MYRA

# Web API

## Documentation

v1.10.3

May 26, 2020

## Contents

**1 Overview** **6**

**2 Security** **7**
2.1 Reference implementations and examples . . . 7
2.2 Generate a signing string . . . 7
2.3 Generate a signing key . . . 8
2.4 Generate a signature . . . 8

**3 Object descriptions** **10**
3.1 Objects returned . . . 10
3.1.1 QueryVO . . . 10
3.1.2 ResultVO . . . 10
3.1.3 ViolationVO . . . 11

**4 Domain** **12**
4.1 Overview . . . 12
4.2 Usage . . . 12
4.2.1 Listing . . . 13
4.2.2 Create . . . 14
4.2.3 Update . . . 15
4.2.4 Delete . . . 16
4.3 Full version of a Domain object . . . 17

**5 DNS** **19**
5.1 Overview . . . 19
5.2 Usage . . . 19
5.2.1 Listing . . . 20
5.2.2 Create . . . 22
5.2.3 Update . . . 23
5.2.4 Delete . . . 24
5.3 Full version of a DnsRecord object . . . 25

**6 Load balance** **27**
6.1 Overview . . . 27
6.2 Usage . . . 27
6.2.1 Listing . . . 28
6.2.2 Create . . . 29
6.2.3 Update . . . 31
6.2.4 Delete . . . 32

**7 Subdomain settings** **33**
7.1 Overview . . . . . 33
7.2 Usage . . . . . 33
7.2.1 Listing . . . . . 34
7.2.2 Update . . . . . 35
7.3 Full version of a Subdomain settings object . . . . . 36

**8 Cache** **45**
8.1 Overview . . . . . 45
8.2 Usage . . . . . 45
8.2.1 Listing . . . . . 45
8.2.2 Create . . . . . 46
8.2.3 Update . . . . . 46
8.2.4 Delete . . . . . 47
8.3 Full version of a Cache object . . . . . 48

**9 CacheClear** **50**
9.1 Overview . . . . . 50
9.2 Usage . . . . . 50
9.2.1 Clear . . . . . 50
9.2.2 Clear of a whole domain . . . . . 51
9.2.3 Pattern matching . . . . . 52
9.3 Full version of a CacheClear object . . . . . 53

**10 Redirects** **54**
10.1 Overview . . . . . 54
10.2 Usage . . . . . 54
10.2.1 Listing . . . . . 55
10.2.2 Create . . . . . 56
10.2.3 Update . . . . . 58
10.2.4 Delete . . . . . 59
10.3 Full version of a Redirect object . . . . . 60

**11 IP black/whitelist** **62**
11.1 Overview . . . . . 62
11.2 Usage . . . . . 62
11.2.1 Listing . . . . . 62
11.2.2 Create . . . . . 63
11.2.3 Update . . . . . 63
11.2.4 Delete . . . . . 64
11.3 Full version of a IpFilter object . . . . . 65

**12 Maintenance** **66**
12.1 Overview . . . . . 66
12.2 Usage . . . . . 66
12.2.1 Listing . . . . . 66
12.2.2 Create . . . . . 67
12.2.3 Update . . . . . 69
12.2.4 Delete . . . . . 69
12.3 Full version of a Maintenance object . . . . . 71

**13 SSL certificates** **73**
13.1 Overview . . . . . 73
13.2 Usage . . . . . 73
13.2.1 Listing . . . . . 74
13.2.2 Create . . . . . 76
13.2.3 Update . . . . . 77
13.2.4 Delete . . . . . 78
13.3 Full version of a SslCertVO object . . . . . 79

**14 Statistics** **83**
14.1 Overview . . . . . 83
14.2 Usage . . . . . 83
14.2.1 Query . . . . . 84
14.3 StatisticVO . . . . . 86
14.3.1 StatisticQueryVO . . . . . 87
14.3.2 Data sources . . . . . 89
14.3.3 Result . . . . . 95

**15 Permissions** **96**
15.1 Overview . . . . . 96
15.2 Usage . . . . . 96
15.2.1 Clone . . . . . 97

**16 IP address ranges of Myra** **98**
16.1 Overview . . . . . 98
16.2 Usage . . . . . 98
16.2.1 Listing . . . . . 99
16.2.2 Create . . . . . 99
16.2.3 Update . . . . . 99
16.2.4 Delete . . . . . 99
16.3 Full version of a IP address ranges object . . . . . 100

**17 Development** **102**
17.1 MYRA_NOCACHE . . . . . 102

**18 Cache (Deprecated) 103**
18.1 Overview . . . 103
18.2 Security . . . 103
18.3 Usage . . . 103
18.4 Command: clear . . . 104

# 1 Overview

The web API of Myra is based on REST using the JSON format for the messages. We are supporting programmatic access to almost every operation you can do on our web backend. You can manage your cache settings accordingly to your deployment cycle or change your DNS settings, to name a few.

# 2 Security

For all REST calls you use your API key[1] to authenticate your user. The whole communication is encrypted via TLS/SSL. To prevent your configuration from man-in-the-middle attacks or phishing attempts it is necessary to add an additional security layer.

For the additional security layer we have added a hashed signature containing a timestamp, which prevents attackers from repeating a call without knowing the secret for a certain period of time.

```
Authorization: MYRA {apiKey}:{signature}
Date: {date}
```

Listing 1: Authorization header

To calculate a signature you first have to concatenate specific request elements to build a string. After building the string you need to generate a signing key and sign your string with your key.

## 2.1 Reference implementations and examples

You can find reference implementations and examples of generating the hashed signature in PHP, Python, and other languages at our GitHub page:

**https://github.com/Myra-Security-GmbH**

## 2.2 Generate a signing string

The specific request elements you need are

- MD5 hash of your request body
- your request method
- your URI,
- the content type, set in your request header
- the date, set in your request header

You concatenate all those elements with a '#' character. In case of a GET request, which cannot contain a request body, please use a MD5 hash of an empty string (d41d8cd98f00b204e9800998ecf8427e).

Please note that the following example contains spaces for readability. Your signing string should not contain any spaces around the '#' character.

[1] You will receive your API key from the Myra support team

```
$signingString := d41d8cd98f00b204e9800998ecf8427e \
    #GET \
    #/en/rapi/cacheSetting/www.example.de \
    #application/json \
    #2014-04-26CET13:04:00+0100
```

Listing 2: Example string to sign for a GET request

## 2.3 Generate a signing key

A signing key is generated in two steps. First step is to create a date key as hex hmac representation, which is created from the current date and your secret[2].

```
$dateKey := hmac_sha256(
    '2014-04-26CET13:04:00+0100',
    'MYRA' + $secret
);

// $secret is: 6b3a71954faf11e4b898001517fa8424

// Calculated $dateKey is
// 1c2a270750de0cc1b8c3522494abd9a04e0b7801be6ece02755fa7bc9f8f5467
```

Listing 3: DateKey creation

Please note: You have to use the same date string as used in the date of your request header where the time is not allowed to differ more than 10 seconds into the future or 30 minutes into the past.

Second step is to use your date key to sign the string 'myra-api-request', which is also created as hex hmac representation.

```
$signingKey := hmac_sha256(
    'myra-api-request',
    $dateKey
);

// Calculated $signingKey is
// 3744ae9c3d3f87c3ce90a99957f9f054266ef9386fc909e2a24a7031c7571ffd
```

Listing 4: SigningKey creation

## 2.4 Generate a signature

To generate a signature you first have to create a signing string and a signing key. After you generated both keys you have to use the signing key to encrypt the signing string via hmac sha512 and return a base64 representation of the generated hash.

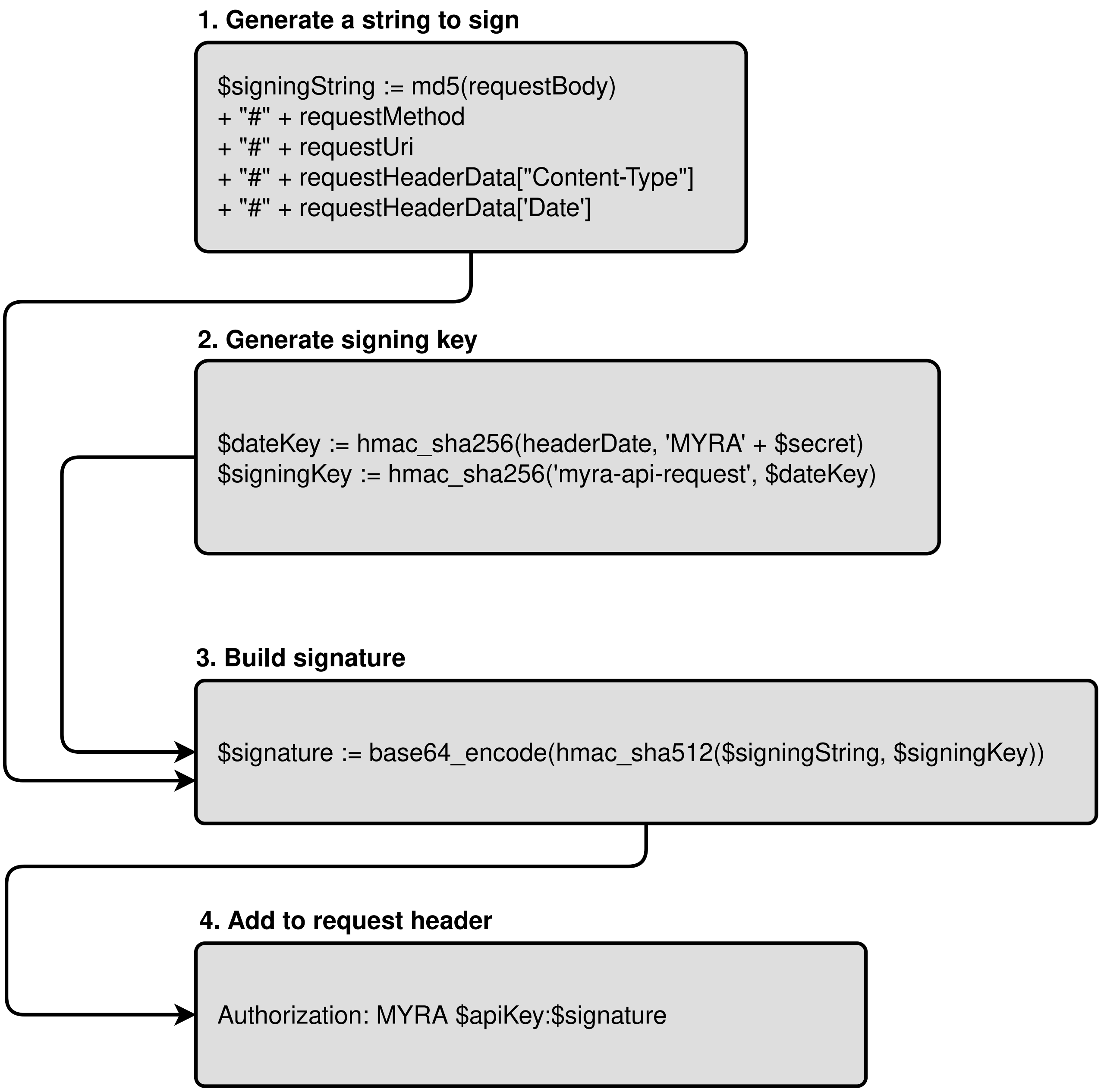
[2] You will receive your secret from the Myra support team

Please note: the hmac sha512 result is used in binary representation.

```
$signature := base64(
    hmac_sha512($signingString, $signingKey)
);

// Calculated $signature is:
// 7OXCjTTssU9DD/mkbhyp5Syup0ufUm1YOWUj66hsxmTctVordMIVLS30pi7CSp1hC7EcZ2q1hvpXJMMNkvAncw==
```

Listing 5: Generate a signature



**1. Generate a string to sign**

```
$signingString := md5(requestBody)
+ "#" + requestMethod
+ "#" + requestUri
+ "#" + requestHeaderData["Content-Type"]
+ "#" + requestHeaderData['Date']
```

**2. Generate signing key**

```
$dateKey := hmac_sha256(headerDate, 'MYRA' + $secret)
$signingKey := hmac_sha256('myra-api-request', $dateKey)
```

**3. Build signature**

```
$signature := base64_encode(hmac_sha512($signingString, $signingKey))
```

**4. Add to request header**

```
Authorization: MYRA $apiKey:$signature
```

# 3 Object descriptions

## 3.1 Objects returned

All API calls will return an HTTP response code of 200 with one of two different object types or in the error case an appropriate HTTP error code.

A GET call will always return an object of the type QueryVO (Query Value Object) and all other calls return an object from the type ResultVO (Result Value Object).

Each object has similar structured objects, which contain data from the current call.

### 3.1.1 QueryVO

An object of the type QueryVO is always returned when receiving a list of objects from an API call.

```
{
    "error"    : false,
    "list"     : [],
    "page"     : 1,
    "count"    : 1,
    "pageSize" : 50
}
```

Listing 6: QueryVO

Table 1: Attributes for QueryVO

| Attribute | Type | Description |
|---|---|---|
| error | Boolean | Shows if the request had an error |
| list | Array | A list of matches for the given page |
| page | Integer | Number of the current page |
| count | Integer | Number of hits |
| pageSize | Integer | Number of matches maximal shown at a page |

### 3.1.2 ResultVO

An object of the type ResultVO is always returned for API calls doing inserts, updates, or deletes.

```
{
    "error"         : false,
    "violationList" : [],
    "targetObject"  : []
}
```

Listing 7: ResultVO

Table 2: Attributes for ResultVO

| Attribute | Type | Description |
|---|---|---|
| error | Boolean | Shows if the request has an error |
| violationList | Array | A list of ViolationVOs to show validation errors |
| targetObject | Array | Returns a list of processed objects |

### 3.1.3 ViolationVO

An object of the type ViolationVO (Violation Value Object) shows the exact error for a specific attribute or a generic error if no path is set.

```
{
    "path"    : "",
    "message" : ""
}
```

Listing 8: ViolationVO

Table 3: Attributes for ViolationVO

| Attribute | Type | Description |
|---|---|---|
| path | String | The attribute name which was not correctly validated |
| message | String | An error message, which describes what was going wrong |

# 4 Domain

## 4.1 Overview

You can manage your domains directly with Myra.

## 4.2 Usage

All domain requests use URL specific parameters; a list of these parameters can be found in the table below.

Table 4: URL parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| page | Number | Yes | Number of the page you want to retrieve (1-based). |

### 4.2.1 Listing

The listing operation returns a list of domains. The list contains domains for the account you are accessing the API with, and also all foreign domains you are allowed to manage.
The listing returns also additional status information about maintenance, pause, or ownership.

The REST request must have the following format:

```
GET /{language}/rapi/domains/{page} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}
```

Listing 9: Example request

A successful call will return a response with a HTTP status code of 200 and the following body:

```
{
"error"    : false,
"list"     : [
    {
        "id"            : 1,
        "modified"      : "2013-12-13T11:30:00+0100",
        "created"       : "2013-12-09T11:30:00+0100",
        "name"          : "example.com",
        "autoUpdate"    : true,
        "paused"        : true,
        "pausedUntil"   : "2014-01-01T11:00:00+0100",
        "owned"         : true,
        "maintenance"   : false
    }
],
"page"     : 1,
"count"    : 1,
"pageSize" : 50
}
```

Listing 10: Example response

### 4.2.2 Create

To create a new domain it is necessary to send a Domain object without the attributes "id" and "modified". Both attributes will be generated by the server and returned after a successful insert is done.

```
PUT /{language}/rapi/domains HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "name"          : "example.com",
    "deleted"       : false,
    "autoUpdate"    : false,
    "autoDns"       : false
}
```

Listing 11: Example

A successful insert will return a ResultVO with the new created Domain object.

```
{
"error": false,
"violationList": [],
"targetObject": [
    {
        "id"            : 1,
        "modified"      : "2013-12-13T11:30:00+0100",
        "created"       : "2013-12-09T11:30:00+0100",
        "name"          : "example.com",
        "autoUpdate"    : true,
        "paused"        : true,
        "pausedUntil"   : "2014-01-01T11:00:00+0100",
        "owned"         : true,
        "maintenance"   : false
    }
  ]
}
```

Listing 12: Result

After a successful insert your configuration will be queued and deployed to our system.

### 4.2.3 Update

Updating a domain is very similar to creating a new one. The main difference is that an update will need the generated "id" and "modified" attributes to identify the object you are trying to update.

Updating a domain allows you **only to change** the **autoUpdate** flag. All other values are ignored. If a domain **contains a typo** you have to **remove and recreate** the domain with the correct name.

```
POST /{language}/rapi/domains HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "id"            : 3,
    "modified"      : "2013-12-09T11:30:00+0100",
    "autoUpdate"    : true
}
```

Listing 13: Example

A successful update will return the same type of object as the create operation. After an update your configuration will be queued and deployed to our system.

### 4.2.4 Delete

For deleting a domain it is only necessary to send the "id" and "modified" attributes as body content.

Removing a domains also removes all configurations on Myra!
This could lead to an outage of your online presence. Please make sure that you are prepared for it. If unsure contact Myra support.

```
DELETE /{language}/rapi/domains HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "id"            : 3,
    "modified"      : "2013-12-09T11:30:00+0100"
}
```

Listing 14: Example

After a delete operation your configuration will be queued and removed from our system.

## 4.3 Full version of a Domain object

```
{
    "id"            : 1,
    "modified"      : "2014-03-12T18:21:49+0100",
    "created"       : "2014-03-06T18:21:49+0100",
    "name"          : "example.com",
    "autoUpdate"    : false,
    "autoDns"       : false,
    "paused"        : true,
    "pausedUntil"   : "2014-04-01T00:00:00+0100",
    "owned"         : true,
    "maintenance"   : false
}
```

Listing 15: Full version of a Domain object

Table 5: Attributes for Domain

| Attribute | Type | Required | Short description |
|---|---|---|---|
| id | Number | UD | Id of the object |
| modified | Date | UD | Date of last modification |
| created | Date | - | Date of creation |
| name | String | C | Domain name |
| autoUpdate | Bool | CU | Auto update flag for the domain |
| autoDns | Bool | C | Auto DNS flag for the domain |
| paused | Bool | - | Shows if Myra is paused for this domain |
| pausedUntil | Date | - | Date until Myra will be automatically reactivated |
| owned | Bool | - | Shows if your current account is the owner for this domain |
| maintenance | Bool | - | Shows if this domain is currently on maintenance |

**C** = Create, **U** = Update, **D** = Delete

**id**

Id is an unique identifier for an object. This value is always a number type and cannot be set while inserting a new object. To update or delete a Domain it is necessary to add this attribute to your object.

**modified**

Identifies the version of the object. To ensure that you are updating the most recent version and not overwriting other changes, you always have to add modified for updates and deletes. This value is always a date type with an ISO8601 format.

**created**

Created is a date type attribute with an ISO8601 format. It will be created by the server after creating a new Domain object. This value is informational so it is not necessary to add this attribute to any API call.

**name**
Identifies the domain by its name. The value cannot be changed after creation. To change a typo you need to remove and recreate the domain.

**autoUpdate**
Shows if the current domain has autoUpdate activated. If autoUpdate is deactivated changes on your configuration are not deployed until you reactivate autoUpdate. This is primary used to change a lot of settings at once to prevent Myra to deploy a half done configuration. In some cases Myra support also deactivates this option to prevent Myra system from removing special configuration settings. Please note that turning autoUpdate off is not correlated to database transactions. This means that any changes are saved but not deployed.

**autoDns**
If autoDns flag is set while creating a new domain Myra tries to get a list of subDomains for this domain. Depending on your DNS provider configuration this may fail or return a incomplete list. For best results Myra recomments to use the subDomain API to create DNS records.

**paused**
Shows if the domain is currently in pause mode.

**pauseUntil**
Shows the date when Myra protection will be reactivated automatically.

**owned**
Identifies if the account is the owner of this domain.

**maintenance**
True if this domain is currently in maintenance mode. Maintenance pages can be configured using the Maintenance API.

# 5 DNS

## 5.1 Overview

The DNS settings lets you configure your DNS related settings like changing your origin IP or setting up a new DNS record.

## 5.2 Usage

All DNS requests use URL specific parameters; a list of these parameters can be found in the table below.

Table 6: URL parameters

| **Parameter** | **Type** | **Required** | **Description** |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain you are trying to list DNS entries for. |
| page | Number | Yes | Number of the page you want to retrieve (1-based). |

### 5.2.1 Listing

To list all DNS objects belonging to a given domain you have to use the following REST request:

```
GET /{language}/rapi/dnsRecords/{domain}/{page} HTTP/1.1
Host: api.myracloud.com
Date: 2014-06-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}
```

Listing 16: Example request for listing DNS objects

A successful call will return a response with a HTTP status code of 200 and the following body:

```
{
"error"    : false,
"list"     : [
    {
        "objectType"      : "DnsRecordVO",
        "id"              : 1,
        "modified"        : "2018-02-02T10:27:25+0100",
        "created"         : "2018-02-02T10:26:08+0100",
        "name"            : "subdomain.example.de",
        "value"           : "127.0.0.1",
        "priority"        : 0,
        "ttl"             : 300,
        "recordType"      : "A",
        "active"          : true,
        "enabled"         : true,
        "paused"          : false,
        "upstreamOptions" : {
            "backup": false,
            "down"  : false
        },
        "alternativeCname": "subdomain-example-de.ax4z.com.",
        "caaFlags"        : 0
    }
],
"page"     : 1,
"count"    : 1,
"pageSize" : 50
}
```

Listing 17: Example response listing DNS records

### Filtering

The DNS listing end-point allows to filter the output by using the following URI parameters:

Table 7: URL parameters

| Parameter | Type | Description | Default |
|---|---|---|---|
| search | String | Filters the DNS records with the specified string in the name or in the value field | null |
| recordTypes | String | Filters the DNS by record type. It is possible to specify more than one record type as a comma separated list | null |
| activeOnly | Boolean | Wen set to "true" only active DNS records are returned | false |
| loadbalancer | Boolean | When set to true only returns the DNS record that are being load balanced | false |

### 5.2.2 Create

The property "sslCertTemplate" is optional. If you do not want to use a SSL certificate from an existing subdomain, then just remove the whole property from the request.

To create a new DNS entry you have to send a DnsRecord object without the attributes "id", "created", and "modified". All attributes are generated by the server and returned after that.

```
PUT /{language}/rapi/dnsRecords/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "name"            : "subdomain.example.de",
    "value"           : "127.0.0.1",
    "ttl"             : 300,
    "recordType"      : "A",
    "active"          : true,
    "sslCertTemplate" : "ssldomain.example.de"
}
```

Listing 18: Example response creating a DNS record

A successful insert will return a ResultVO with the newly created DnsRecord and all records which are matching the same name.

```
{
"error"         : false,
"violationList" : [],
"targetObject"  : [
    {
    "id"           : 1,
    "modified"     : "2014-03-06T18:21:49+0100",
    "created"      : "2014-03-06T18:21:49+0100",
    "name"         : "subdomain.example.de",
    "value"        : "127.0.0.1",
    "ttl"          : 300,
    "recordType"   : "A",
    "active"       : true
    }
  ]
}
```

Listing 19: Example response creating a DNS record

After a successful insert your configuration will be queued and deployed to our system.

### 5.2.3 Update

Updating a DNS record is very similar to creating a new one. You will need to provide the generated "id" and "modified" attributes to identify the version of object you are trying to update.

```
POST /{language}/rapi/dnsRecords/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "id"            : 3,
    "name"          : "subdomain.example.de",
    "value"         : "127.0.0.2",
    "ttl"           : 300,
    "recordType"    : "A",
    "active"        : true,
    "modified"      : "2013-12-09T11:30:00+0100"
}
```

Listing 20: Example response updating a DNS record

To see a valid response, take a look at the response from the create section. After an update your configuration will be queued and deployed to our system.

### 5.2.4 Delete

To delete a DNS record it is only necessary to send "id" and "modified" as body content to identify the version of the record you are trying to delete.

```
DELETE /{language}/rapi/dnsRecords/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "id"             : 3,
    "modified"       : "2013-12-09T11:30:00+0100"
}
```

Listing 21: Example response deleting a DNS record

After a delete operation your configuration will be queued and deployed to our system.

## 5.3 Full version of a DnsRecord object

```
{
    "id"               : 1,
    "modified"         : "2013-12-11T11:35:00+0100",
    "created"          : "2013-12-09T11:35:00+0100",
    "name"             : "subdomain.example.de",
    "value"            : "127.0.0.1",
    "ttl"              : 300,
    "recordType"       : "A",
    "alterativeCname" : "subdomain-example-de.ax4z.com",
    "active"           : true
}
```

Listing 22: Full version of a DnsRecord object

Table 8: Attributes for DNS records

| Attribute | Type | Required | Short description |
|---|---|---|---|
| id | Number | UD | Id of the object |
| modified | Date | UD | Date of last modification |
| created | Date | - | Date of creation |
| name | String | CU | Subdomain name of a DNS record |
| ttl | String | CU | Time to live |
| recordType | String | CU | A recordType to identify the type of a record. Valid types are: A, AAAA, MX, CNAME, TXT, NS, SRV and PTR |
| sslCertTemplate | String | C | A FQDN in the same domain |
| alternativeCname | String | - | The alternative CNAME that points to the record. |
| active | Boolean | CU | Define wether this subdomain should be protected by Myra or not |
| value | String | CU | Depends on the record type. Typically an IPv4/6 address or a domain entry |
| priority | Number | CU | Priority of MX records |
| port | Number | CU | Port for SRV records |

**C** = Create, **U** = Update, **D** = Delete

**id**

Id is an unique identifier for an object. This value is always a number type and cannot be set while inserting a new object. To update or delete a DnsRecord it is necessary to add this attribute to your object.

**modified**

Identifies the version of the object. To ensure that you are updating the most recent version and not overwriting other changes, you always have to add modified for updates and deletions. This value is always a date type with an ISO8601 format.

**created**

Created is a date type attribute with an ISO8601 format. It will be created by the server after creating a new DnsRecord object. This value is only informational so it is not necessary to add this an attribute to any API call.

**ttl**

Time to live limits the time a DNS record entry can be cached by DNS servers. This typically sets the timeframe how long changes need to be propagated to the DNS servers. This is a numeric value and is given in seconds.

**recordType**

This value sets the type of the DNS record. All available record types are A, AAAA, MX, CNAME, TXT, NS, SRV and PTR. PTR records are only useable on .arpa domains.

**sslCertTemplate**

This value allows to bind a new DNS record to the same SSL certificate as the given FQDN. The new record will also receive the same IP address as the sslCertTemplate record. To prevent invalid usage it is only allowed to use a FQDN in the same domain as the new record. The usage of this property is optional. If you do not want to use an existing SSL certificate, then remove the whole property.

**alternativeCname**

The alternative CNAME that points to the current record. This attribute is generated by the API and cannot be set.

**active**

This information shows if this record is protected by Myra.

**value**

Depending of the recordType this property is used to set an IP address or a domain name.

**priority**

Priority for MX and SRV records.

**port**

Port information for SRV records.

# 6 Load balance

## 6.1 Overview

The load balance setting lets you configure the way different IP addresses pointing to the same DNS record name behave.

## 6.2 Usage

It is possible to retrieve or modify the load balance settings by using the DNS record endpoint.

All the load balance requests use URL specific parameters; a list of these parameters can be found in the table below.

Table 9: URL parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain you are trying to list DNS entries for. |
| page | Number | Yes | Number of the page you want to retrieve (1-based). |

### 6.2.1 Listing

To list all load balance objects belonging to a given domain you have to use the DNS record endpoint by specifying the *loadbalancer* filter:

```
GET /{language}/rapi/dnsRecords/{domain}/{page}?loadbalancer=true&recordTypes=A,AAAA,CNAME HTTP/1.1
Host: api.myracloud.com
Date: 2014-06-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}
```

Listing 23: Example request for listing DNS objects

A successful call will return a response with a HTTP status code of 200 and the following body:

```
{
"error"    : false,
"list"     : [
    ...
    {
        "objectType"      : "DnsRecordVO",
        "id"              : 2,
        "modified"        : "2018-02-02T10:27:25+0100",
        "created"         : "2018-02-02T10:26:08+0100",
        "name"            : "subdomain.example.de",
        "value"           : "1.1.1.2",
        "priority"        : 0,
        "ttl"             : 300,
        "recordType"      : "A",
        "active"          : true,
        "enabled"         : true,
        "paused"          : false,
        "upstreamOptions" : {
            backup     : true
            down       : false
            failTimeout: 3
            maxFails   : 2
            weight     : 2
        },
        "alternativeCname": "subdomain-example-de.ax4z.com.",
        "caaFlags"        : 0
    }
    ...
],
"page"     : 1,
"count"    : 3,
"pageSize" : 50
}
```

Listing 24: Example response listing DNS records

### 6.2.2 Create

To create a new loadbalance entry you have to create a DnsRecord object without the attributes "id", "created", and "modified" but with the same *name* field as the record is being balanced. All attributes are generated by the server and returned after that.
The DNS record end-point allows to specify load balancing settings within the ***upstreamOptions*** field. The *upstreamOptions* field is an object containing the following fields:

Table 10: upstreamOptions

| **Field** | **Type** | **Description** |
|---|---|---|
| weight | Number | Defines the priority for the origin server, the higher the number, the higher the priority of the server |
| maxFails | Number | Sets the number of unsuccessful attempts to communicate with the origin that should happen in the duration set by the failTimeout parameter to consider the origin unavailable for a duration also set by the failTimeout parameter |
| failTimeout | Number | The time during which the specified number of unsuccessful attempts to communicate with the origin should happen to consider the origin unavailable and the period of time the origin will be considered unavailable |
| backup | Boolean | Marks the origin server as backup. The servers marked as backup will receive the requests when the primary servers are unavailable |
| down | Boolean | Marks the origin server as permanently unavailable |

```
PUT /{language}/rapi/dnsRecords/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "name"             : "subdomain.example.de",
    "value"            : "1.1.1.2",
    "ttl"              : 300,
    "recordType"       : "A",
    "active"           : true,
    "upstreamOptions": {
        backup      : true
        down        : false
        failTimeout: 3
        maxFails    : 2
        weight      : 2
    }
}
```

Listing 25: Example response creating a DNS record

A successful insert will return a ResultVO with the newly created DnsRecord and all records which are matching the same name.

```
{
"error"         : false,
"violationList" : [],
"targetObject"  : [
    {
        "id"           : 2,
        "modified"     : "2018-02-02T10:27:25+0100",
        "created"      : "2018-02-02T10:26:08+0100",
        "name"         : "subdomain.example.de",
        "value"        : "1.1.1.2",
        "ttl"          : 300,
        "recordType"   : "A",
        "active"       : true,
        "upstreamOptions": {
            backup     : true
            down       : false
            failTimeout: 3
            maxFails   : 2
            weight     : 2
        }
    }
  ]
}
```

Listing 26: Example response creating a DNS record

After a successful insert your configuration will be queued and deployed to our system.

### 6.2.3 Update

Updating a load balanced DNS record is very similar to creating a new one. You will need to provide the generated "id" and "modified" attributes to identify the version of object you are trying to update.

```
POST /{language}/rapi/dnsRecords/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "id"             : 2,
    "modified"       : "2018-02-02T10:27:25+0100",
    "name"           : "subdomain.example.de",
    "value"          : "1.1.1.2",
    "ttl"            : 300,
    "recordType"     : "A",
    "active"         : true,
    "upstreamOptions": {
        backup     : true
        down       : false
        failTimeout: 3
        maxFails   : 2
        weight     : 2
    }
}
```

Listing 27: Example response updating a DNS record

To see a valid response, take a look at the response from the create section. After an update your configuration will be queued and deployed to our system.

### 6.2.4 Delete

To delete a DNS record it is only necessary to send "id" and "modified" as body content to identify the version of the record you are trying to delete.
See 5.2.4 for an example.

# 7 Subdomain settings

## 7.1 Overview

The (sub)domain settings API lets you adjust the Myra configuration for your domain or specific subdomains.

## 7.2 Usage

All (sub)domain settings requests use URL specific parameters; a list of these parameters can be found in the table below.

Table 11: URL parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain you are trying to list (sub)domain settings for. |

### 7.2.1 Listing

To list all (sub)domain settings objects belonging to a given domain you have to use the following REST request:

```
GET /{language}/rapi/subdomainSetting/{domain}   HTTP/1.1
Host: api.myracloud.com
Date: 2014-06-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}
```

Listing 28: Example request for listing (sub)domain settings objects

A successful call will return a response with a HTTP status code of 200 and the following body:

```
{
    domain: {
        "cdn"                   : true,
        "access_log"            : false
    },
    parent: {
        "access_log"            : false,
        "antibot"               : no,
        "myra_ssl_header"       : false,
        "request_limit_level"   : 6000
    }

}
```

Listing 29: Example response listing (sub)domain settings object

Additionally you can request a already merged representation of the (sub)domain settings by sending the query parameter "flat". The result returned will be a flat object with all (sub)domain specific and inherited settings merged correctly.

```
GET /{language}/rapi/subdomainSetting/{domain}?flat   HTTP/1.1
Host: api.myracloud.com
Date: 2014-06-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}
```

Listing 30: Example request for listing (sub)domain settings objects

```
{
     "cdn"                   : true,
     "access_log"            : false,
     "antibot"               : no,
     "myra_ssl_header"       : false,
     "request_limit_level"   : 6000

}
```

Listing 31: Example response listing (sub)domain settings object

### 7.2.2 Update

To create new (sub)domain settings you have to send a (sub)domain settings object. All attributes are generated by the server and returned after that.

```
POST /{language}/rapi/subdomainSetting/{domain}   HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "access_log"          : false,
    "antibot"             : "auto",
    "cdn"                 : true,
    "image_optimization"  : false,
    "request_limit_block" : false,
    "rewrite"             : "no",
    "security_level"      : "high",
    "spdy"                : true
}
```

Listing 32: Example request creating a (sub)domain settings

A successful insert will return a ResultVO with the newly created (sub)domain settings and all settings which are matching the same name.

```
{
"error": false,
"violationList": [],
"targetObject": [
    {
        "access_log"          : false,
        "antibot"             : "auto",
        "cdn"                 : true,
        "image_optimization"  : false,
        "request_limit_block" : false,
        "rewrite"             : "no",
        "security_level"      : "high",
        "spdy"                : true
    }
  ]
}
```

Listing 33: Example response creating (sub)domain settings

After a successful insert your configuration will be queued and deployed to our system.

## 7.3 Full version of a Subdomain settings object

The (sub)domain settings object contains two properties. The settings for the given (sub)domain are listed in "domain". All inherited settings are listed in "parent". Settings in "domain" always overwrite the inherited setting with the same name.

```
{
    "domain": {},
    "parent": {
        "access_log"                 : false,
        "antibot"                    : "no",
        "antibot_threshold"          : "20",
        "balancing_method"           : "round_robin",
        "block_not_whitelisted"      : false,
        "cache_enabled"              : false,
        "cache_revalidate"           : true,
        "cdn"                        : false,
        "diffie_hellman_exchange"    : 1024,
        "enforce_cache_ttl"          : false,
        "hsts"                       : false,
        "hsts_include_subdomains"    : false,
        "hsts_max_age"               : 31536000,
        "hsts_preload"               : false,
        "image_optimization"         : true,
        "ipv6_active"                : true,
        "myra_ssl_header"            : false,
        "next_upstream"              : [
            "error",
            "timeout",
            "invalid_header"
        ],
        "only_https"                 : false,
        "origin_connection_header"   : "none",
        "proxy_cache_key"            : "$scheme:$host:$uri:$query_string",
        "proxy_cache_stale"          : [
            "updating"
        ],
        "proxy_connect_timeout"      : 20,
        "proxy_read_timeout"         : 20,
        "request_limit_block"        : false,
        "request_limit_level"        : 6000,
        "request_limit_report"       : false,
        "rewrite"                    : "no",
        "rewrite_regex"              : "",
        "security_level"             : "standard",
        "source_protocol"            : "same",
        "spdy"                       : false,
        "streaming_regex"            : ""
    }
}
```

Listing 34: Full version of a (sub)domain settings object

Table 12: Attributes for (sub)domain settings

| **Attribute** | **Type** | **Required** | **Short description** |
|---|---|---|---|
| access_log | Boolean | U | Activate separated access log |
| antibot | Boolean | U | JavaScript based bot detection |
| antibot_threshold | Number | U | Ratio of uncached to cached requests |
| balancing_method | String | U | Specifies with which method requests are balanced between upstream servers |
| block_not_whitelisted | Boolean | U | Block all IPs, which are not whitelisted |
| cache_enabled | Boolean | U | Turn caching on or off |
| cache_revalidate | Boolean | U | Enable stale cache item revalidation |
| cdn | Boolean | U | Use subdomain as Content Delivery Node (CDN) |
| diffie_hellman_exchange | Integer | U | The Diffie-Hellman key exchange parameter length |
| hsts | Boolean | U | HSTS Strict Transport Security (HSTS) |
| hsts_include_subdomains | Boolean | U | HSTS includeSubDomains directive |
| hsts_max_age | Number | U | HSTS max-age |
| hsts_preload | Boolean | U | HSTS preload directive |
| forwarded_for_replacement | String | U | Set your own X-Forwarded-For header |
| image_optimization | Boolean | U | Optimization of images |
| ipv6_active | Boolean | U | Allow connections via IPv6 to your systems |
| myra_ssl_header | Boolean | U | Activates the X-Myra-SSL Header |
| next_upstream | String | U | Specifies the error that mark the current upstream as "down" |
| only_https | Boolean | U | Shall the origin server always be requested via HTTPS? |
| origin_connection_header | String | U | Connection header |
| proxy_cache_bypass | String | U | Name of the cookie which forces Myra to deliver the response not from cache |
| proxy_cache_stale | String | U | Determines in which cases a stale cached response can be used when an error occurs |
| proxy_connect_timeout | Number | U | Timeout for establishing a connection to the upstream server |
| proxy_read_timeout | Number | U | Timeout for reading the upstream response |

**C** = Create, **U** = Update, **D** = Delete

Table 12: Attributes for (sub)domain settings

| Attribute | Type | Required | Short description |
|---|---|---|---|
| request_limit_block | Boolean | U | Show CAPTCHA after exceeding the configured request limit |
| request_limit_level | Number | U | Sets how many requests are allowed from an IP per minute |
| rewrite | String | U | Enable the JavaScript optimization |
| rewrite_regex | String | U | Regular Expression to match the JavaScript files to optimize |
| security_level | String | U | Activate Web Application Filter |
| source_protocol | String | U | Protocol to query the origin server |
| spdy | Boolean | U | Activates the SPDY protocol |
| streaming_regex | String | U | Regular expression to match the streaming URL |

**C** = Create, **U** = Update, **D** = Delete

**access_log**

Activate separated access log. A access log from each Myra node delivering your website will be saved. You can download the access log files via sftp from custlogs.myracloud.com.
**Default value:** false
**Allowed values:** true, false

**antibot**

Activate JavaScript based bot detection. Bots will then be automatically blocked when accessing your website.
**Default value:** no
**Allowed values:** yes, no, auto

**antibot_threshold** (Advanced)

Sets the ratio of uncached to cached requests. This options is only used when Antibot is set to "auto".
**Default value:** 20
**Allowed values:** 0 - 100

**balancing_method**

Specifies with which method requests are balanced between upstream servers. The default behavior is the round-robin balancing. The value ip_hash will cause Myra to forward the same client IP always to the same upstream server. The value least_conn will cause Myra to forward the request to the upstream server with least connections.
**Default value:** round_robin
**Allowed values:** round_robin, ip_hash, least_conn

**block_not_whitelisted**

Block all IPs which are not whitelisted in the IP filter settings.
**Default value:** false
**Allowed values:** true, false

**cache_enabled**

Turn caching on or off. If you enable the cache, you also have to define the objects to be cached in the cache settings.
**Default value:** false
**Allowed values:** true, false

**cache_revalidate**

If enabled, expired cache items will be requested with the additional HTTP header "If-Modified-Since" and "If-None-Match".
**Default value:** true
**Allowed values:** true, false

**cdn** (Advanced)

Should this subdomain be used as Content Delivery Node (CDN). After enabling the CDN you will be able to create buckets and upload files using the Myra upload API.
**Default value:** false
**Allowed values:** true, false

**diffie_hellman_exchange**

Defines the size of the Diffie-Hellman key exchange parameters in bits. Please, note that Java 6 and 7 do not support Diffie-Hellman parameters larger than 1024 bits. If your server expects to receive connections from java 6 clients and wants to enable PFS, it must provide a DHE parameter of 1024 bits.
**Default value:** 1024
**Allowed values:** 1024, 2048

**enforce_cache_ttl** (Advanced)

Enforce using given cache TTL settings instead of origin cache information. This will set the Cache-Control header max-age to the given TTL.
**Default value:** false
**Allowed values:** true, false

**hsts** (Advanced)

Enable HSTS protection for a domain. This will tell browsers to use secure https connections only when interacting with your domain.
**Default value:** false
**Allowed values:** true, false

**hsts_include_subdomains** (Advanced)

This will extend the HSTS protection for all subdomains.
**Default value:** false
**Allowed values:** true, false

**hsts_max_age** (Advanced)

Specified how long the HSTS header is valid before the browser has to revalidate.
**Default value:** 31536000
**Allowed values:** 0 - 2147483647

**hsts_preload** (Advanced)

Allow the domain to be added to the HSTS preload list used by all major browsers (https://hstspreload.appspot.com/).
**Default value:** false
**Allowed values:** true, false

**forwarded_for_replacement**

Set your own X-Forwarded-For header. The value of this header contains the originating IP address of the request.
**Default value:** X-Forwarded-For
**Allowed values:** A valid HTTP header name

**image_optimization**

Activate lossless optimization of JPEG and PNG images (recommended setting).
**Default value:** true
**Allowed values:** true, false

**ipv6_active**

Allow connections via IPv6 to your systems. IPv4 connections will be forwarded in any case.
**Default value:** true
**Allowed values:** true, false

**myra_ssl_header** (Advanced)

Activate the X-Myra-SSL Header, which indicates if a request was received via SSL.
**Default value:** false
**Allowed values:** true, false

**next_upstream** (Advanced)

Specify in which case the current upstream should be marked as "down". The values can be arbitrary combined, expect the value "off".
**Default value:** [error, timeout, invalid_header]
**Allowed values:** error, timeout, invalid_header, http_500, http_502, http_503, http_504, off

**error, timeout:**

An generic error or timeout occured while connecting to the origin server, sending the request or reading the response.

**invalid_header:**

Invalid or empty response from the origin server.

**http_500, http_502, http_503, http_504:**

Origin server responded with the matching response code.

**off:**

Completly disables the next_upstream behavior.

**only_https**

If activated, Myra will forward all requests to the origin using HTTPS regardless of the used protocol of the originating request.
**Default value:** false
**Allowed values:** true, false

**origin_connection_header** (Advanced)

Sets the Connection header, which is transmitted to the origin with a request.

**Default value:** none

**Allowed values:** upgrade, close, none

**proxy_cache_bypass** (Advanced)

Defines the name of the cookie which forces Myra to deliver the response not from cache. The values of the cookie must be not empty or equal to 0 to enable bypassing. **Default value:** MYRA_NOCACHE

**Allowed values:** A string containing only letters, digits and underscores.

**proxy_cache_stale**

Determines in which cases a stale cached response can be used when an error occurs during communication with your server. The values can be arbitrary combined, expect the value "off".

**Default value:** [updating]

**Allowed values:** error, timeout, invalid_header, updating, http_500, http_502, http_503, http_504, http_403, http_404, off

**error, timeout:**

An generic error or timeout occured while connecting to the origin server, sending the request or reading the response.

**invalid_header:**

Invalid or empty response from the origin server.

**http_500, http_502, http_503, http_504:**

Origin server responded with the matching response code.

**off:**

Completly disables the next_upstream behavior.

**updating:**

Allows sending a stale cached response while it is being updated to minimize connctions to the origin servers.

**proxy_connect_timeout**

Defines a timeout in seconds for establishing a connection with the origin server. The timeout cannot be greater than 60 seconds.

**Default value:** 20

**Allowed values:** 1 - 60

**proxy_read_timeout**

Defines a timeout in seconds for reading a response from the proxied server. The timeout is set only between two successive read operations, not for the transmission of the whole response.

**Default value:** 1200

**Allowed values:** 1 - 1200

**request_limit_block**

If activated, the user has to solve a CAPTCHA after exceeding the configured request limit.
**Default value:** false
**Allowed values:** true, false

**request_limit_level**

Define how many requests are allowed from an IP per minute. If this limit is reached, the IP will be blocked. If request_limit_block is enabled, the user can solve a CAPTCHA to unblock his IP address.
**Default value:** 6000
**Allowed values:** 100, 1000, 4000, 6000, 12000, 18000

**request_limit_report**

If activated, an email will be send containing blocked ip addresses that exceeded the configured request limit.
**Default value:** false
**Allowed values:** true, false

**rewrite** (Advanced)

Enable automated JavaScript optimization. All JavaScript is collected and executed at the end of the page. This significantly decreases the DOM content loaded time. If not all JavaScript files should be collected you can set the value to "regex" and specify the regex to use while matching filenames in the option "rewrite_regex".
**Default value:** no
**Allowed values:** yes, no, regex

**rewrite_regex** (Advanced)

Regular Expression which is used to match the JavaScript files which should be optimized. All files not matching this pattern will not be touched. This value will only be used if the option "rewrite" has the value "regex".
**Default value:** (empty)
**Allowed values:** Valid regular expression

**security_level**

Activate the Web Application Filter to block Layer 7 attacks.
**Default value:** standard
**Allowed values:** high, standard, off

**source_protocol**

Define which protocol should be used when passing a request to your servers. The value "same" will ensure that the same protocol is used as in the originating request to Myra. The "http" and "https" value will force Myra to always use the specified protocol when connecting.
**Default value:** same
**Allowed values:** same, http, https

**spdy**

Activate the high performance HTTP/2 protocol. Please note that you have to enable HTTPS for Myra to get HTTP/2 enabled. **Default value:** false
**Allowed values:** true, false

**streaming_regex** (Advanced)

Regular expression, which is used to match the streaming URL.
**Default value:** (empty)
**Allowed values:** Valid regular expression

# 8 Cache

## 8.1 Overview

The cache settings allows you to configure paths, which should be cached by Myra.

## 8.2 Usage

All cache settings use URL specific parameters; a list of these parameters can be found in the table below.

Table 13: Parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain you are trying to list cache rules for. |
| page | Number | Yes | Number of the page you want to retrieve (1-based). |

### 8.2.1 Listing

To list all CacheSetting objects belonging to a given domain you have to use the following REST request:

```
GET /{language}/rapi/cacheSettings/{domain}/{page} HTTP/1.1
Host: api.myracloud.com
Authorization: MYRA {apiKey}:{signature}
Date: 2014-05-02T07:17+0200
```

Listing 35: Example for listing CacheSetting objects

A successful call will return a response with a HTTP status of 200 and the following body:

```
{
"error"    : false,
"list"     : [
    {
        "id"            : 1,
        "modified"      : "2013-12-09T11:35:00+0100",
        "created"       : "2013-12-09T11:35:00+0100",
        "path"          : "/",
        "ttl"           : "300",
        "type"          : "prefix"
    }
],
"page"     : 1,
"count"    : 2,
"pageSize" : 50
}
```

Listing 36: Example

### 8.2.2 Create

To create a new CacheSetting you need to send a CacheSetting object without the attributes "id", "created", and "modified". All those attributes are generated by the server and returned to you after a successful insert.

```
PUT /{language}/rapi/cacheSettings/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "path"         : "/",
    "ttl"          : "300",
    "type"         : "prefix"
}
```

Listing 37: Example

A successful insert will return a ResultVO with the newly created CacheSetting returned as in the targetObject list.

```
{
"error": false,
"violationList": [],
"targetObject": [
    {
        "id"           : 1,
        "modified"     : "2014-03-06T18:21:49+0100",
        "created"      : "2014-03-06T18:21:49+0100",
        "path"         : "/",
        "ttl"          : "300",
        "type"         : "prefix"
    }
  ]
}
```

Listing 38: Result

### 8.2.3 Update

Updating a CacheSetting is very similar to create a new one. The main difference is that an update will need the "id" and "modified" attributes to identify the version of the object you are trying to update.

```
POST /{language}/rapi/cacheSettings/{domain} HTTP/1.1
Host: api.myracloud.com
Authorization: MYRA {apiKey}:{signature}
Date: 2014-05-02T07:17+0200

{
    "id"           : 1,
    "modified"     : "2014-03-06T18:21:49+0100",
    "created"      : "2014-03-06T18:21:49+0100",
    "path"         : "/assets",
    "ttl"          : "300",
    "type"         : "prefix"
}
```

Listing 39: Example for updating a CacheSetting

To see a valid response, take a look at the response from the create section. After an update your configuration will be queued and deployed to our system.

### 8.2.4 Delete

To delete a cache rule you only need to send "id" and "modified" as body content.

```
DELETE /{language}/rapi/cacheSettings/{domain} HTTP/1.1
Host: api.myracloud.com
Authorization: MYRA {apiKey}:{signature}
Date: 2014-05-02T07:17+0200

{
    "id"             : 3,
    "modified"       : "2013-12-09T11:30:00+0100"
}
```

Listing 40: Example for deleting a CacheSetting

After a delete operation is done your configuration will be queued and deployed to our system.

## 8.3 Full version of a Cache object

```
{
    "id"           : 1,
    "modified"     : "2013-12-12T11:35:00+0100",
    "created"      : "2013-12-09T11:35:00+0100",
    "path"         : "/",
    "ttl"          : "300",
    "type"         : "exact"
}
```

Listing 41: Full version of a Cache object

Table 14: Attributes for cache settings

| Attribute | Type | Required | Short description |
|---|---|---|---|
| id | Number | UD | Id of the object |
| modified | Date | UD | Date of last modification |
| created | Date | - | Date of creation |
| path | String | CU | Path which must match to cache response |
| ttl | String | CU | Time to live |
| type | String | CU | Type how path should match |

**C** = Create, **U** = Update, **D** = Delete

**id**

Id is an unique identifier for an object. This value is always a number type and cannot be set while inserting a new object. To update or delete a cache setting it is necessary to add this attribute to your object.

**modified**

Identifies the version of the object. To ensure that you are updating the most recent version and not overwriting other changes, you always have to add the modified timestamp for updates and deletes. This value is always a date type with an ISO8601 format.

**created**

Created is a date type attribute with an ISO8601 format. It will be created by the server after creating a new cache setting object. This value is only informational so it is not necessary to add this an attribute to any API call.

**path**

A request will be matched against this path to decide if this request is cacheable or not. It is possible to write a regexp in this attribute. It is not allowed to use start ’^’ or end ’$’ regexp characters as it they are generated depending on the given type.

**ttl**

Time to live limits the lifespan of a cached response for the given path. This is a numeric value and is given in seconds. Special case is 'like origin server', which uses the TTL returned by your origin server.

**type**

This will tell the server how to match the given path against a request. Available options are 'prefix', 'suffix' and 'exact'.

# 9 CacheClear

## 9.1 Overview

CacheClear allows you to delete your cached files on all Myra nodes. You can use wildcard expressions to handle complex clear actions as well.

## 9.2 Usage

All cache clear actions use URL specific parameters; a list of these parameters can be found in the table below.

Table 15: Parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain you are trying to clear cache for. |

### 9.2.1 Clear

To enqueue a clear operation you need to send a CacheClear object. The "fqdn" attribute in the object must be the domain or a subdomain of the domain mentioned in the URL.

```
PUT /{language}/rapi/cacheClear/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "fqdn"           : "www.example.com",
    "resource"       : "/*.jpg",
    "recursive"      : true
}
```

Listing 42: Example for a cache clear operation

A successfully enqueued cache clear will return a ResultVO with the given CacheClear in the targetObject list.

```
{
"error": false,
"violationList": [],
"targetObject": [
    {
        "fqdn"           : "www.example.com",
        "resource"       : "/*.jpg",
        "recursive"      : true
    }
  ]
}
```

Listing 43: Result

### 9.2.2 Clear of a whole domain

To enqueue a cache clear operation for your complete domain, you have to slightly change the domain parameter. You have to use ALL:domain instead of domain only. The "fqdn" attribute in the following object will be ignored.

```
PUT /{language}/rapi/cacheClear/ALL:{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "resource"       : "/*.jpg",
    "recursive"      : true
}
```

Listing 44: Example

A successfully enqueued cache clear will return a ResultVO. For each active subdomain the API returns a CacheClearVO as part of the targetObject list. Every CacheClearVO contains the subdomain as "fqdn" attribute and the rest of the given data such as "resource" and "recursive".

```
{
"error": false,
"violationList": [],
"targetObject": [
    {
        "fqdn"           : "www.example.com",
        "resource"       : "/*.jpg",
        "recursive"      : true
    },
    {
        "fqdn"           : "www2.example.com",
        "resource"       : "/*.jpg",
        "recursive"      : true
    }
  ]
}
```

Listing 45: Result

### 9.2.3 Pattern matching

Internally we use the fnmatch (flags=FNM_PATHNAME) function to find the matching resources that should be deleted. To allow you to do recursive deletion in folders, the flag "recursive" was added.

| Pattern | Recursive | Resource | Result |
|---|---|---|---|
| /*.js | No | /main.js | **match** |
| /*.js | No | /folder/main.js | **no match** |
| /*.js | No | /testmain.js | **no match** |
| *.js | Yes | /assets/script.js | **match** |
| *.js | Yes | /assets/jquery/jquery.js | **match** |
| *.js | Yes | /main.js | **match** |
| *.js | Yes | /main.css | **no match** |
| *.js | Yes | /assets/js/source.map | **no match** |
| /assets/*.js | No | /assets/script.js | **match** |
| /assets/*.js | No | /asset/script.js | **no match** |
| /assets/*.js | No | /main.js | **no match** |
| /assets/*.js | No | /folder/js/script.js | **no match** |
| /assets/*.js | Yes | /assets/script.js | **match** |
| /assets/*.js | Yes | /assets/jquery/jquery.js | **match** |
| /assets/*.js | Yes | /main.js | **no match** |
| /assets/*.js | Yes | /js/angular.js | **no match** |
| /*.* | Yes | /main.js | **match** |
| /* | Yes | /main.js | **match** |

## 9.3 Full version of a CacheClear object

```
{
    "fqdn"          : "subdomain.example.de",
    "resource"      : "/example-dir/*.jpg",
    "recursive"     : false
}
```

Listing 46: Full version of a CacheClear object

Table 17: Attributes for CacheClear

| **Attribute** | **Type** | **Required** | **Short description** |
|---|---|---|---|
| fqdn | String | C | Subdomain name for a maintenance entry |
| resource | String | C | Resource pattern to delete or empty for a full cache clear |
| recursive | Boolean | C | Should the pattern also matched against subdirectories |

**C** = Create, **U** = Update, **D** = Delete

**fqdn**

Shows a FQDN (fully qualified domain name) for a maintenance. This attribute shows the domain to handle maintenance for.

**resource**

A pattern that is used to find the resources you are going to delete. This pattern can contain * as wildcard character.

**recursive**

Should the pattern also matches against subdirectories.

# 10 Redirects

## 10.1 Overview

You can manage your redirects directly with Myra, which saves requests to your origin server. The redirect option of Myra offers you the possibility of using regular expressions.

## 10.2 Usage

All redirect requests use URL specific parameters; a list of these parameters can be found in the table below.

Table 18: URL parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain you are trying to list redirects for. |
| page | Number | Yes | Number of the page you want to retrieve (1-based). |

Table 19: Query parameters (Deprecated)

| Parameter | Type | Required | Description |
|---|---|---|---|
| expertMode | Bool | No | When creating a new redirect with expertMode enabled, Myra will skip checking the rule for endless looping. |

### 10.2.1 Listing

The listing operation returns a list of redirects for the given domain. The REST request must have the following format:

```
GET /{language}/rapi/redirects/{domain}/{page} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}
```

Listing 47: Example request

A successful call will return a response with a HTTP status code of 200 and the following body:

```
{
"error"   : false,
"list"    : [
    {
        "id"             : 1,
        "modified"       : "2013-12-13T11:30:00+0100",
        "created"        : "2013-12-09T11:30:00+0100",
        "subDomainName" : "www.example.com.",
        "source"         : "/",
        "destination"    : "$scheme://www.example.com/a.html",
        "type"           : "redirect",
        "matchingType"   : "prefix"
    }
],
"page"    : 1,
"count"   : 1,
"pageSize": 50
}
```

Listing 48: Example response

### 10.2.2 Create

To create a new redirect it is necessary to send a Redirect object without the attributes "id" and "modified". Both attributes will be generated by the server and returned after a successful insert is done.

```
PUT /{language}/rapi/redirects/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "source"          : "/example-source",
    "destination"     : "http://www.myracloud.com",
    "type"            : "permanent",
    "matchingType"    : "prefix",
    "expertMode"      : "false"
}
```

Listing 49: Example

Using the "expertMode" attribute

```
PUT /{language}/rapi/redirects/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "source"          : "/example-source",
    "destination"     : "http://www.myracloud.com",
    "type"            : "permanent",
    "matchingType"    : "prefix",
    "expertMode"      : "true"
}
```

Listing 50: Example

A successful insert will return a ResultVO with the new created Redirect object.

```
{
"error": false,
"violationList": [],
"targetObject": [
    {
      "id"           : 1,
      "modified"     : "2014-03-06T18:21:49+0100",
      "created"      : "2014-03-06T18:21:49+0100",
      "subDomainName": "www.example.com",
      "source"       : "/example-source",
      "destination"  : "http://www.myracloud.com",
      "type"         : "permanent",
      "matchingType" : "prefix"
    }
  ]
}
```

Listing 51: Result

After a successful insert your configuration will be queued to and deployed to our system.

### 10.2.3 Update

Updating a redirect is very similar to creating a new one. The main difference is that an update will need the generated "id" and "modified" attributes to identify the object you are trying to update.

```
POST /{language}/rapi/redirects/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "id"             : 3,
    "modified"       : "2013-12-09T11:30:00+0100",
    "source"         : "/example-source",
    "destination"    : "http://www.myracloud.com",
    "type"           : "permanent",
    "matchingType"   : "suffix",
    "expertMode"     : "false"
}
```

Listing 52: Example

A successful update will return the same type of object as the create operation. After an update your configuration will be queued and deployed to our system.

### 10.2.4 Delete

For deleting a Redirect it is only necessary to send the "id" and "modified" attributes as body content.

```
DELETE /{language}/rapi/redirects/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "id"             : 3,
    "modified"       : "2013-12-09T11:30:00+0100"
}
```

Listing 53: Example

After a delete operation your configuration will be queued and deployed to our system.

## 10.3 Full version of a Redirect object

```
{
    "id"            : 1,
    "modified"      : "2014-03-12T18:21:49+0100",
    "created"       : "2014-03-06T18:21:49+0100",
    "subDomainName" : "www.example.com",
    "source"        : "/example-source",
    "destination"   : "http://www.myracloud.com",
    "type"          : "permanent",
    "matchingType"  : "prefix",
    "expertMode"    : "false"
}
```

Listing 54: Full version of a Redirect object

Table 20: Attributes for Redirect

| Attribute | Type | Required | Short description |
|---|---|---|---|
| id | Number | UD | Id of the object |
| modified | Date | UD | Date of last modification |
| created | Date | - | Date of creation |
| subDomainName | String | - | Subdomain of redirect |
| source | String | CU | Location to match against |
| destination | String | CU | Target where redirect should point to |
| type | String | CU | Type of redirection |
| matchingType | String | - | Type to match the redirect defaults to 'exact'. (Default will change in upcoming releases to 'prefix') |
| expertMode | Bool | CU | Enable or disable the expertMode |

**C** = Create, **U** = Update, **D** = Delete

**id**

Id is an unique identifier for an object. This value is always a number type and cannot be set while inserting a new object. To update or delete a Redirect it is necessary to add this attribute to your object.

**modified**

Identifies the version of the object. To ensure that you are updating the most recent version and not overwriting other changes, you always have to add modified for updates and deletes. This value is always a date type with an ISO8601 format.

**created**

Created is a date type attribute with an ISO8601 format. It will be created by the server after creating a new Redirect object. This value is only informational so it is not necessary to add this attribute to any API call.

**subDomainName**

Identifies the subdomain via a FQDN (Full Qualified Domain Name) where this redirect belongs to. This value cannot be changed through the object's attribute as it is set via URL parameter.

**source**

Location to match your query against, it is also possible to match against a regexp instead of hard coded locations.

**destination**

The destination you want your customer redirect to. This can be a valid HTTP(S) address or a relative location on your domain.

**type**

The redirect type how your customer is redirected. This can be an HTTP 301 (permanent) redirect, which is cacheable by browsers and search crawlers. Another option is an HTTP 302 (redirect) redirect which is usually not cached by browsers and crawlers. Valid options are 'permanent' and 'redirect'.

**matchingType**

The matching type allows you to change the way how the redirect is matched. This field allows three different values: 'prefix', 'suffix', and 'exact'.

**prefix**

If you're using matching type 'prefix' the '^' will be placed in front of your redirect.

**suffix**

The option 'suffix' places a '$' at the end of your redirect.

**exact**

On 'exact' your regex will be used like it is without changes.

**expertMode**

Enable or disable the expertMode. When expertMode is enabled, Myra will skip checking the rule for endless looping.

# 11 IP black/whitelist

## 11.1 Overview

The IP black/whitelist of Myra lets you grant or deny access from individual IP addresses or subnets.

## 11.2 Usage

All IP black/whitelist REST calls use URL specific parameters; a list of these parameters can be found in the table below.

Table 21: Parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain or subdomain you are trying to list IP filters for. |
| page | Number | Yes | Number of the page you want to retrieve (1-based). |

### 11.2.1 Listing

To list all IP black/whitelist objects belonging to a given domain you have to use the following REST request:

```
GET /{language}/rapi/ipfilter/{domain}/{page} HTTP/1.1
Host: api.myracloud.com
Authorization: MYRA {apiKey}:{signature}
Date: 2014-05-02T07:17+0200
```

Listing 55: Example for listing IP black/whitelist objects

A successful call will return a response with a HTTP status of 200 and the following body:

```
{
"error"    : false,
"list"     : [
    {
        "id"           : 1,
        "modified"     : "2013-12-09T11:35:00+0100",
        "created"      : "2013-12-09T11:35:00+0100",
        "type"         : "WHITELIST",
        "value"        : "1.2.3.4/32"
    }
],
"page"     : 1,
"count"    : 1,
"pageSize" : 50
}
```

Listing 56: Example

### 11.2.2 Create

To create a new IP black/whitelist setting, you need to send an IpFilter object without the attributes "id", "created" and "modified". All those attributes are generated by the server and returned to you after a successful insert.

```
PUT /{language}/rapi/ipfilter/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "type"  : "BLACKLIST",
    "value" : "1.2.3.4/32"
}
```

Listing 57: Example

```
PUT /{language}/rapi/ipfilter/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "type"  : "WHITELIST",
    "value" : "192.168.0.0/24"
}
```

Listing 58: Example 2

A successful insert will return a ResultVO with the newly created IpFilter object returned in the targetObject list.

```
{
"error": false,
"violationList": [],
"targetObject": [
    {
        "id"       : 1,
        "modified" : "2014-03-06T18:21:49+0100",
        "created"  : "2014-03-06T18:21:49+0100",
        "type"     : "WHITELIST",
        "value"    : "1.2.3.4/32"
    }
  ]
}
```

Listing 59: Result

### 11.2.3 Update

Updating an IP black/whitelist setting is very similar to creating a new one. The main difference is that an update will need the "id" and "modified" attributes to identify the version of the object you are trying to update.

```
POST /{language}/rapi/ipfilter/{domain} HTTP/1.1
Host: api.myracloud.com
Authorization: MYRA {apiKey}:{signature}
Date: 2014-05-02T07:17+0200

{
    "id"       : 3,
    "modified" : "2013-12-09T11:30:00+0100",
    "type"     : "WHITELIST",
    "value"    : "5.6.7.8/32"
}
```

Listing 60: Example for updating an IP black/whitelist setting

To see a valid response, take a look at the response from the create section. After an update your configuration will be queued and deployed to our system.

### 11.2.4 Delete

To delete a filter rule you only need to send "id" and "modified" as body content.

```
DELETE /{language}/rapi/ipfilter/{domain} HTTP/1.1
Host: api.myracloud.com
Authorization: MYRA {apiKey}:{signature}
Date: 2014-05-02T07:17+0200

{
    "id"       : 3,
    "modified" : "2013-12-09T11:30:00+0100"
}
```

Listing 61: Example for deleting an IP black/whitelist setting

After a delete operation is done your configuration will be queued and deployed to our system.

## 11.3 Full version of a IpFilter object

```
{
    "id"            : 1,
    "modified"      : "2013-12-12T11:35:00+0100",
    "created"       : "2013-12-09T11:35:00+0100",
    "type"          : "WHITELIST",
    "value"         : "1.2.3.4/32"
}
```

Listing 62: Full version of an IpFilter object

Table 22: Attributes for IP filter

| Attribute | Type | Required | Short description |
|---|---|---|---|
| id | Number | UD | Id of the object |
| modified | Date | UD | Date of last modification |
| created | Date | - | Date of creation |
| type | String | CU | Type of the filter rule ("WHITELIST", "BLACKLIST") |
| value | String | CU | Single IP address or CIDR notation |

**C** = Create, **U** = Update, **D** = Delete

**id**

Id is an unique identifier for an object. This value is always a number type and cannot be set while inserting a new object. To update or delete a cache setting it is necessary to add this attribute to your object.

**modified**

Identifies the version of the object. To ensure that you are updating the most recent version and not overwriting other changes, you always have to add the modified timestamp for updates and deletes. This value is always a date type with an ISO8601 format.

**created**

Created is a date type attribute with an ISO8601 format. It will be created by the server after creating a new cache setting object. This value is only informational so it is not necessary to add this an attribute to any API call.

**type**

This specifies how the rule is applied.

**value**

The value of an IP filter rule can contain a single IP address or a CIDR notation. IPv4 and IPv6 are both supported. An IP filter for IPv6 can only contain a /128 subnet.

# 12 Maintenance

## 12.1 Overview

The maintenance functionality allows you to set a maintenance page which is directly served from Myra servers. This is useful when you are going to maintain your servers and remove all load from your server.

## 12.2 Usage

For a better user experience, our system will periodically remove all expired maintenance pages.

All maintenance calls use URL specific parameters; a list of these parameters can be found in the table below.

Table 23: Parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain you are trying to list maintenance pages for. |
| page | Number | Yes | Number of the page you want to retrieve (1-based). |

### 12.2.1 Listing

To get a list of all maintenance objects belonging to your given domain you have to use the following REST request:

To list the maintenance pages of a single subdomain replace the domain parameter with your subdomain. This will return only the maintenance pages for the given subdomain.

```
GET /{language}/rapi/maintenance/{domain}/{page} HTTP/1.1
Host: api.myracloud.com
Authorization: MYRA {apiKey}:{signature}
Date: 2014-05-02T07:17+0200
```

Listing 63: Example for listing Maintenance objects

A successful call will return a response with a HTTP status of 200 and the following body:

```
{
"error"    : false,
"list"     : [
    {
        "id"           : 1,
        "modified"     : "2013-12-09T11:35:00+0100",
        "created"      : "2013-12-09T11:35:00+0100",
        "fqdn"         : "www.example.de.",
        "content"      : "<html>Your maintenance page</html>",
        "start"        : "2013-12-10T00:00:00+0100",
        "end"          : "2013-12-10T09:00:00+0100",
        "active"       : false
    }
],
"page"     : 1,
"count"    : 1,
"pageSize" : 50
}
```

Listing 64: Example

### 12.2.2 Create

To create a new scheduled Maintenance you need to send a Maintenance object without the attributes "id", "created", "modified", "active", and "fqdn". All those attributes are generated by the server and returned to you after a successful insert.

If your current time is between "start" and "end" date your maintenance page will be activated immediately!

To enqueue a maintenance page for a subdomain you have to change the domain URL parameter to the subdomain you want to create a maintenance page for.

To enqueue a maintenance page for all subdomains you have to change the domain URL parameter to ALL:domain.

```
PUT /{language}/rapi/maintenance/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "content" : "<html>Your maintenance page</html>",
    "start"   : "2014-09-01T00:00:00+0200",
    "end"     : "2014-09-01T09:00:00+0200"
}
```

Listing 65: Example

If you prefer a default page from Myra you can do that by sending the following request:

```
PUT /{language}/rapi/maintenance/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "start"          : "2014-09-01T00:00:00+0200",
    "end"            : "2014-09-01T09:00:00+0200",
    "defaultPage"    : {
        "facebook"   : "URL",
        "twitter"    : "URL",
        "custom"     : {
            "label"  : "label",
            "url"    : "URL"
        }
    }
}
```

Listing 66: Example of creating a default maintenance page

You can also reference an already existing maintenance page to reuse it's content instead of sending the content in the request:

```
PUT /{language}/rapi/maintenance/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "start"          : "2014-09-01T00:00:00+0200",
    "end"            : "2014-09-01T09:00:00+0200",
    "contentFrom"    : "test.domain.de"
}
```

Listing 67: Example of referencing content of an maintenance page

A successful insert will return a ResultVO with the newly created Maintenance returned as targetObject. The response will also show whether the new maintenance was instantly activated by the API. When you use a default page instead of using your own, the default page will be returned in the "content" attribute. All attributes including "defaultPage" are optional. Custom link is only usable when using custom.label and custom.url together.

```
{
"error"         : false,
"violationList" : [],
"targetObject"  : [
    {
        "id"           : 1,
        "modified"     : "2014-03-06T18:21:49+0100",
        "created"      : "2014-03-06T18:21:49+0100",
        "fqdn"         : "www.example.com",
        "start"        : "2014-09-01T00:00:00+0200",
        "end"          : "2014-09-01T09:00:00+0200",
        "content"      : "<html>Maintenance page</html>",
        "active"       : false
    }
  ]
}
```

Listing 68: Result of a default maintenance page

### 12.2.3 Update

Updating a Maintenance is very similar to create a new one. The main difference is that an update will need the "id" and "modified" attributes to identify the version of the object you are trying to update.

```
POST /{language}/rapi/maintenance/{domain} HTTP/1.1
Host: api.myracloud.com
Authorization: MYRA {apiKey}:{signature}
Date: 2014-05-02T07:17+0200

{
    "id"             : 3,
    "modified"       : "2013-12-09T11:30:00+0100",
    "start"          : "2014-09-04T00:00:00+0100",
    "end"            : "2014-09-04T09:00:00+0100",
    "content"        : "<html>Maintenance page</html>"
}
```

Listing 69: Example for updating a Maintenance

To see a valid response, take a look at the response from the create section. After an update your configuration will be queued and deployed to our system.

### 12.2.4 Delete

If the maintenance page you are going to delete is active, then it will be automatically disabled!

To delete a maintenance you only need to send "id" and "modified" as body content.

```
DELETE /{language}/rapi/maintenance/{domain} HTTP/1.1
Host: api.myracloud.com
Authorization: MYRA {apiKey}:{signature}
Date: 2014-05-02T07:17+0200

{
    "id"             : 3,
    "modified"       : "2013-12-09T11:30:00+0100"
}
```

Listing 70: Example for deleting a Maintenance

After a delete operation is done your configuration will be queued and deployed to our system.

## 12.3 Full version of a Maintenance object

```
{
    "id"           : 1,
    "modified"     : "2013-12-11T11:35:00+0100",
    "created"      : "2013-12-09T11:35:00+0100",
    "fqdn"         : "subdomain.example.de",
    "start"        : "2014-08-27T09:00:00+0100",
    "end"          : "2014-08-27T18:00:00+0100",
    "content"      : "<html>content</html>",
    "active"       : true,
    "defaultPage"  : {
        "twitter"       : "URL",
        "facebook"      : "URL",
        "custom"        : {
          "label"       : "label",
          "url"         : "URL"
        }
    }
}
```

Listing 71: Full version of a Maintenance object

Table 24: Attributes for Maintenance

| Attribute | Type | Required | Short description |
|---|---|---|---|
| id | Number | UD | Id of the object |
| modified | Date | UD | Date of last modification |
| created | Date | - | Date of creation |
| fqdn | String | CU | Subdomain name for a maintenance entry |
| start | Date | CU | Date of maintenance to start |
| end | Date | CU | Date of maintenance to end |
| content | String | CU | HTML content to show when maintenance is active |
| contentFrom | String | C | Use content of maintenance page from the given fqdn |
| active | Boolean | - | Shows if a maintenance is currently active |
| defaultPage | Object | - | Allows to append urls for twitter and facebook used by the generated default page |

**C** = Create, **U** = Update, **D** = Delete

**id**

Id is an unique identifier for an object. This value is always a number type and cannot be set, while inserting a new object. To update or delete a Maintenance it is necessary to add this attribute to your object.

**modified**

Identifies the version of the object. To ensure that you are updating the most recent version and not overwriting other changes, you always have to add modified for updates and deletions. This value is always a date type with an ISO8601 format.

**created**

Created is a date type attribute with an ISO8601 format. It will be created by the server after creating a new Maintenance object. This value is only informational so it is not necessary to add this an attribute to any API call.

**fqdn**

Shows a FQDN (fully qualified domain name) for a maintenance. This attribute shows the domain to handle maintenance for.

**start**

Start is a date type attribute with an ISO8601. This attribute shows the start date for a maintenance. This date have to be lower than end or null to start now.

**end**

End is a date type attribute with an ISO8601 and shows the end date for a maintenance. This date have to be higher than start or null to end now.

**content**

HTML content to show as maintenance page. Please note that it is not possible to include resources from the domain you have set to maintenance mode. If your maintenance page contains images use a different domain or use inline base64 encoded images.

**contentFrom**

This property can be used instead of the property content to reference an existing maintenance page's content. Instead of sending the actual content, specify a valid FQDN here. This will copy the content from the referenced maintenance page to the newly created.

**active**

This information shows if this a maintenance page is currently active. You cannot set this attribute directly instead you have to set start and end attribute to activate maintenance.

**defaultPage**

This allows an API user to set a maintenance with a default page generated by Myra. To append customer specific parts (Facebook and Twitter) you can append this keys to the given object and set an URL to the specific service. Both URLs can be used simultaneously. The defaultPage option also allows you to set a customer specific link on the generated page.

# 13 SSL certificates

## 13.1 Overview

This endpoint lets you manage your SSL certifcates.

## 13.2 Usage

All SSL requests use URL specific parameters. A list of these parameters can be found in the table below.

Table 25: URL parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain you are working on. |
| page | Number | Yes | Number of the page you want to retrieve (1-based). |

### 13.2.1 Listing

To list all SSL certificates that belong to a given domain, you have to use the following REST request:

```
GET /{language}/rapi/certificates/{domain}/{page} HTTP/1.1
Host: api.myracloud.com
Date: 2014-06-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}
```

Listing 72: Example request for listing SSL certificate objects

A successful call will return a response with a HTTP status code of 200 and the following body:

```
{
"error"    : false,
"list"     : [
  {
    "objectType"          : "SslCertVO",
    "id"                  : 98765,
    "modified"            : "2016-06-28T15:33:25+0200",
    "created"             : "2016-06-28T15:33:25+0200",
    "subject"             : "OU=Domain Control Validated Free SSL, CN=myracloud.com",
    "algorithm"           : "RSA-SHA1",
    "validFrom"           : "2012-10-18T00:00:00+0000",
    "validTo"             : "2013-01-16T23:59:59+0000",
    "fingerprint"         : "7A:E6:79:37:42:08:EF:12:D2:38:B0:EE:E3:1F:C5...",
    "serialNumber"        : "58:B8:3F:43:98:56:79:35:90:E9:6C:41:30:F1:7D:0D",
    "subjectAlternatives" : [
      "myracloud.com",
      "www.myracloud.com"
    ],
    "intermediates": [
    {
      "objectType"   : "SslIntermediateVO",
      "subject"      : "C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=...",
      "algorithm"    : "RSA-SHA1",
      "validFrom"    : "2006-12-01T00:00:00+0000",
      "validTo"      : "2020-05-30T10:48:38+0000",
      "fingerprint"  : "59:99:9D:FD:06:B3:CB:1F:73:D6:6A:81:D5:51:B0...",
      "serialNumber" : "2E:79:83:2E:90:88:87:EA:8B:8E:F3:1A:6E:E6:7A:44",
      "issuer"       : "C=US, ST=UT, L=Salt Lake City, O=The USERTRUST Network, OU=..."
    },
    {
      "objectType"   : "SslIntermediateVO",
      "subject"      : "C=US, ST=UT, L=Salt Lake City, O=The USERTRUST Network, OU=...",
      "algorithm"    : "RSA-SHA1",
      "validFrom"    : "2005-06-07T08:09:10+0000",
      "validTo"      : "2020-05-30T10:48:38+0000",
      "fingerprint"  : "21:3F:AD:03:B1:C5:23:47:E9:A8:0F:29:9A:F0:89...",
      "serialNumber" : "46:EA:F0:96:05:4C:C5:E3:FA:65:EA:6E:9F:42:C6:64",
      "issuer"       : "C=SE, O=AddTrust AB, OU=AddTrust External TTP Network, CN=..."
    }
    ],
    "wildcard"           : false,
    "extendedValidation" : false,
    "subdomains" : [
      "www.myracloud.com"
    ]
  }
],
"page"     : 1,
"count"    : 1,
"pageSize" : 50
}
```

Listing 73: Example request listing SSL certificates

### 13.2.2 Create

To upload and assign a new certificate send a SslCertVO object without the attributes "id", "created", and "modified". All attributes are generated by the server and will be returned after that.

```
PUT /{language}/rapi/certificates/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "objectType"       : "SslCertVO",
    "cert"             : "(( certificate ))",
    "certRefreshForce" : false,
    "certToRefresh"    : 54321,
    "intermediates"    : [
        {
            cert: "(( intermediate certificate or chain of certificates (PEM) ))"
        }
    ],
    "key"              : "(( private key ))",
    "subdomains"       : [
        "www.myracloud.com",
        "www2.myracloud.com"
    ]
}
```

Listing 74: Example request of uploading a SSL certificate

A successful insert will return a ResultVO with the assigned subdomains and intermediates. Please note that the API will never return the certificate nor the private key.
After a successful upload your configuration will be queued and deployed to our system.

### 13.2.3 Update

When updating a certificate the API does not allow you to change the certificate or the key. If you would like to do so, you need to upload the certificate again. It is possible to change the intermediate chain as well as the assigned subdomains.

```
POST /{language}/rapi/certificates/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "id"            : 3,
    "modified"      : "2013-12-09T11:30:00+0100",
    "intermediates" : [
      {
        "cert" : "(( intermediate or chain of intermediates (PEM) ))"
      }
    ],
    "subdomains" : [
        "www3.myracloud.com"
    ]
}
```

Listing 75: Example request updating a SSL certificate

To see an example of a valid response you can take a look at the response example in the create section. After an update your configuration will be queued and deployed to our system.

### 13.2.4 Delete

It is not necessary to delete a certificate. When a certificate has not been assigned to any subdomain for more than one week it will be automatically removed.

## 13.3 Full version of a SslCertVO object

```
{
    "objectType"       : "SslCertVO",
    "cert"             : "(( certificate ))",
    "certRefreshForce" : false,
    "certToRefresh"    : 54321,
    "intermediates"    : [
        {
            cert: "(( intermediate certificate or chain of certificates (PEM) ))"
        }
    ],
    "key"              : "(( private key ))",
    "subdomains"       : [
        "www.myracloud.com",
        "www2.myracloud.com"
    ]
}
```

Listing 76: Full version of a SslCertVO object for uploading

```
{
    "objectType"  : "SslCertVO",
    "id"          : 12345,
    "modified"    : "2016-04-12T15:03:23+0200",
    "created"     : "2016-04-12T15:03:23+0200",
    "subject"     : "C=DE, O=Example certificate, ST=Bavaria, L=Munich, CN=myracloud.com",
    "algorithm"   : "RSA-SHA256",
    "validFrom"   : "2016-04-12T09:01:26+0200",
    "validTo"     : "2019-04-17T23:59:59+0200",
    "fingerprint" : "43:51:A1:B5:FC:8B:B7:0a:3a:a9:b1:0f:66:73:a8:37:27:47:DD:EC:69:CA:31:71...",
    "serialNumber": "80:3D:33:8A:0E:FC:18:C4",
    "subjectAlternatives": [
        "www.myracloud.com",
        "www2.myracloud.com"
    ],
    "intermediates": [
      {
        "objectType"   : "SslIntermediateVO",
        "subject"      : "C=SE, O=AddTrust AB, OU=AddTrust..., CN=AddTrust External CA Root",
        "algorithm"    : "RSA-SHA256",
        "validFrom"    : "2014-07-09T17:14:50+0200",
        "validTo"      : "2021-07-09T17:09:04+0200",
        "fingerprint"  : "49:F7:28:C0:0C:FC:F5:B4:43:D6:6D:E9:D3:81:1F:64:F8:29:B1:1D:8D:94...",
        "serialNumber" : "72:7B:21:6",
        "issuer"       : "C=SE, O=AddTrust AB, OU=AddTrust..., CN=AddTrust External CA Root2"
        }
    ],
    "wildcard": false,
    "extendedValidation": false,
    "subdomains": [
        "www.myracloud.com",
        "www2.myracloud.com"
    ]
}
```

Listing 77: Full version of a SslCertVO object

Table 26: Attributes for SslCertVO

| Attribute | Type | Required | Short description |
|---|---|---|---|
| id | Number | UD | Id of the object |
| modified | Date | UD | Date of last modification |
| created | Date | - | Date of creation |
| cert | String | C | Certificate |
| certRefreshForced | Boolean | C | True to force certificate update |
| certToRefresh | Number | C | Id of the certificate to refresh |
| key | String | C | Unencrypted private key |
| subject | String | - | Subject of the certificate |
| algorithm | String | - | Signature algorithm of the certificate |
| validFrom | Date | - | Date and time the certificate is valid from |
| validTo | Date | - | Date and time the certificate is valid to |
| fingerprint | String | - | RSA 256 fingerprint of the certificate |
| serialNumber | String | - | Serial number of the certificate |
| subjectAlternatives | Array | - | Sub domain(s) the certificate is valid for |
| intermediates | Array | - | An array of intermediate certificate(s) |
| wildcard | Boolean | - | True if the certificate contains a wildcard domain |
| extendedValidation | Boolean | - | True if the certificate has extended validation |
| subDomains | Array | C | List of subdomains where to assign the certificate |

**C** = Create, **U** = Update, **D** = Delete

**id**

Id is an unique identifier for an object. This value is always a number type and cannot be set while inserting a new object. To update or delete a cache setting it is necessary to add this attribute to your object.

**modified**

Identifies the version of the object. To ensure that you are updating the most recent version and not overwriting other changes, you always have to add the modified timestamp for updates and deletes. This value is always a date type with an ISO8601 format.

**created**

Created is a date type attribute with an ISO8601 format. It will be created by the server after creating a new cache setting object. This value is only informational so it is not necessary to add this attribute to every API call.

**cert**
Cert contains the certificate.

**certRefreshForced**
Every time a certificate is refreshed with another non-matching certificate the operation is interrupted with an error. Setting certRefreshForced will ignore such errors and refresh the certificate anyway. Please use it only, if you are sure you can ignore an error when refreshing a certificate.

**certToRefresh**
This property allows you to update an already existing certificate with a new one without changing IP addresses.

**subject**
Shows the subject of the uploaded certificate.

**algorithm**
Contains the signature algorithm.

**validFrom**
Time when the certificate starts to be valid. This property is a date type with an ISO8601 format.

**validTo**
Time when the certificate expires. This property is a date type with an ISO8601 format.

**fingerprint**
Fingerprint of the certificate.

**serialNumber**
Serial number of the certificate.

**subjectAlternatives**
Contains a list of subdomains which can be validated using this certificate. This list also contains the CN of the subject.

**intermediates**
Contains a list of intermediate certificates to be used in order to generate a chain of trust. The intermediates are filtered and sorted based on subject / issuer relationship. Uploading a partial or a completely different chain will result in an empty list. When returned, the list contains a set of SslIntermediateVO objects.

**wildcard**
This property shows whether the certificate is valid for multiple subdomains of a domain. The certificate needs to have a *.domain.tld subject to return true.

**extendedValidation**
True if the browser handles the certificate as extended validation. We use the OIDs from Google Chrome™ to measure the extended validation level.

**key**

The unencrypted private key that matches your certificate.

**subdomains**

A list of subdomains assigned to this certificate.

# 14 Statistics

## 14.1 Overview

The statistics API lets you fetch statistical data of your domains like requests, traffic, performance, or health.

## 14.2 Usage

All static requests use URL specific parameters; a list of these parameters can be found in the table below.

Table 27: URL parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |

### 14.2.1 Query

To fetch statistical data of your domains, you can use the following REST request:

```
POST /{language}/rapi/statistic/query HTTP/1.1
Host: api.myracloud.com
Date: 2014-06-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "query" : {
        "aggregationInterval"    : "hour",
        "dataSources"            : {
            "myDataSource" : {
                "source" : "bytes_cache_hits",
                "type"   : "stats"
            }
        },
        "startDate"              : "2015-01-27T00:00:00+0100",
        "endDate"                : "2015-01-27T12:00:59+0100",
        "fqdn"                   : ["ALL:example.com"],
        "type"                   : "fqdn"
    }
}
```

Listing 78: Example request for statistics

A successful call will return the original query and the actual result as an object with the key "result". For each data source given in the query a resultset will be returned with the same name.

```
{
    "query" : {
        "aggregationInterval"    : "hour",
        "dataSources"            : {
            "myDataSource" : {
                "source" : "bytes_cache_hits",
                "type"   : "stats"
            }
        },
        "startDate"              : "2015-01-27T00:00:00+0100",
        "endDate"                : "2015-01-27T12:00:59+0100",
        "fqdn"                   : ["ALL:example.com"],
        "type"                   : "fqdn"
    },
    "result": {
        "myDataSource": {
            "avg": 5866621.8906448,
            "max": 65366760,
            "min": 0,
            "sum": 18561991662
        },
        "requests_histogram": {
            "1422399600000": {
                "avg": 334.70860927152,
                "max": 1609,
                "min": 1,
                "sum": 101082
            }
        }
    }
}
```

Listing 79: Example response for a successful statistic API call

## 14.3 StatisticVO

```
{
    "query"         : {},
    "result"        : {}
}
```

Listing 80: StatisticVO

Table 28: Attributes for StatisticVO

| Attribute | Type | Description |
|---|---|---|
| query | Object | Contains the StatisticQueryVO |
| result | Object | Returns the detailed custom data for the requested domain |

**query**

Information about the structure of $query can be found in section 14.3.1. The given query object is returned unchanged in the response of the API call.

**result**

The result object is ignored for requests and should be empty. It will contain the result of the query specified in $query.

### 14.3.1 StatisticQueryVO

```
{
    "aggregationInterval"    : "hour",
    "dataSources"            : {
        "name" : {
            "source"   : "bytes_cached",
            "type"     : "stats"
        }
    },
    "startDate"              : "2015-01-27T00:00:00+0100",
    "endDate"                : "2015-01-27T12:00:59+0100",
    "fqdn"                   : [ "www.example.com" ],
    "type"                   : "fqdn"
}
```

Listing 81: StatisticQueryVO

Table 29: Attributes for StatisticQueryVO

| Attribute | Type | Description |
|---|---|---|
| aggregationInterval | String | The interval for aggregating the data points |
| dataSources | Object | List of data sources and output type |
| startDate | DateTime | Start of the aggregation interval |
| endDate | DateTime | End of the aggregation interval |
| fqdn | Array | A list of FQDN |
| type | String | Mode for selecting domains which should be used |

**aggregationInterval**

The statistics can be requested in various aggregation intervals. The requested data will be split into buckets of the given date interval. This applies only to data requested as histogram. The supported intervals are: "5m", "hour", "day", and "week".

**dataSources**

Information about the structure can be found in section 14.3.2.

**startDate**

Included start of the requested period. It should be in ISO8601 format. For example: "2015-03-13T12:00:00+01:00".

**endDate**

Included end of the requested period. It should be in ISO8601 format.
For example: "2015-04-21T12:00:00+01:00".

**fqdn**

Contains a list of FQDN for which statistics should be generated. Note that you can also use

'ALL:fqdn.de' as domain name to include data for all subdomains. This value is only used if the $type is set to 'fqdn'.

**type**

Mode for selecting domains which should be used.

**fqdn:** Process the FQDN list given in $fqdn

### 14.3.2 Data sources

With the statistics API, you can query various information about request types and how they were handled.

```
{
    "name"    : {
        "source" : "bytes_cached",
        "type"   : "stats"
    }
}
```

Listing 82: Data sources

Table 30: Attributes for Data sources

| Attribute | Type | Description |
|---|---|---|
| name | String | Arbitrary name of the dataset |
| source | String | Category of the requested data source |
| type | String | Type of the data aggregation |

**name**

The given name is used to name the corresponding result set in the API response. The name may only contain [a-zA-Z0-9_] characters.

**source**

See the tables Request data sources and Traffic data sources for a list of possible data source names to use.

**type**

The statistic data can be requested in two different aggregation types.

**stats:** Data as object containing min/max/avg/sum values.

**histogram:** Response will consist of multiple objects containing the value for every aggregation bucket.

**Requests**

Myra distinguishes incoming requests as SSL and non-SSL depending on the protocol used by the client initiating the request. You can also retrieve information about whether the response was sent from the Myra cache or from origin system.

Table 31: Request data sources

| **Data source** | **S** | **N** | **C** | **U** | **Description** |
|---|---|---|---|---|---|
| requests | X | X | X | X | Total amount of requests handled by Myra |
| requests_ssl | X |  | X | X | Amount of requests received via SSL |
| requests_nonssl |  | X | X | X | Amount of requests received not via SSL |
| requests_cached | X | X | X |  | Total amount of requests delivered from cache |
| requests_cached_ssl | X |  | X |  | Amount of requests via SSL delivered from cache |
| requests_cached_nonssl |  | X | X |  | Amount of requests via non-SSL delivered from cache |
| requests_uncached | X | X |  | X | Total amount of requests passed to the origin |
| requests_uncached_ssl | X |  |  | X | Total amount of requests via SSL passed to the origin |
| requests_uncached_nonssl |  | X |  | X | Total amount of requests via non-SSL passed to the origin |
| requests_cache_hits | X | X | X |  | Ratio of total cached requests to uncached requests in percent |
| requests_cache_hits_ssl | X |  | X |  | Ratio of SSL cached requests to SSL uncached requests in percent |
| requests_cache_hits_nonssl |  | X | X |  | Ratio of non-SSL cached requests to non-SSL uncached requests in percent |

**S** = SSL, **N** = non-SSL, **C** = Cached, **U** = Uncached

**Data source**
Name of the data source.

**SSL**
Includes amount of requests sent and answered via SSL connection.

**non-SSL**
Includes amount of requests sent and answered via non-SSL connection.

**Cached**
Includes amount of requests answered from the Myra cache without querying the origin server.

**Uncached**

Includes amount of requests answered by Myra after passing the request to the origin server and sending the corresponding response.

### Traffic

Myra distinguishes transferred data as SSL and non-SSL traffic depending on the protocol used by the client initiating the request. You can also retrieve information about whether the response was sent from the Myra cache or was fetched from the origin system.

Table 32: Traffic data sources

| Datasource | S | N | C | U | Description |
|---|---|---|---|---|---|
| bytes | X | X | X | X | Total amount of outgoing data in bytes |
| bytes_ssl | X |  | X | X | Amount of data sent via SSL |
| bytes_nonssl |  | X | X | X | Amount of data sent via non-SSL |
| bytes_cached | X | X | X |  | Total amount of data delivered from the cache |
| bytes_cached_ssl | X |  | X |  | Amount of data via SSL delivered from the cache |
| bytes_cached_nonssl |  | X | X |  | Amount of data via non-SSL delivered from the cache |
| bytes_uncached | X | X |  | X | Total amount of data passed through from the origin |
| bytes_uncached_ssl | X |  |  | X | Amount of data passed via SSL from the origin |
| bytes_uncached_nonssl |  | X |  | X | Amount of data passed via non-SSL from the origin |
| bytes_cache_hits | X | X | X |  | Ratio of total bytes delivered from the cache in percent |
| bytes_cache_hits_ssl | X |  | X |  | Ratio of bytes delivered via SSL from the cache in percent |
| bytes_cache_hits_nonssl |  | X | X |  | Ratio of bytes delivered via non-SSL from the cache in percent |

**S** = SSL, **N** = non-SSL, **C** = Cached, **U** = Uncached

**Data source**
Name of the data source.

**SSL**
Includes bytes transferred via SSL connection.

**non-SSL**
Includes bytes transferred via non-SSL connection.

**Cached**
Includes bytes of all responses transferred from Myra cache without querying the origin server.

**Uncached**
Includes bytes of all responses transferred from Myra without caching.

### Other

In addition to traffic and request statistics, Myra allows you to view detailled information about geographic distribution of your visitors, the performance of your origin server and the HTTP response codes of your application.

Table 33: Other data sources

| Datasource | S | N | C | U | Description |
|---|---|---|---|---|---|
| upstream_performance | X | X |  | X | Average upstream response time |
| response_codes | X | X | X | X | HTTP response codes of total requests |
| country_codes | X | X | X | X | Total requests by country |

**S** = SSL, **N** = non-SSL, **C** = Cached, **U** = Uncached

### 14.3.3 Result

```
{
    "bytes_cached_stats": {
        "avg" : 5866621.8906448,
        "max" : 65366760,
        "min" : 0,
        "sum" : 18561991662
    }

    "requests_histogram": {
        "1422399600000": {
            "avg" : 334.70860927152,
            "max" : 1609,
            "min" : 1,
            "sum" : 101082
        }
    }
}
```

Listing 83: Result

Table 34: Attributes for StatisticQueryVO

| Attribute | Type | Description |
|---|---|---|
| name | String | Name of the result set according to chosen name in the request |
| avg | Float | The average value of the requested source |
| max | Float | The maximal value of the requested source |
| min | Float | The minimal value of the requested source |
| sum | Float | The sum of the requested source |

**Info**

The structure of the returned data depends on the requested type. Data sources requested with type "stats" will be returned as an object with min/max/avg/sum keys as seen in "bytes_cached_stats" above. Responses for type "histogram" will contain multiple objects with the key "value", one object for every aggregation bucket. Please note that the **timestamp** used as key is in **milliseconds**.

# 15 Permissions

## 15.1 Overview

The permission endpoint lets you manage the permissions of a domain.

## 15.2 Usage

Most of the permission endpoint requests use URL specific parameters. A list of these parameters can be found in the table below.

Table 35: URL parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| domain | String | Yes | The domain you are working on. |
| page | Number | Yes | Number of the page you want to retrieve (1-based). |

### 15.2.1 Clone

Be aware that cloning permission can revert your own permissions from the target domains.

Clone permissions allows you to clone all permissions from a domain to multiple other domains. This transfer process remove all users / permission from the target domain and copies all users and their respective permissions. The domain given in the URL will be the source of the clone process.

Cloning permission requires that you are allowed to read the permission from the source domains and edit permissions on the target domains, otherwise you will get a HTTP 403 permission denied error. The clone process is transaction safe.

```
PUT /{language}/rapi/permissions/clone/{domain} HTTP/1.1
Host: api.myracloud.com
Date: 2014-05-02T07:17+0200
Authorization: MYRA {apiKey}:{signature}

{
    "targetDomainList" : [
        "example.com", "target-two.com"
    ]
}
```

Listing 84: Example request for clone permissions

After the successfull clone process the new permissions are considert instantly.

# 16 IP address ranges of Myra

## 16.1 Overview

You can use the Myra IP address ranges as a whitelist on your firewall, to configure Real-IP in Nginx or mod_remoteip in Apache, or to restrict requests to only the Myra IP address ranges.

## 16.2 Usage

All IP address ranges (networks) requests use URL specific parameters; a list of these parameters can be found in the table below.

Table 36: URL parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| language | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |

### 16.2.1 Listing

The listing operation returns a list of IP address ranges. The list contains Myra's IPv4 and IPv6 address ranges which are used to connect to your system.
The REST request must have the following format:

```
GET /{language}/rapi/networks HTTP/1.1
Host: api.myracloud.com
Date: 2019-01-21T17:31+0200
Authorization: MYRA {apiKey}:{signature}
```

Listing 85: Example request

A successful call will return a response with a HTTP status code of 200 and the following body:

```
{
"error"    : false,
"list"     : [
    {
        "objectType"  : "IpRangeVO",
        "id"          : 37,
        "modified"    : "2018-10-15T21:49:41+0200",
        "created"     : "2018-06-25T13:29:09+0200",
        "comment"     : "",
        "network"     : "103.51.165.0/24",
        "enabled"     : true,
        "validFrom"   : "2018-06-25T02:00:00+0200"
    },
    ...
],
"page"     : 1,
"count"    : 29,
"pageSize" : 29
}
```

Listing 86: Example response

### 16.2.2 Create

Creating new IP address ranges is not possible.

### 16.2.3 Update

Updating IP address ranges is not possible.

### 16.2.4 Delete

Deleting IP address ranges is not possible.

## 16.3 Full version of a IP address ranges object

```
{
        "objectType"  : "IpRangeVO",
        "id"          : 37,
        "modified"    : "2018-10-15T21:49:41+0200",
        "created"     : "2018-06-25T13:29:09+0200",
        "comment"     : "",
        "network"     : "103.51.165.0/24",
        "enabled"     : true,
        "validFrom"   : "2018-06-25T02:00:00+0200"
}
```

Listing 87: Full version of a IP address ranges object

Table 37: Attributes for Domain

| Attribute | Type | Required | Short description |
|---|---|---|---|
| objectType | String | - | The object type |
| id | Number | - | Id of the object |
| modified | Date | - | Date of last modification |
| created | Date | - | Date of creation |
| comment | String | - | A comment |
| network | String | - | IP address range in CIDR notation |
| enabled | Bool | - | Whether the IP address range is used in production or not |
| validFrom | Date | - | From which date on the IP address range is valid |

**C** = Create, **U** = Update, **D** = Delete

**objectType**
ObjectType describes the object.

**id**
Id is an unique identifier for an object. This value is always a number type.

**modified**
Identifies the version of the object. This value is always a date type with an ISO8601 format.

**created**
Created is a date type with an ISO8601 format.

**comment**
Comment contains additional information about this object.

**network**
Network contains the IP address range in CIDR notation.

**enabled**

Enabled shows whether this entry is valid or not.

**validFrom**

ValidFrom gives information about from which date on the IP address range is valid.

# 17 Development

## 17.1 MYRA_NOCACHE

While a developer is working on a new feature or testing Myra's behaviour, it is sometimes necessary to prevent the caching layer from doing its work. To prevent Myra from caching your request(s) you have to set a cookie on your domain with MYRA_NOCACHE=1. Now all your requests targeting the cookie domain are responded from your origin server.

# 18 Cache (Deprecated)

## 18.1 Overview

To caching interface is implemented as a REST service, the endpoint for this API is the myracloud.com website. Currently only clearCache is available.

## 18.2 Security

The authentication is done using an API key on a per user basis.

## 18.3 Usage

The REST request must have the following format:

```
GET /{lang}/api/{command}/{apiKey}/{properties}/ HTTP/1.1
Host: app.myracloud.com
```

Listing 88: Example

Table 38: Parameters

| Parameter | Type | Required | Description |
|---|---|---|---|
| lang | String | Yes | Sets the preferred language of the error messages. Valid values are currently only "en" for English or "de" for German. |
| command | String | Yes | The command which should be executed. Accepted values: clear. |
| apiKey | String | Yes | An API key for user authentication. |
| properties | String | Depends | Various data, differs for each command. |

### 18.4 Command: clear

The {properties} parameter (www.example.com) is used to transport the domain. This domain is used to purge the cache for the given domain only.

```
GET /{lang}/api/clear/{apiKey}/www.example.com/ HTTP/1.1
Host: app.myracloud.com
```

Listing 89: Example

Table 39: Response codes

| Return code | Description |
|---|---|
| 200 OK | OK |
| 403 Authentication failed | Domain is not registered for this API key |
| 404 Unknown Command | Currently not implemented command |

# Document history

| Version | Date | Author | Description |
|---|---|---|---|
| 0.1a | 2014-02-11 | Bruno Lorenz | Initial document |
| 0.2 | 2014-06-27 | Vahe Sahakyan | Review and updates |
| 0.3 | 2014-08-27 | Bruno Lorenz | Added Maintenance |
| 0.4 | 2014-10-16 | Bruno Lorenz | Appended cache clear API |
| 0.5 | 2014-11-17 | Bruno Lorenz | Updated cache clear API |
| 0.6 | 2014-11-28 | Anja Förster | Added IP filter API |
| 0.8 | 2015-01-28 | Anja Förster | Added Statistics API |
| 0.9 | 2015-05-21 | Bruno Lorenz | Append new attribute to DnsRecords |
| 1.0 | 2015-07-06 | Bruno Lorenz | Append domain API |
| 1.1 | 2015-07-06 | Anja Förster | Append (sub)domain settings API |
| 1.2 | 2015-11-04 | Bruno Lorenz | Append cache clear pattern matching |
| 1.4 | 2016-03-08 | Bruno Lorenz | Append matchingType to redirect API |
| 1.5 | 2016-08-30 | Bruno Lorenz | Append SSL and permission chapter |
| 1.7 | 2017-05-16 | Richard Prillwitz | Changes on Redirect/expertMode |
| 1.8 | 2018-05-17 | Carlo Di Fulco | Loadbalance |
| 1.9 | 2019-01-17 | Hakan K.Yilmaz | Re-org and signature examples |

Please feel free to contact us, if you have any questions about our API, best practices, or feedback.

Please contact us:

**Torsten Zinke**
Senior Key Account Manager & Authorized Officer
Information Security Officer
+49 89 / 41 41 41 - 343
torsten.zinke@myrasecurity.com

**Hakan K.Yilmaz**
VP Platform Engineering
+49 89 / 41 41 41 - 323
hakan.kyilmaz@myrasecurity.com

Our **support hotline** is available via: +49 89 / 41 41 41 - 346

and **email ticketing** via: support@myrasecurity.com

www.myrasecurity.com
Myra Security GmbH | Landsberger Str. 187 | 80687 München