  "name": "{domain}",
  "type": "SRV",
  "priority": {priority},
  "weight": {weight},
  "port": {port},
  "target": "{target}",
  "ttl": 3600
}
```

name Fully qualified domain name (FQDN) of the record

priority

	The priority of this target host
weight	Server selection mechanism
port	The port on this target host of this service
target	The domain name of the target host
ttl	Time to live (TTL) for the record in seconds

6.2.3.6 MX record

```
{
  "name": "{domain}",
  "type": "MX",
  "preference": {preference},
  "exchange": "{mx_domain}",
  "ttl": 3600
}
```

name	Fully qualified domain name (FQDN) of the record
preference	Integer which specifies the preference given to this record among others at the same owner
exchange	FQDN which specifies a host willing to act as a mail exchange for the owner name
ttl	Time to live (TTL) for the record in seconds

6.2.3.7 CNAME record

```
{
  "name": "{name}",
  "type": "CNAME",
  "cname": "{domain}",
  "ttl": 3600
}
```

name	Fully qualified domain name (FQDN) of the record
cname	FQDN which specifies the canonical or primary name for the owner
ttl	Time to live (TTL) for the record in seconds

6.3 Response

6.3.1 Status Code

201 OK	Success
400 Bad Request	TTL is below allowed minimum
401 Unauthorized	The server could not verify the credentials from the authorization header.
404 Not Found	The specified zone does not exist.
409 Conflict	Requested record conflicts with existing one.

6.3.2 Response Header

Location	The URI of the new DNS record.
----------	--------------------------------

6.4 Examples

6.4.1 Successful Request

Request

```
POST /dns/test.com/records
Authorization: Basic {base64_encoded(username:password)}
```

```
{
  "name": "alt.test.com",
  "type": "CNAME",
  "cname": "altdomain.com"
}
```

Response

201 Created
Location: <https://api.hostway.com/dns/test.com/records/447>

6.4.2 Unauthorized Request

Request

POST /dns/test.com/records

```
{
  "name": "alt.test.com",
  "type": "CNAME",
  "cname": "altdomain.com"
}
```

Response

401 Unauthorized

6.4.3 Attempt to create duplicate record

Request

POST /dns/test.com/records
Authorization: Basic {base64_encoded(username:password)}

```
{
  "name": "alt.test.com",
  "type": "CNAME",
  "cname": "altdomain.com"
}
```

Response

409 Conflict

```
{
  "conflict": {"message": "Data conflict", "code": 409, "details": ""}
}
```

6.5 See also

- [DNS API](#)
- [Common Features](#)
- [RFC 1035](#)
- [RFC 2782](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

7 Dns-zone-records-recordId:PUT

8 PUT /dns/{zone}/records/{recordId}

Update an existing zone record.

8.1 Contents

- 1 PUT /dns/{zone}/records/{recordId}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 A record
 - 1.1.4.2 AAAA record
 - 1.1.4.3 NS record
 - 1.1.4.4 TXT record
 - 1.1.4.5 SRV record
 - 1.1.4.6 MX record
 - 1.1.4.7 CNAME record
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - ◆ 1.3 Examples
 - ◇ 1.3.1 Unauthorized Request
 - ◇ 1.3.2 Successful Update
 - ◆ 1.4 See also

8.2 Request

PUT /dns/{zone}/records/{recordId}

8.2.1 Request Parameters

zone - *string*
The DNS zone of the record.

recordId - *string*
The unique Id of the record.

8.2.2 URI Parameters

None.

8.2.3 Request Headers

Content-Type
application/json

Authorization
Base64-encoded username & password string

8.2.4 Request Body

Added support for wildcard domain names.
Ex:

```
*
*.somedomain.com
The wildcard record name is stored in the database as it is supplied.
If it's missing the zone origin(i.e. '') the API will return it with the zone
appended, i.e. '*.domain.com'.
```

Added support for @ records, which represent the origin itself.
Ex:

```
@
The API will return the record with the zone origin appended to the name, i.e. '@.domain.com'
```

8.2.4.1 A record

```
{
  "name": {domain},
  "address": {ip_address},
  "ttl": 3600
}
```

8.2.4.2 AAAA record

```
{
  "name": {domain},
  "address": {ipv6_address},
  "ttl": 3600
}
```

8.2.4.3 NS record

```
{
  "name": {domain},
  "nsdname": {ns_domain},
  "ttl": 3600
}
```

8.2.4.4 TXT record

```
{
  "name": {domain},
  "data": {data},
  "ttl": 3600
}
```

8.2.4.5 SRV record

```
{
  "name": {domain},
  "priority": {priority},
  "weight": {weight},
  "port": {port},
  "target": {target},
  "ttl": 3600
}
```

8.2.4.6 MX record

```
{
  "name": {domain},
  "preference": {preference},
  "exchange": {mx_domain},
  "ttl": 3600
}
```

8.2.4.7 CNAME record

```
{
  "name": {name},
  "cname": {domain},
  "ttl": 3600
}
```

8.3 Response

8.3.1 Status Code

200 OK
 Success

400 Bad Request
 TTL is below allowed minimum

401 Unauthorized

403 Forbidden

409 Conflict

8.3.2 Response Body

None.

8.4 Examples

8.4.1 Unauthorized Request

Request

```
PUT /dns/test.com/records/1234
{
  "name": "test",
  "cname": "otherdomain.com."
}
```

Response

```
401 Unauthorized
```

8.4.2 Successful Update

Request

```
PUT /dns/test.com/records/1234
Authorization: Basic {base64_encoded(username:password)}
{
  "name": "test1",
  "cname": "otherdomain1.com."
}
```

Response

```
200 OK
{
  "name": "test1",
  "type": "CNAME",
  "cname": "otherdomain1.com."
}
```

8.5 See also

- [DNS API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

9 Dns-zone-records-recordId:DELETE

10 DELETE /dns/{zone}/records/{recordId}

Delete zone record.

10.1 Contents

- 1 DELETE /dns/{zone}/records/{recordId}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◆ 1.2 Request Headers
 - ◇ 1.2.1 Request Body
 - ◆ 1.3 Response
 - ◇ 1.3.1 Status Code
 - ◆ 1.4 Examples
 - ◆ 1.5 See also

10.2 Request

DELETE /dns/{zone}/records/{recordId}

10.2.1 Request Parameters

zone - *string*
The DNS zone name.

recordId - *string*
The unique record Id.

10.2.2 URI Parameters

None.

10.3 Request Headers

Authorization
Base64-encoded username & password string

10.3.1 Request Body

None.

10.4 Response

10.4.1 Status Code

204 No Content
Success

401 Unauthorized

403 Forbidden

404 Not Found
The specified zone or record Id don't exist.

10.5 Examples

Request

```
DELETE /dns/test.com/records/447
Authorization: Basic {base64_encoded(username:password)}
```

Response:

204 No Content

10.6 See also

- [DNS API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

11 Dns-heartbeat:GET

12 GET /dns/heartbeat

12.1 Contents

- 1 GET /dns/heartbeat
 - ◆ 1.1 Request
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Codes
 - ◆ 1.3 See also

12.2 Request

GET /dns/heartbeat

12.3 Response

12.3.1 Status Codes

200 OK
Success

500 Internal Server Error
Internal error occurred.

12.4 See also

- [DNS API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

13 Dns-healthcheck:GET

14 GET /dns/healthcheck

14.1 Contents

- 1 GET /dns/healthcheck
 - ◆ 1.1 Request
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Codes
 - ◆ 1.3 See also

14.2 Request

GET /dns/healthcheck

14.3 Response

14.3.1 Status Codes

200 OK
Success

400 Bad Request
If any of the database connections is down.

500 Internal Server Error
If something is wrong with the DNS API.

14.4 See also

- [DNS API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

15 Common Features

All Hostway APIs support common features for authentication and authorization, pagination, references to related resources, etc.

- [Authentication](#)
- [Hypermedia](#)
- [Pagination](#)
- [Server Response Codes](#)
- [Filtering](#)

[download this selection of articles as a PDF book](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)