

Full API documentation

- Overview
 - Basics
 - Security
 - Returned data format
 - Failure response format
- API documentation
 - General
 - Health check
 - Schedules
 - Schedules - get schedules
 - Devices
 - Devices - get device by ID
 - Devices - get device by address
 - Devices - get devices
 - Devices - create new device
 - Devices - update device
 - Devices - delete device
 - Backups
 - Backups - get device backups
 - Backups - get device latest backup
 - Backups - get latest backups
 - Running tasks
 - Run job - discover device
 - Run job - discover un-discovered devices
 - Run job - backup device

Overview

Basics

The Unimus API is a JSON based RESTfull API. It utilizes common HTTP methods such as **GET**, **POST**, **PATCH** and **DELETE** in order to identify action you want to perform.

When submitting a request that contains a body you must include a header for **Content-type** that specifies **application/json**.

API Version 1 is supported by Unimus 1.6.x and older. API version 1 is no longer supported in latest Unimus releases.

Security

Every request has to include **Authorization** header following Bearer scheme as shown below.

```
Authorization: Bearer <token>
```

Log into your local Unimus instance and navigate to **User management > API tokens** section to create new token.

Returned data format

All responses are returned as JSON. If a response can ever contain a single item it will be formatted as follow:

```
{
  "data": {
    "some key": "some value",
    "another key": "another value"
  }
}
```

Multi-item responses are paginated automatically. The maximum amount of items that can be returned in a single query is 50, default is 20. You may specify the number of items to be returned in response by appending a query parameter **size**.

You may specify the page you wish to view by appending a parameter **page**, default is 0. Multi-item responses are formatted as follow:

```
{
  "data": [
    {
      "some key": "some value",
      "another key": "another value"
    },
    {
      "some key": "some value",
      "another key": "another value"
    }
  ],
  "paginator": {
    "totalCount": 12,
    "totalPages": 6,
    "page": 0,
    "size": 2
  }
}
```

All successful response are sent with an HTTP response code **2xx**.

Failure response format

If an API response fails, it will be sent with a **non-2xx** response code and JSON in the following format:

```
{
  "timestamp": 1511904038642
  "code": 400
  "error": "Bad Request"
  "message": "This is why it fails"
}
```

The HTTP status codes currently returned by the API are **400** if request syntax is wrong, **401** if request is unauthorized, **404** if requesting with item ID and item is not found or request path is not supported by the API, **405** if the method is not supported, **415** if media type is not supported, **503** if request can not be served. Any internal application errors will respond with a **500** code and we will be grateful if u report them to us.

API documentation

General

Health check

Get Unimus health status.

Method: **GET**

```
http://example.unimus/api/v1/health
```

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" "http://<example.unimus>/api/v1/health"
```

Success 200

Field	Type	Description
status	String	OK - Unimus is ready to handle all requests. LICENSING_UNREACHABLE - License server is unreachable. All request will proceed except add, remove and update device. ERROR - Unimus require user interaction to solve the problem. All requests will be refused.

Success response

HTTP/1.1 200 OK

```
{
  "data": {
    "status": "OK"
  }
}
```

Schedules

Schedules - get schedule

Get an individual schedule.

Method: **GET**

```
http://example.unimus/api/v1/schedules/<scheduleId>
```

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" "http://<example.unimus>/api/v1/schedules/<scheduleId>"
```

Parameter

Field	Type	Description
scheduleId	Number	The ID of the schedule

Success 200

Field	Type	Description
id	Number	The ID of the schedule
createTime	Number	Schedule creation time in seconds
periodicity	String	Schedule periodicity. Possible values: MONTHLY , WEEKLY , DAILY , HOURLY
dom	Number	Day of month
dow	Number	Day of week
hour	Number	Hour of day
min	Number	Minute of hour
name	String	Schedule pretty name

Success response

HTTP/1.1 200 OK

```
{
  "data": {
    "id": 7,
    "createTime": 1511863250,
    "periodicity": "MONTHLY",
    "dom": 1,
    "dow": 1,
    "hour": 0,
    "min": 6,
    "name": "Every month, on day 1, at 00:06."
  }
}
```

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	404
error	String	HTTP code message
message	String	Error message

Error response
HTTP/1.1 404 NOT FOUND <pre>{ "timestamp": 1511862182524, "code": 404, "error": "Not Found", "message": "Schedule with id 11 not found" }</pre>

Schedules - get schedules

Get a list of all schedules in Unimus.

Method: **GET**

http://example.unimus/api/v1/schedules?page=:pageIndex&size=:pageSize

CURL

<code>curl -H "Accept: application/json" -H "Authorization: Bearer <token>" "http://<example.unimus>/api/v1/schedules?page=0&size=20"</code>
--

Parameter

Name	Type	Description
pageIndex	Number	Page index (OPTIONAL)
pageSize	Number	Page size (OPTIONAL)

Success 200

Name	Type	Description
id	Number	The ID of the schedule
createTime	Number	Schedule creation time in seconds
periodicity	String	Schedule periodicity. Possible values: MONTHLY , WEEKLY , DAILY , HOURLY
dom	Number	Day of month
dow	Number	Day of week
hour	Number	Hour of day
min	Number	Minute of hour
name	String	Schedule pretty name

Success response

HTTP/1.1 200 OK

```
{
  "data": [
    {
      "id": 5,
      "createTime": 1511863239,
      "periodicity": "MONTHLY",
      "dom": 5,
      "dow": 1,
      "hour": 0,
      "min": 0,
      "name": "Every month, on day 5, at 00:00."
    },
    {
      "id": 7,
      "createTime": 1511863250,
      "periodicity": "MONTHLY",
      "dom": 1,
      "dow": 1,
      "hour": 0,
      "min": 6,
      "name": "Every month, on day 1, at 00:06."
    }
  ],
  "paginator": {
    "totalCount": 7,
    "totalPages": 4,
    "page": 1,
    "size": 2
  }
}
```

Devices

Devices - get device by ID

Get an individual device by ID.

Method: **GET**

```
http://example.unimus/api/v1/devices/:deviceId?attr=:attributes
```

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" "http://<example.unimus>/api/v1/devices/<deviceId>?attr=s,c"
```

Parameter

Name	Type	Description
deviceId	Number	The ID of the device
attributes	String	Comma separated graph attribute nodes. (OPTIONAL) Possible values for device schedule: schedule, sch, s for device credential: credential, cred, c

Success 200

Name	Type	Description
id	Number	The ID of the device
createTime	Number	Device creation time in seconds
address	String	Hostname, IPv4 or IPv6
description	String	Device description
port	Number	Device port
connector	String	Connector type, possible values: SSH, TELNET
vendor	String	Vendor
type	String	Type
model	String	Model
schedule	Object	Device schedule
credential	Object	Credential used to login to the device

Success response

HTTP/1.1 200 OK

```
{
  "data": {
    "id": 1,
    "createTime": 1511339013,
    "address": "198.18.0.29",
    "description": "Back bone device",
    "port": 22,
    "connector": "SSH",
    "vendor": "MikroTik",
    "type": "RouterOS",
    "model": "Simulated-RouterOS",
    "schedule": null,
    "credential": {
      "id": 1,
      "username": "test",
      "password": "test",
      "sshKey": null
    }
  }
}
```

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	400
error	String	HTTP code message
message	String	Error message

Error response
<pre> HTTP/1.1 400 BAD REQUEST { "timestamp": 1511869830038, "code": 400, "error": "Bad Request", "message": "Argument deviceId type mismatch" }</pre>

Devices - get device by address

Get an individual device by address.

Method: **GET**

```
http://example.unimus/api/v1/devices/findByAddress/:address?attr=:attributes
```

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" "http://<example.unimus>/api/v1/devices/findByAddress/<address>?page=0&size=20&attr=s,c"
```

Parameter

Name	Type	Description
address	String	The IPv4, IPv6 or hostname of the device
attributes	String	Comma separated graph attribute nodes. (OPTIONAL) Possible values for device schedule: schedule, sch, s for device credential: credential, cred, c

Success 200

Name	Type	Description
id	Number	The ID of the device
createTime	Number	Device creation time in seconds
address	String	Hostname, IPv4 or IPv6
description	String	Device description
port	Number	Device port
connector	String	Connector type, possible values: SSH, TELNET

vendor	String	Vendor
type	String	Type
model	String	Model
schedule	Object	Device schedule
credential	Object	Credential used to login to the device

Success response
HTTP/1.1 200 OK <pre>{ "data": { "id": 1, "createTime": 1511339473, "address": "198.18.0.39", "description": "Main switch", "port": 22, "connector": "SSH", "vendor": "MikroTik", "type": "RouterOS", "model": "Simulated-RouterOS", "schedule": null, "credential": { "id": 4, "username": "secret user", "password": "top secret", "sshKey": null } } }</pre>

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	404
error	String	HTTP code message
message	String	Error message

Error response
HTTP/1.1 404 BAD REQUEST <pre>{ "timestamp": 1511862182524, "code": 404, "error": "Not Found", "message": "Device with address some_address not found" }</pre>

Devices - get devices

Get a list of devices.

Method: **GET**

```
http://example.unimus/api/v1/devices?page=:pageIndex&size=:pageSize&attr=:attributes
```

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" "http://<example.unimus>/api/v1/devices?page=0&size=20&attr=S,C"
```

Parameter

Name	Type	Description
pageIndex	Number	Page index (OPTIONAL)
pageSize	Number	Page size (OPTIONAL)
attributes	String	Comma separated graph attribute nodes.(OPTIONAL) Possible values for device schedule: schedule, sch, s for device credential: credential, cred, c

Success 200

Name	Type	Description
id	Number	The ID of the device
createTime	Number	Device creation time in seconds
address	String	Hostname, IPv4 or IPv6
description	String	Device description
port	Number	Device port
connector	String	Connector type, possible values: SSH, TELNET
vendor	String	Vendor
type	String	Type
model	String	Model
schedule	Object	Device schedule
credential	Object	Credential used to login to the device

Success response

HTTP/1.1 200 OK

```
{
  "data": [
    {
      "id": 2,
      "createTime": 1511339013,
      "address": "198.18.0.2777",
      "description": null,
      "port": 22,
      "connector": null,
      "credential": null,
      "vendor": "MikroTik",
      "type": "RouterOS",
      "model": "Simulated-RouterOS",
      "schedule": {
        "id": 1,
        "createTime": 1511339012,
        "periodicity": "DAILY",
        "dom": 1,
        "dow": 1,
        "hour": 3,
        "min": 0,
        "name": "Every day at 03:00."
      }
    }
  ],
  "paginator": {
    "totalCount": 29,
    "totalPages": 29,
    "page": 1,
    "size": 1
  }
}
```

Devices - create new device

Create a new device.

Method: **POST**

http://example.unimus/api/v1/devices

CURL

curl -H "Accept: application/json" -H "Content-type: application/json" -H "Authorization: Bearer <token>" -d '{"address":"<address>","description":"<description>"}' "http://<example.unimus>/api/v1/devices"

Parameter

Name	Type	Description
------	------	-------------

address	String	Device address
description	String	Description of the new device
scheduleId	Number	Schedule id If null then device will be backed up according to system default schedule

Success 201

Name	Type	Description
id	Number	The ID of the device
createTime	Number	Device creation time in seconds
address	String	Hostname, IPv4 or IPv6
description	String	Device description
schedule	Object	Device schedule
port	Number	null
connector	String	null
vendor	String	null
type	String	null
model	String	null
credential	Object	null

Success response

HTTP/1.1 201 CREATED

```
{
  "id": 94,
  "createTime": 1511876477,
  "address": "10.10.10.10",
  "description": null,
  "port": null,
  "connector": null,
  "credential": null,
  "vendor": null,
  "type": null,
  "model": null,
  "schedule": {
    "id": 1,
    "createTime": 1511339012,
    "periodicity": "DAILY",
    "dom": 1,
    "dow": 1,
    "hour": 3,
    "min": 0,
    "name": "Every day at 03:00."
  }
}
```

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	422
error	String	HTTP code message
message	String	Error message

Error response
<pre> HTTP/1.1 422 UNPROCESSABLE ENTITY { "timestamp": 1511876680885, "code": 422, "error": "Unprocessable Entity", "message": "Device already exist" }</pre>

Devices - update device

Update device address, description or schedule time.

Method: **PATCH**

<http://example.unimus/api/v1/devices/:deviceId>

CURL

```
curl -H "Accept: application/json" -H "Content-type: application/json" -H "Authorization: Bearer <token>" -X PATCH -d '{"address":"<address>","description":"<description>","scheduleId":"<scheduleId>"}' "http://<example.unimus>/api/v1/devices/<deviceId>"
```

Parameter

Name	Type	Description
deviceId	Number	The ID of the device
address	String	Device address
description	String	Device description
scheduleId	Number	Schedule id. If null then no schedule update occurs If -1 then device will be backed up according to system default schedule

Success 200

Name	Type	Description
id	Number	The ID of the device
createTime	Number	Device creation time in seconds
address	String	Hostname, IPv4 or IPv6

description	String	Device description
port	Number	Device port
connector	String	Connector type, possible values: SSH , TELNET
vendor	String	Vendor
type	String	Type
model	String	Model
schedule	Object	Device schedule
credential	Object	Credential used to login to the device

Success response
HTTP/1.1 200 OK <pre>{ "id": 96, "createTime": 1511879360, "address": "198.18.0.18", "description": "aaaa", "port": 22, "connector": "SSH", "vendor": "MikroTik", "type": "RouterOS", "model": "Simulated-RouterOS", "schedule": { "id": 1, "createTime": 1511339012, "periodicity": "DAILY", "dom": 1, "dow": 1, "hour": 3, "min": 0, "name": "Every day at 03:00." }, "credential": { "id": 1, "username": "test", "password": "test", "sshKey": null } }</pre>

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	404
error	String	HTTP code message
message	String	Error message

Error response
HTTP/1.1 404 NOT FOUND <pre>{ "timestamp": 1511880124270, "code": 404, "error": "Not Found", "message": "Device not found" }</pre>

Devices - delete device

Delete a single device from Unimus.

Method: **DELETE**

http://example.unimus/api/v1/devices/:deviceId

CURL

<pre>curl -H "Accept: application/json" -H "Authorization: Bearer <token>" -X DELETE "http://<example.unimus>/api/v1/devices/<deviceId>"</pre>
--

Parameter

Name	Type	Description
deviceId	Number	The ID of the device to delete

Success 200

Name	Type	Description
success	String	A message stating that the deletion was successful

Success response
HTTP/1.1 200 OK <pre>{ "data": { "success": "true" } }</pre>

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	404
error	String	HTTP code message

message	String	Error message
---------	--------	---------------

```
HTTP/1.1 404 NOT FOUND

{
  "timestamp": 1511884051726,
  "code": 404,
  "error": "Not Found",
  "message": "Device with id 102 not found"
}
```

Backups

Backups - get device backups

Get a list of all device backups. Backups are ordered descending by create time so the latest backup will always be at index 0 in the list. The list can be empty if the device has no backup yet.

Method: **GET**

```
http://example.unimus/api/v1/devices/:deviceId/backups?page=:pageIndex&size=:pageSize
```

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" "http://<example.unimus>/api/v1/devices/1/backups?page=0&size=20"
```

Parameter

Name	Type	Description
deviceId	Number	The ID of the device
pageIndex	Number	Page index (OPTIONAL)
pageSize	Number	Page size (OPTIONAL)

Succeeds 200

Name	Type	Description
id	Number	The ID of the backup
createTime	Number	Backup creation time
type	String	Backup type. Could be TEXT or BINARY
bytes	Array	Backup as byte array

Success response
HTTP/1.1 200 OK <pre>{ "data": [{ "id": 29, "createTime": 1511886237, "type": "TEXT", "bytes": "lyBTaW11bGF0ZWQgTWlrcm9UaWsgUm91dGVyT1Mgc3lzdGVtCiMgc29mdHdhcmUgaWQgPSBub25lCiMKL3NIY3Rpb24gYQpjb21tYW5klGEKL3NIY3Rpb24gYmIKY29tbWFuZCBiYgovc2VjdGlvb2Y2MKY29tbWVhZCBiY2MK" }], "paginator": { "totalCount": 5, "totalPages": 5, "page": 0, "size": 1 } }</pre>

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	404
error	String	HTTP code message
message	String	Error message

Error response
HTTP/1.1 404 NOT FOUND <pre>{ "timestamp": 1511886913012, "code": 404, "error": "Not Found", "message": "Device with id 964 not found" }</pre>

Backups - get device latest backup

Get device latest backup. Backup can be null if the device has no backups yet.

Method: **GET**

http://example.unimus/api/v1/devices/:deviceId/backups/latest

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" "http://<example.unimus>/api/v1/devices/<deviceId>/backups/latest"
```

Parameter

Name	Type	Description
deviceId	Number	The ID of the device

Success 200

Name	Type	Description
id	Number	The ID of the backup
createTime	Number	Backup creation time
type	String	Backup type. Could be TEXT or BINARY
bytes	Array	Backup as byte array

Success response

HTTP/1.1 200 OK

```
{
  "data": {
    "id": 29,
    "createTime": 1511886237,
    "type": "TEXT",
    "bytes": "lyBTaW11bGF0ZWQgTWlrcm9UaWsgUm91dGVyT1Mgc3lzdGVtCiMgc29mdHdhcmUgaWQgPSBub25lCiMKL3NIY3Rpb24gYQpb21tYW5kIGEKL3NIY3Rpb24gYmIKY29tbWFuZCBiYgovc2VjdGlvb2Y2MKY29tbWFuZCBjY2MK"
  }
}
```

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	404
error	String	HTTP code message
message	String	Error message

Error response

HTTP/1.1 404 NOT FOUND

```
{
  "timestamp": 1511889424942,
  "code": 404,
  "error": "Not Found",
  "message": "Device with id 12 not found"
}
```

Backups - get latest backups

Get a list of latest backup of devices with no particular order. Each backup is wrapped with device id and address.

Method: **GET**

```
http://example.unimus/api/v1/devices/backups/latest?page=:pageIndex&size=:pageSize&id=:deviceIds
```

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" "http://<example.unimus>/api/v1/devices/backups/latest?page=0&size=20&id=<deviceIds>"
```

Parameter

Name	Type	Description
pageIndex	Number	Page index (OPTIONAL)
pageSize	Number	Page size (OPTIONAL)
deviceIds	String	Comma separated device ids. If this parameter is not specified then backups of all devices will be returned according to pagination.

Success 200

Name	Type	Description
deviceId	Number	The ID of the device
address	String	Hostname, IPv4 or IPv6
backup	Object	Backup object

Success response

HTTP/1.1 200 OK

```
{
  "data": [
    {
      "deviceId": 108,
      "address": "198.18.0.18",
      "backup": {
        "id": 43,
        "createTime": 1511892002,
        "type": "TEXT",
        "bytes": "lyBTaW11bGF0ZWQgTWlrcm9UaWsgUm91dGVyT1Mgc3lzdGVtCiMgc29mdHdhcmUgaWQgPSBub25lCiMKL3NIY3Rpb24gYQpjb21tYW5kIGEKL3NIY3Rpb24gYmIKY29tbWFuZCBiYgovc2VjdGlvbiBjY2MKY29tbWFuZCBjY2MK"
      }
    }
  ],
  "paginator": {
    "totalCount": 12,
    "totalPages": 12,
    "page": 0,
    "size": 1
  }
}
```

Running tasks

Run job - discover device

Trigger discovery job with all devices in the system. If u want to narrow down devices included in job then specify their IDs in request parameter.

Method: **PATCH**

```
http://example.unimus/api/v1/jobs/discovery?id=:deviceIds
```

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" -X PATCH "http://<example.unimus>/api/v1/jobs/discovery?id=<deviceIds>"
```

Parameter

Name	Type	Description
deviceIds	String	Comma separated device ids to run discovery on

Success 202

Name	Type	Description
accepted	Number	Number of started discoveries
refused	Number	Number of refused discoveries. Discovery or backup currently running

Success response
<pre> HTTP/1.1 202 ACCEPTED { "data": { "accepted": 12, "refused": 0 } }</pre>

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	404
error	String	HTTP code message
message	String	Error message

Error response
<pre> HTTP/1.1 403 NOT FOUND { "timestamp": 1511897481896, "code": 404, "error": "Not Found", "message": "Devices not found [3]" }</pre>

Run job - discover un-discovered devices

Trigger discovery job on all undiscovered devices.

Method: **PATCH**

```
http://example.unimus/api/v1/jobs/discovery/undiscovered
```

CURL

```
curl -H "Accept: application/json" -H "Authorization: Bearer <token>" -X PATCH "http://<example.unimus>/api/v1/jobs/discovery/undiscovered"
```

Success 202

Name	Type	Description
------	------	-------------

accepted	Number	Number of started discoveries
refused	Number	Number of refused discoveries. Discovery or backup currently running

Success response
<pre> HTTP/1.1 202 ACCEPTED { "data": { "accepted": 12, "refused": 0 } }</pre>

Run job - backup device

Trigger backup job with all devices in the system. If u want to narrow down devices included in job then specify their IDs in request parameter.

Method: **PATCH**

http://example.unimus/api/v1/jobs/backup?id=:deviceIds

CURL

<code>curl -H "Accept: application/json" -H "Authorization: Bearer <token>" -X PATCH "http://<example.unimus>/api/v1/jobs/backup?id=<deviceIds>"</code>

Parameter

Name	Type	Description
deviceIds	String	Comma separated device ids to run backup on

Success 202

Name	Type	Description
accepted	Number	Number of started backups
refused	Number	Number of refused backups
undiscovered	Number	Unable to backup un-discovered devices

Success response
<pre> HTTP/1.1 202 ACCEPTED { "data": { "accepted": 0, "refused": 2, "undiscovered": 0 } }</pre>

Error 4xx

Name	Type	Description
timestamp	Number	Current timestamp
code	Number	404
error	String	HTTP code message
message	String	Error message

Error response
HTTP/1.1 403 NOT FOUND <pre>{ "timestamp": 1511899365179, "code": 404, "error": "Not Found", "message": "Devices not found [16, 23, 15]" }</pre>