Developer API Documentation

FreeboxOS Gateway APi allow access to Freebox Server settings and apps.

This API can be used to develop companion apps for Smartphone, or provide an alternative to FreeboxOS web app.

General Information

API Version

Api version will always use the following format : “major.minor” where major and minor are integers

Current API version is “15.0” Current major API version is: 15

When an API is marked as *unstable*, you can use it but it may change or disappear at any time!

When an API is not documented you should not use it!

Other API will be maintained for at least 1 Freebox release.

Api Changes

Api changes from version 1.1 to 2.0

Download Api Changes

- Added 2 new error_code:
  - invalid_address
  - port_conflict
- Added a new *‘cookies’* parameter when adding a download by url. This allow browser plugins to pass cookies along with url. This can be useful for session based authentication.
- Added new `DownloadStats` attributes:
  - conn_ready
  - nb_peer
  - blocklist_entries
  - blocklist_hits
  - dht_stats
- Deprecate path attribute for `DownloadFile`
- Add new attributes to `DownloadFile`
  - filepath
  - name
  - mimetype
- Added a blacklist API to control bittorrent peers blacklist entries

Download Configuration Api Changes

- Added new `DownloadConfiguration` attributes
  - main_port
  - dht_port

UPnP IGD Api Changes

- Added a host property to `UPnPRedir`

Connection api Changes (v2)

- Added a ipv6ll property to `ConnectionIpv6Configuration`
- Added a snr_10, attn_10 property to `XdslStats`

RRD Api Changes

- Added new entries for net database:
  - vpn_rate_down
  - vpn_rate_up
- Added new entries for temp database:
  - cpum
  - cpub
  - sw
  - hdd
  - fan_speed
- Deprecate entries for temp database:
  - temp1
  - temp2
  - temp3

System Api Changes

- Added uptime_val attribute to `SystemConfig`

Wifi Api Changes

- Completely rework Wifi API to be able to handle multiple Access Points.

New API (v2)

- Added an Incoming port configuration Api
- Added a VPN Client Api
- Added a VPN Server Api

Api changes from version 2.0 to 3.0

Connection api Changes (v3)

- Added a ginp, rtx_tx, rtx_c, rtx_uc property to `XdslStats`

New API (v3)

- Some tv, epg and pvr api have been added. Those api are undocumented and should be considered UNSTABLE (may be modified without further notice).

Api changes from version 3.0 to 4.0

Secure Access

- The Freebox OS API can now be reached over HTTPS. All applications MUST switch to https access. Unsecure access will be removed at some point.

Deprecated api (v4)

- The old upload api as been deprecated in favor of the WebSocket upload api. The v3 upload api will be removed in next firmware release. All new apps should only use websocket upload api. However tracking of uploads has not been changed.

Changed API

- The File System api now return more details error codes, and can now return 'access_denied' and 'disk_full' in case of IO errors
- `SystemConfig` has new 'disk_status', 'box_flavor' attributes
- `ConnectionStatus` now expose 'ipv4_port_range' for customers that don't have a 'full' IPv4
- Added new 'port_outside_range' error_code when attempting to use a port outside of assigned 'ipv4_port_range'
- Added 'remote_access_min_port' and 'remote_access_max_port' to `ConnectionConfiguration`
- Added 'min_port', 'max_port' for `IncomingPortConfig`, `VPNServerConfig`
- Added 'readonly' for `IncomingPortConfig`
- Added 'allow_remote_access' for `FtpConfig`
- Added 'mark_all_as_read' and 'delete_all' for Call api
- Added 'enabled_ipv6' and 'node_count_ipv6' for `DhtStats`
- Added 'preview_url' to `DownloadFile` for bittorrent downloads
- Added 'info_hash', 'piece_length' to `Download` for bittorrent downloads

New API (v4)

- Added `StorageConfig` api
- Added Download Pieces information

Api changes from version 4.0 to 5.0

Deprecated api (v5.0)

- The old upload api as been deprecated since v4 in favor of the WebSocket upload api. The v3 upload api will be removed in next firmware release. All new apps should only use websocket upload api. However tracking of uploads has not been changed.

Changed API (v5.0)

- Added 'wps_enabled', 'wps_uuid' to `WifiBssConfig` wps configuration
- Changed `WifiBss` logic to expose both 'bss_params' and 'shared_bss_params' and telling which one is currently used with the new field 'use_shared_params'. This replaces the 'use_default_config' from `WifiBssConfig` and 'is_main_bss' from `WifiBssStatus`

New API (v5.0)

- Added wifi `WifiCustomKey` api
- Added wifi `WifiWpsSession` api
- Added wifi `DHCPv6Config` api

Api changes from version 5.0 to 6.0

Changed API (v6.0)

- Added optional 'filename' parameter, to download "Add by url" api.

New API (v6.0)

- Added Home API
- Added Player API
- Added Notification API

Api changes from version 6.0 to 7.0

Changed API (v7.0)

- The api_version contains less information when called unauthenticated and remotely.

New API (v7.0)

- Added VM API for Freebox Delta.

Api changes from version 7.0 to 8.0

Deprecated API (v8.0)

- Parental control API is no longer usable. It has been replaced by the Profile API.

New API (v8.0)

- Profile API is simpler to use and replaces parental control API.

Api changes from version 8.0 to 8.1

New API (v8.1)

- Wifi has a new diagnostic API
- New language API

Api changes from version 8.1 to 8.2

A new way to discover a remote connection port change has been added. It is recommended to implement it as fallback mechanism, since the port can now change automatically once unreachable over IPv4.

Changed API (v8.2)

- New LAN browser device type (car): `LanHost.host_type`.
- File system task now have source and destination info: `FsTask.from`
- Fix file system rm issue preventing status to be correctly updated

Newly documented API (v8.2)

- RAID API is now documented. It is still considered unstable.
- VM API is now documented. It is still considered unstable.
- WebSocket event API has now additional documentation.

New API (v8.2)

- Added Language API to allow changing box language.

Api changes from version 8.2 to 8.3

New API (v8.3)

- Added File System Advice API to help user configure the storage attached to the Freebox.

Api changes from version 8.3 to 8.4

Changed API (v8.4)

- New Wifi api error code 'inval_wps_hidden_ssid' when trying to enable WPS with hidden SSID.

Api changes from version 8.4 to 8.5

Changed API (v8.5)

- Camera API does not require "camera" permission any more to list cameras. The permission is still needed to access camera records and live stream.
- Add camera lan id in camera API result to find the corresponding lan host in lan browser API.
- Add API to retrieve channel survey history

Api changes from version 8.5 to 9.0

Changed API (v9.0)

- WiFi API was extended to support 6Ghz band and 802.11ax (HE)

Api changes from version 9.0 to 9.1

Changed API (v9.1)

- New diagnostics API for network throughput slowness detection.

Api changes from version 9.1 to 10.0

Deprecated API (v10.0)

- The Connection API for xDSL/4G aggregation is no longer usable. It has been replaced by separate endpoints providing respectively LTE connection status and aggregation status.

Changed API (v10.0)

- The Connection API has been changed to not mix aggregation and LTE connection status.
- The Connection API exposes Internet Backup connection status.

Api changes from version 10.0 to 10.1

Changed API (v10.1)

Call Api Changes

- Expose phone number associated with the subscription
- Expose voicemails left on the line

Api changes from version 10.1 to 10.2

New API (v10.2)

Wifi State

- Add wifi global state API
- Deprecate expected_phys in wifi global configuration API

Api changes from version 10.2 to 11.0

New API (v11.0)

Update

- API to get the box update status

Standby

- API to configure box standby (either WiFi or box standby)

System

- API to shutdown box

SFP

- API to configure LAN SFP port on supported platforms

API change (v11.0)

Notification

- Update notification API to be able to customize notification server

Wifi

- Add custom_key_ssid to BSS status
- Standby API supersedes WiFi planning API (which may be removed in the future)

Api changes from version 11.0 to 11.1

API change (v11.1)

System

- The SystemModelInfo object can contain additional fields to indicate Eco-WiFi and WOP support

IPv6 Connection

- Add ipv6_prefix_firewall field to IPv6 configuration object, in order to enable the IPv6 firewall on secondary prefixes

Api changes from version 11.1 to 11.2

New API (v11.2)

File system

- Add api to get a FileInfo list from a list of file paths

Api changes from version 11.2 to 12.0

API change (v12.0)

Wifi

- The WifiApStatus field of WifiAp object has a new value stopping when a stop operation is pending due to param or disabled state
- The WifiAllowedComb object now have a psc field to indicate that this channel combination is using a Primary Scanning Channel (PSC)

Notifications

- Add new lan_host notification type to be notified when a new host is connected to the box for the first time
- Add new password_change notification type to be notified when the admin password has been changed

Api changes from version 12.0 to 12.1

API change (v12.1)

File System

- Add exifMode optional parameter to file list API to get exif data from supported images (jpeg, heic)

Wifi

- WifiCustomKeyParams can now have a max_use_count of 0. This means the key has no restriction of how many users can use it to associate to the ap.

Api changes from version 12.1 to 12.2

API change (v12.2)

LCD

- Add settings to control Freebo Ultra Limited Edition LED strip configuration

System

- Add capability flag to know if the Freebox Model supports LED strip configuration

Api changes from version 12.2 to 13.0

API change (v13.0)

Wifi

- The WifiApStatus field of WifiAp object has a new value 'disabled_temp' when AP is disabled temporarily.
- A new field named 'temp_disable_remaining_time' has been added to WifiAp object.
- Add API /wifi/temp_disable

Api changes from version 13.0 to 14.0

API change (v14.0)

Wifi

- Add new BSS encryption value wpa23_psk_ccmp_mrsno. When targeting an api version older than 14, this new encryption value is replaced by wpa2_psk_ccmp.
- Add new gcmp256 field in BSS config.
- Add new BSS info to inform if access point supports wep encryption or not
- The guest wifi is now using a dedicated network, and the name of the network can be changed. Use the WifiCustomKeyConfig api to enable/configure it.
- Added support for MLO (Multi Link Operation) configurations. See the MLOConfig API

Lan browser

- Add new lan host types.
- Add categories to lan/browser/types API.

API Update

LCD Configuration

- Add hide_led parameter to control the power LED on supported Freebox models

Api changes from version 14.0 to 15.0

API change (v15.0)

File system

- The file listing API returns an object rather than simply an array of entries
- The file listing API supports pagination

Freebox discovery

To discover a Freebox supporting this API you can either use mDNS, or make a HTTP request to mafreebox.freebox.fr to get API information.

Discovery using mDNS

This is the preferred method since it does not require to know the Freebox IP address.

The Freebox broadcasts the "_fbx-api._tcp" service

On iOS devices, you can use a NSNetServiceBrowser

On Android devices, you can use Network Service Discovery or JmDNS

On the TXT record you can obtain the following information:

Key	Description
api_version	The current API version on the Freebox
device_type	(DEPRECATED: use box_model)
api_base_url	The API root path on the HTTP server
uid	The device unique id
api_domain	The domain to use in place of hardcoded Freebox ip
https_available	Tells if https has been configured on the Freebox
https_port	Port to use for remote https access to the Freebox Api
box_model_name	Box model display name
box_model	Box model

Currently the existing box models are

box_model	Description
fbxgw-r1/full	Freebox Server (v6) revision 1
fbxgw-r2/full	Freebox Server (v6) revision 2
fbxgw-r1/mini	Freebox Mini revision 1
fbxgw-r2/mini	Freebox Mini revision 2
fbxgw-r1/one	Freebox One revision 1
fbxgw-r2/one	Freebox One revision 2
fbxgw7-r1/full	Freebox v7 revision 1
fbxgw8-r1/full	Freebox v8 revision 1
fbxgw9-r1/full	Freebox v9 revision 1

Discovery using HTTP

If you can, avoid this method because it requires to use a hardcoded address to retrieve API information.

If you make a HTTP get request on http://mafreebox.freebox.fr/api_version you can get the same API information as provided in mDNS.

Example request:

```
GET /api_version HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
{
   "uid": "23b86ec8091013d668829fe12791fdab",
   "device_name": "Freebox Server",
   "box_model": "fbxgw7-r1/full",
   "box_model_name": "Freebox v7 (r1)",
   "api_version": "15.0",
   "api_base_url": "/api/",
   "api_domain": "example.fbxos.fr",
   "https_available": true,
   "https_port": 3615
}
```

Only the fields available to build the API request URL (see below) are available if you connect remotely.

Discovery using HTTPS

Discovery using HTTPS works the same as discovery on HTTP. You can do an HTTP GET request on https://mafreebox.freebox.fr/api_version . You need to validate the certificate as explained below in HTTPS access.

Discovery using HTTPS is preferred to HTTP discovery if you can't use mDNS. You MUST implement the certificate validation in your app in order to use the API.

Building the API request URL

Once you've discovered a Freebox on the local network you can access the API at the following URL:

```
https://[api_domain]:[freebox_port]/[api_base_url]/v[major_api_version]/[api_url]
```

or for local access

https://mafreebox.freebox.fr/[api_base_url]/v[major_api_version]/[api_url]

Example:

```
https://example.fbxos.fr:3615/api/v15/login/
```

Remote connection port change discovery

When the https connection fails to a previously recorded https://[api_domain]:[https_port], you should attempt to discover if https_port has changed. This can happen either automatically (port is no longer valid), or manually if the user decided to change the port.

The https port is announced in a DNS "_https._tcp" **SRV** record. For example, for domain [example.fbxos.fr], the SRV record will be:

```
# _service._proto.name.      TTL class SRV priority weight port  target
_https._tcp.example.fbxos.fr 300 IN    SRV 13       37     12345 example.fbxos.fr
```

Here, only the "port" field of the SRV record is relevant, i.e **12345**. The SRV field is only populated for the https port, and only for the [api_domain] field of the API information.

Port change discovery is important to maintain remote connectability.

API conventions

Most API uses the REST architecture, pay attention to the http methods used for each request.

For requests with a body, you must use "application/json" content-type unless otherwise stated.

The API response is always a JSON object using utf8 encoding.

`APIResponse`

`success` **boolean** ***Read-only***
: indicates if the request was successful

`result` **object** ***Read-only***
: the result of the request.

: (It may be omitted if the request does not expect any result)

`error_code` **string** ***Read-only***
: In case of request error, this error_code provides information about the error.

: The possible error_code values are documented for each API.

`msg` **string** ***Read-only***
: In cas of error, provides a French error message relative to the error

Successful response example

```
{
   success: true,
   result: {
      logged_in: false,
      challenge: "WpsbHdkBpRpHLMGQHZ1ri1uUqa4ce6Dw"
   }
}
```

Error response example

```
{
   msg: "Requête invalide",
   success: false,
   error_code: "invalid_request"
}
```

The HTTP response code can also be used to error reason, for instance if you attempt to access to an API with invalid credential you will get a 403 error, or if you attempt to call an API with an invalid path you will get a 404 error.

HTTPS Access

Each Freebox is now automatically assigned a random domain name (api_domain), and an associated TLS certificate to enable secure access to API.

This is enabled by default and all applications MUST now use HTTPS to access the api. Unsecure access will be removed at some point.

Certificates used for HTTPS access are emitted by either 'Freebox ECC Root CA' in case of ECDSA access, or 'Freebox Root CA' in case of RSA.

You must validate the certificate chain, by using the following Root CA certificates:

Freebox ECC Root CA

```
-----BEGIN CERTIFICATE-----
MIICWTCCAd+gAwIBAgIJAMaRcLnIgyukMAoGCCqGSM49BAMCMGExCzAJBgNVBAYT
AkZSMQ8wDQYDVQQIDAZGcmFuY2UxDjAMBgNVBAcMBVBhcmlzMRMwEQYDVQQKDApG
cmVlYm94IFNBMRwwGgYDVQQDDBNGcmVlYm94IEVDQyBSb290IENBMB4XDTE1MDkw
MTE4MDIwN1oXDTM1MDgyNzE4MDIwN1owYTELMAkGA1UEBhMCRlIxDzANBgNVBAgM
BkZyYW5jZTEOMAwGA1UEBwwFUGFyaXMxEzARBgNVBAoMCkZyZWVib3ggU0ExHDAa
BgNVBAMME0ZyZWVib3ggRUNDIFJvb3QgQ0EwdjAQBgcqhkjOPQIBBgUrgQQAIgNi
AASCjD6ZKn5ko6cU5Vxh8GA1KqRi6p2GQzndxHtuUmwY8RvBbhZ0GIL7bQ4f08ae
JOv0ycWjEW0fyOnAw6AYdsN6y1eNvH2DVfoXQyGoCSvXQNAUxla+sJuLGICRYiZz
mnijYzBhMB0GA1UdDgQWBBTIB3c2GlbV6EIh2ErEMJvFxMz/QTAfBgNVHSMEGDAW
gBTIB3c2GlbV6EIh2ErEMJvFxMz/QTAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB
/wQEAwIBhjAKBggqhkjOPQQDAgNoADBlAjA8tzEMRVX8vrFuOGDhvZr7OSJjbBr8
gl2I70LeVNGEXZsAThUkqj5Rg9bV8xw3aSMCMQCDjB5CgsLH8EdZmiksdBRRKM2r
vxo6c0dSSNrr7dDN+m2/dRvgoIpGL2GauOGqDFY=
-----END CERTIFICATE-----
```

Freebox Root CA

```
-----BEGIN CERTIFICATE-----
MIIFmjCCA4KgAwIBAgIJAKLyz15lYOrYMA0GCSqGSIb3DQEBCwUAMFoxCzAJBgNV
BAYTAkZSMQ8wDQYDVQQIDAZGcmFuY2UxDjAMBgNVBAcMBVBhcmlzMRAwDgYDVQQK
DAdGcmVlYm94MRgwFgYDVQQDDA9GcmVlYm94IFJvb3QgQ0EwHhcNMTUwNzMwMTUw
OTIwWhcNMzUwNzI1MTUwOTIwWjBaMQswCQYDVQQGEwJGUjEPMA0GA1UECAwGRnJh
bmNlMQ4wDAYDVQQHDAVQYXJpczEQMA4GA1UECgwHRnJlZWJveDEYMBYGA1UEAwwP
RnJlZWJveCBSb290IENBMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA
xqYIvq8538SH6BJ99jDlOPoyDBrlwKEp879oYplicTC2/p0X66R/ft0en1uSQadC
sL/JTyfgyJAgI1Dq2Y5EYVT/7G6GBtVH6Bxa713mM+I/v0JlTGFalgMqamMuIRDQ
tdyvqEIs8DcfGB/1l2A8UhKOFbHQsMcigxOe9ZodMhtVNn0mUyG+9Zgu1e/YMhsS
iG4Kqap6TGtk80yruS1mMWVSgLOq9F5BGD4rlNlWLo0C3R10mFCpqvsFU+g4kYoA
dTxaIpi1pgng3CGLE0FXgwstJz8RBaZObYEslEYKDzmer5zrU1pVHiwkjsgwbnuy
```

```
WtM1Xry3Jxc7N/i1rxFmN/4l/Tcb1F7x4yVZmrzbQVptKSmyTEvPvpzqzdxVWuYi
qIFSe/njl8dX9v5hjbMo4CeLuXIRE4nSq2A7GBm4j9Zb6/l2WIBpnCKtwUVlroKw
NBgB6zHg5WI9nWGuy3ozpP4zyxqXhaTgrQcDDIG/SQS1GOXKGdkCcSa+VkJ0jTf5
od7PxBn9/TuN0yYdgQK3YDjD9F9+CLp8QZK1bnPdVGywPfL1iztngF9J6JohTyL/
VMvpWfS/X6R4Y3p8/eSio4BNuPvm9r0xp6IMpW92V8SYL0N6TQQxzZYgkLV7TbQI
Hw6v64yMbbF0YS9VjS0sFpZcFERVQiodRu7nYNC1jy8CAwEAAaNjMGEwHQYDVR0O
BBYEFD2erMkECujilR0BuER09FdsYIebMB8GA1UdIwQYMBaAFD2erMkECujilR0B
uER09FdsYIebMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMA0GCSqG
SIb3DQEBCwUAA4ICAQAZ2Nx8mWIWckNY8X2t/ymmCbcKxGw8Hn3BfTDcUWQ7GLRf
MGzTqxGSLBQ5tENaclbtTpNrqPv2k6LY0VjfrKoTSS8JfXkm6+FUtyXpsGK8MrLL
hZ/YdADTfbbWOjjD0VaPUoglvo2N4n7rOuRxVYIij11fL/wl3OUZ7GHLgL3qXSz0
+RGW+1oZo8HQ7pb6RwLfv42Gf+2gyNBckM7VVh9R19UkLCsHFqhFBbUmqwJgNA2/
3twgV6Y26qlyHXODUfV3arLCwFoNB+IIrde1E/JoOry9oKvF8DZTo/Qm6o2KsdZ
dxs/YcIUsCvKX8WCKtH6la/kFCUcXIb8f1u+Y4pjj3PBmKI/1+Rs9GqB0kt1otyx
Q6bqxqBSgsrkuhCfRxwjbfBgmXjIZ/a4muY5uMI0gbl9zbMFEJHDojhH6TUB5qd0
JJlI61gldaT5Ci1aLbvVcJtdeGhElf7pOE9JrXINpP3NOJJaUSueAvxyj/WWoo0v
4KO7njox8F6jCHALNDLdTsX0FTGmUZ/s/QfJry3VNwyjCyWDy1ra4KWoqt6U7SzM
d5jENIZChM8TnDXJzqc+mu00cI3icn9bV9flYCXLTIsprB21wVSMh0XeBGylKxeB
S27oDfFq04XSox7JM9HdTt2hLK96x1T7FpFrBTnALzb7vHv9MhXqAT90fPR/8A==
-----END CERTIFICATE-----
```

If you want your app to work in Italy, in addition to changing the default domain, you should trust this ECC Root CA:

```
-----BEGIN CERTIFICATE-----
MIICOjCCAcCgAwIBAgIUI0Tu7zsrBJACQIZgLMJobtbdNn4wCgYIKoZIzj0EAwIw
TDELMAkGA1UEBhMCSVQxDjAMBgNVBAgMBUl0YWx5MQ4wDAYDVQQKDAVJbGlhZDEd
MBsGA1UEAwwUSWxpYWRib3ggRUNDIFJvb3QgQ0EwHhcNMjAxMTI3MDkzODEzWhcN
NDAxMTIyMDkzODEzWjBMMQswCQYDVQQGEwJJVDEOMAwGA1UECAwFSXRhbHkxDjAM
BgNVBAoMBUlsaWFkMR0wGwYDVQQDDBRJbGlhZGJveCBFQ0MgUm9vdCBDQTB2MBAG
ByqGSM49AgEGBSuBBAAiA2IABMryJyb2loHNAioY8IztN5MI3UgbVHVP/vZwcnre
ZvJOyDvE4HJgIti5qmfswlnMzpNbwf/MkT+7HAU8jJoTorRm1wtAnQ9cWD3Ebv79
RPwtjjy3Bza3SgdVxmd6fWPUKaNjMGEwHQYDVR0OBBYEFDUij/4lpoJ+kOXRyrcM
jf2RPzOqMB8GA1UdIwQYMBaAFDUij/4lpoJ+kOXRyrcMjf2RPzOqMA8GA1UdEwEB
/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMAoGCCqGSM49BAMCA2gAMGUCMQC6eUV1
pFh4UpJOTc1JToztN4ttnQR6rIzxMZ6mNCe+nhjkohWp24pr7BpUYSbEizYCMAQ6
LCiBKV2j7QQGy7N1aBmdur17ZepYzR1YV0eI+Kd978aZggsmhjXENQYVTmm/XA==
-----END CERTIFICATE-----
```

and this RSA Root CA:

```
-----BEGIN CERTIFICATE-----
MIIFiTCCA3GgAwIBAgIUTXoJE/kJnSKpxk5FjcmqmGah9zcwDQYJKoZIhvcNAQEL
BQAwTDELMAkGA1UEBhMCSVQxDjAMBgNVBAgMBUl0YWx5MQ4wDAYDVQQKDAVJbGlh
ZDEdMBsGA1UEAwwUSWxpYWRib3ggUlNBIFJvb3QgQ0EwHhcNMjAxMTI3MDkzODEy
WhcNNDAxMTIyMDkzODEyWjBMMQswCQYDVQQGEwJJVDEOMAwGA1UECAwFSXRhbHkx
DjAMBgNVBAoMBUlsaWFkMR0wGwYDVQQDDBRJbGlhZGJveCBSU0EgUm9vdCBDQTCC
AiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBANXKZSyCmix6jt7jUmaCP4XF
caF4azeYZuA8A4sWQmQXRWTDj8oNClE5w7zo5qUYzHIBOubKY7hhIU7RXYR5Bdny
arNRoo5ZBplgEkv3G00IgXY2/lCywPQ8WorAn0k/uaRce239r6EkGC3fxCA3Asnc
q9lNkUoWaf0GktJai0DuW7bNY8cq+vzZpy/36ey0LQ4OoehfiA6vlUTVWakpjecJ
ller1RfVlgEH26wnerGge3LYBZv27XiahCft54AQLxRY3H/z8XpKsPnJJrrhEvSo
2p64Bd+g7ZbzCdeakrypjVC/eWn14UzbcBVgh0p4F4990LuGxLVqyh6XcZOSSi01
4fpca5xPDCiohEX7ehMLpdURbhKzPj17IpwTmonfVmxkvV8rca1PqhDPEOouwPtc
M55eCgtwgSBeDznFKD7s+az/SZYC16GTgyXTCd2lId/J1unZ4pdzNVMAglTpnGgz
eQkHvfcVYdJj49tOtW0OpSPBiNIC6LCVY9wtH5dRMm0k+A8QDP+9HQaOs3LIUMwu
WGePw6r+eXUYw/2yO0z3zI/63hOpzZVixW+T7h3SY5B+sTrxR9fRD1oyk/rPV4I3
X5mZnyzSowjcN3+hSkGIZBleMO3CHaYleIf1/9HHhCJCVeeJ4kwEWY18Z0A+ohFh
D/dipgwmLCDH1/irDT4pAgMBAAGjYzBhMB0GA1UdDgQWBBTcW1RrTVIizaqkrkTI
CSw86qDJkTAfBgNVHSMEGDAWgBTcW1RrTVIizaqkrkTICSw86qDJkTAPBgNVHRMB
Af8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjANBgkqhkiG9w0BAQsFAAOCAgEAOfi6
fCuVLJD+vttO34cdB3i5hofmNrzgLh/spnwdm4y9EvvVqDvLdVLEIbvKf0QEcW0Y
dwP1BgmKwwHVv9YydHov8Jr4ANoGGXJnPLPcYDhRnixYEQmlTwSL/CLUcQ2hQWXx
Oc0k1jJB7uk6TPdX2YJyW4NpIcwI2sa5Dg/L8PqM0/pMYnMyG1hBwUc2M2qg3qTJ
zeiYT9zBHxS/JXA40yH4g9NzcFisVuYrfmINb11GmeqClm2OWehSdgdv9tEph3NW
ntJTENRrDvuj/pGZsnbofzgHNN6/nanymmrEPxG+xUGLIAW7zFndTKityhJ9FRqF
ultoZR2D19hh+n1277TSCPRJzUpq9rrfiqukjua3UjBzEvevnmSbLs1bXcNAxFYN
oZZ2euHoBv+E3BHjGik4RUkEJYtf5Xh+iffk4zTMfKBERn40fB7yF1xzxyoziltL
VxfueF9V6N7qjo5Ia7kiShXXsB+QdQdweuxWm1pPYmMbfTxNEqFUs3GhwEjzLaJc
cJOedwCT4ntbyCcTQaRlDL8QFjdE4gNm2ZaoG+gqGTLPS55H+ZvLsgUCiR5YY44N
G2Gkv4w/V/eB3eAvd5lgm6oOe8ehdr5JdpD6wnW2GOHs4SBdBo6yR+4RgEimNmgF
Yu11tlZsB2Iw/TT1EyPVb5z6tK4wUgWLNFAvjXU=
-----END CERTIFICATE-----
```

Authentication

Unless otherwise stated API access must be authenticated using the procedure described in the following document

Login

Each application identified with an *app_name* must gain access to Freebox API before being able to use the api. This procedure can only be initiated from the local network, and the user must have access to the Freebox front panel to grant access to the app. Note that since you must be on the local network, you must do your requests on https://mafreebox.freebox.fr for the initial app authorization, since adding a new app is forbidden from the outside network, and then use the generated [api_domain]:[freebox_port] for subsequent accesses.

Once the user authorize the app, the app will be provided with a unique *app_token* associated with a set of default permissions.

This *app_token* must be store securely by the app, and will not be exchanged in clear text for the following requests.

Note that the user can revoke the *app_token*, or edit its permissions afterwards. For instance if the user resets the admin password, app permissions will be reset.

Then the app will need to open a *session* to get an *auth_token*. The app will then be authenticated by adding this session_token in HTTP headers of the following requests. The validity of the *auth_token* is limited in time and the app will have to renew this *auth_token* once in a while.

Obtaining an *app_token*

TokenRequest object

TokenRequest objects have the following attributes

TokenRequest

- **app_id** string
  A unique app_id string
- **app_name** string
  A descriptive application name (will be displayed on lcd)
- **app_version** string
  app version
- **device_name** string
  The name of the device on which the app will be used

Request authorization

This is the first step, the app will ask for an *app_token* using the following call. A message will be displayed on the Freebox LCD asking the user to grant/deny access to the requesting app.

Once the app has obtained a valid app_token, it will not have to do this procedure again unless the user revokes the app_token.

POST /api/v8/login/authorize/

Example request:

```
POST /api/v8/login/authorize/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "app_id": "fr.freebox.testapp",
   "app_name": "Test App",
   "app_version": "0.0.7",
   "device_name": "Pc de Xavier"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true,
   "result": {
      "app_token": "dyNYgfK0Ya6FWGqq83sBHa7TwzWo+pg4fDFUJHShcjVYzTfaRrZzm93p7OTAfH/0",
      "track_id": 42
   }
}
```

Track authorization progress

Once the authorization request has been made, the app should monitor the token status by using the following API and using the *track_id* returned by the previous call.

The status can have one of the following values:

Status	Description
unknown	the app_token is invalid or has been revoked
pending	the user has not confirmed the authorization request yet
timeout	the user did not confirmed the authorization within the given time
granted	the app_token is valid and can be used to open a session
denied	the user denied the authorization request

The app should monitor the status until it is different from pending. You MUST implement this monitoring, otherwise your authorization will be invalid, even if the user grants you access.

`GET /api/v8/login/authorize/{track_id}`

Example request:

```
GET /api/v8/login/authorize/42 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "status": "pending",
        "challenge": "Bj6xMqoe+DCHD44KqBljJ579seOXNWr2"
    }
}
```

Example response once the user has validated the request:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "status": "granted",
        "challenge": "Bj6xMqoe+DCHD44KqBljJ579seOXNWr2"
    }
}
```

Obtaining a *session_token*

To protect the *app_token* secret, it will never be used directly to authenticate the application, instead the API will provide a challenge the app will combine to its *app_token* to open a session and get a *session_token*

The app will then have to include the *session_token* in the HTTP headers of the following requests

SessionStart object

SessionStart objects have the following attributes:

`SessionStart`

`app_id` **string**
: Same app_id used in `TokenRequest` to get the *app_token*

`app_version` **string**
: app version

`password` **string**
: The password computed using the *challenge* and the *app_token*

To compute the password you have to compute the hmac-sha1 of the challenge and the app_token

password = hmac-sha1(app_token, challenge)

Getting the challenge value

The challenge returned by the API will change frequently and have a limited time validity.

There are several ways of getting the current challenge value, it will always be included in response requesting the app authentication. It is also included in the authorization tracking API response.

You can also explicitly request a challenge with the following API

`GET /api/v8/login/`

Example request:

```
GET /api/v8/login/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "logged_in": false,
        "challenge": "VzhbtpR4r8CLaJle2QgJBEkyd8JPb0zL"
```

```
    }
}
```

Opening a session

Once you have the challenge you just need use the following API to get a *session_token*

NOTE: in case of session opening failure, ensure that the box you're connected is the one you expect by checking the uid returned in the answer.

In the response you get your app permissions. App permissions are:

App permission	Description
settings	Allow modifying the Freebox settings (reading settings is always allowed)
contacts	Access to contact list
calls	Access to call logs
explorer	Access to filesystem
downloader	Access to downloader
parental	Access to parental control (obsolete)
pvr	Access personal video recorder
profile	Access to user profile management

NOTE: A permission not listed in app permissions is equivalent to having this permission set to false.

`POST /api/v8/login/session/`

Example request:

```
POST /api/v8/login/session/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "app_id": "fr.freebox.testapp",
   "password": "d4da8517c2c25b1b145f2e5ba91bd0589fc0053d"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true,
   "result" : {
         "session_token" : "35JYdQSvkcBYK84IFMU7H86clfhS75OzwlQrKlQN1gBch\/Dd62RGzDpgC7YB9jB2",
         "challenge":"jdGL6CtuJ3Dm7p9nkcIQ8pjB+eLwr4Ya",
         "permissions": {
               "downloader": true,
         }
    }
}
```

Example response with invalid password:

```
HTTP/1.1 403 Forbidden
Content-Type: application/json; charset=utf-8
```

```
{
    "msg": "Erreur d'authentification de l'application",
    "success": false,
    "uid": "23b86ec8091013d668829fe12791fdab",
    "error_code": "invalid_token",
    "result": {
         "challenge": "DLjXFEf1kaDwAEn6xRUnEVPU++gnjiSn"
    }
}
```

Closing the current session

to close the current session you can use the following call

`POST /api/v8/login/logout/`

Example request:

```
POST /api/v8/login/logout/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true,
}
```

Make an authenticated call to the API

Once you have a valid *session_token* you should use it by add the the HTTP header **X-Fbx-App-Auth**

Example request:

```
GET /api/v8/login/session/ HTTP/1.1
Host: mafreebox.freebox.fr
X-Fbx-App-Auth: 35JYdQSvkcBYK84IFMU7H86clfhS75OzwlQrKlQN1gBch\/Dd62RGzDpgC7YB9jB2
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true,
   "result" : {
         [
            ...
         ]
    }
}
```

Authentication errors

When attempting to access the API, you may encounter the following authentication errors:

NOTE that in this case the HTTP 403 return code will be used as well

Error	Description
auth_required	Invalid session token, or not session token sent

Error	Description
invalid_token	The app token you are trying to use is invalid or has been revoked
pending_token	The app token you are trying to use has not been validated by user yet
insufficient_rights	Your app permissions does not allow accessing this API
denied_from_external_ip	You are trying to get an app_token from a remote IP
invalid_request	Your request is invalid
ratelimited	Too many auth error have been made from your IP
new_apps_denied	New application token request has been disabled
apps_denied	API access from apps has been disabled
internal_error	Internal error

WebSocket API

WebSocket allow bidirectional communication between your api client and the Freebox. This allow more interactivity without the need of frequently polling data from the Freebox.

For WebSocket access, you must use the same Authentication mechanism as for regular http api request. This means that you must include a proper **X-Fbx-App-Auth** header when you open the WebSocket connection.

Once the connection is established, most of messages sent via the WebSocket are text based (using utf-8 as per WebSocket specifications) and encoded as JSON objects.

The WebSocket frames maximum accepted size is 1 MB

WebSocket API conventions

As for HTTP api, the client can make requests to the Freebox (the available requests are specified per api).

The requests use the following format:

`WebSocketRequest`

- `request_id` `int` ***Optionnal***
  if you specify a request_id in your request, it will be added in the corresponding reply, so that you can correlate responses to the request
- `action` `string`
  the request 'action'
  (available actions are described in each api)

Other fields, related to a specific action, will be used as 'action' parameters

Responses to such requests will have the following format:

`WebSocketResponse`

- `request_id` `int`
  if you set a request_id in your WebSocketRequest, the same request_id will be returned in the associated response
- `action` `string`
  the action specified in the associated WebSocketRequest
- `success` `boolean` ***Read-only***
  indicates if the request was successful
- `result` `object` ***Read-only***
  the result of the request.
  (It may be omitted if the request does not expect any result)
- `error_code` `string` ***Read-only***
  In case of request error, this error_code provides information about the error.
  The possible error_code values are documented for each API.
- `msg` `string` ***Read-only***
  In cas of error, provides a French error message relative to the error

When the Freebox wants to send a notification on WebSocket it will have the following format:

`WebSocketNotification`

- `action` `enum` ***Read-only***
  The action will have the value 'notification'
- `success` `boolean` ***Read-only***
  will be True
- `source` `string` ***Read-only***
  The name of the source of the notification
- `event` `string` ***Read-only***
  The name of event that generated the notification
- `result` `object` ***Read-only***
  the content of the notification (may be omitted if no data has to be transferred along with the notification)

WebSocket event API

This API is used to send events to an application, removing the need to poll when waiting for a long operation to complete. It follows the conventions of the WebSocket API.

This is a text websocket that sends json, one per line.

`GET /api/v8/ws/event`

The application sends `RegisterACtion` to subscribe to an event channel. It will subsequently receive events on this websocket.

Register Action

`RegisterAction`

- `action` `enum`
  Value should be "register"
- `events[]` `array of enum`
  List of events to subscribe for. Possible values:

event name	event result object type	Description
vm_state_changed	VmStateChange	VM status has changed
vm_disk_task_done	VmDiskTask	VM disk task done
lan_host_l3addr_reachable	LanHost	LAN machine had an L3 address (IPv4 or IPv6) become reachable. Usually when a machine appears on the network, or changes IP.
lan_host_l3addr_unreachable	LanHost	LAN machine had an L3 address (IPv4 or IPv6) become unreachable. Usually when a machine disappears from the network (after a timeout), or changes IP.

Response is usually `{"success": true, "action": "register"}`

Events will be sent as `WebSocketNotification` ; the event name will be split in source (prefix) and event (suffix). For example, vm_disk_task_done will have source "vm", and event "disk_task_done":

```
{
   "action" : "notification",
   "success" : true,
   "source" : "vm"
   "event" : "disk_task_done",
   "result" : {
      "done" : true,
      "error" : false,
      "id" : 1
   }
}
```

API List

Air Media

AirMedia API

This API allows you to multimedia stream to any airmedia device reachable by the Freebox, as well as configuring the airmedia server hosted on the Freebox Server.

AirMedia Errors

When attempting to access the file airmedia API, you may encounter the following errors:

error_code	Description
unknown_target	No airmedia device with this name in range
no_client	No airmedia client connected
set_pass	Unable to update password
set_onscreen_code	Unable to activate onscreen code
no_ctrl	Remote control is unavailable
http	Internal HTTP error
bad_session	No stream session found
bad_name	Invalid airmedia name
bad_device_id	No device with this id
bad_remote_id	No remote control with this id
req_in_progress	You should try again, another request is still processing
fetch	Unable to get slideshow information
no_display	No screen available
playback_state	Invalid playback state
no_slideshow_srv	Slideshow is not supported
no_mem	Internal error
inout_file	Unable to read input file
no_volume_control	Volume control is not available
connect	Error connecting to the airmedia device
unauthorized	This device requests a password
unsupported_media	The device does not support this format
bad_type	Invalid file type
unimplemented	Unimplemented

AirMedia Config Object

AirMedia config has the following attributes:

`AirMediaConfig`

`enabled` `bool`
Enable/Disable the airmedia server

`password` `string` ***Write-only***
If not empty, the client will have to enter a password to be able to use this airmedia server

AirMedia Configuration API

Get the current AirMedia configuration

`GET /api/v8/airmedia/config/`

Returns the current AirMediaConfig

Example request:

```
GET /api/v8/airmedia/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   success: true,
   result: {
      enabled: true
   }
}
```

Update the current AirMedia configuration

`PUT /api/v8/airmedia/config/`

Update the current AirMediaConfig

Example request:

```
PUT /api/v8/airmedia/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "enabled": true,
   "password": "3615"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   success: true,
   result: {
      enabled: true
   }
}
```

AirMedia Receiver Object

AirMedia receivers have the following attributes

`AirMediaReceiver`

`name` `string` ***Read-only***
AirMedia name

`password_protected` `bool` ***Read-only***
Is set to true the receiver is protected by a password

`capabilities` `map` ***Read-only***
List of receiver capabilities from the following list

Capability	Description
photo	can display photos
audio	can play audio files
video	can play video files
screen	can display remote screen

Get the list of available AirMedia receivers

You can get the list of `AirMediaReceiver` connected to the Freebox Server using this API

`GET /api/v8/airmedia/receivers/`

Get the list of `AirMediaReceiver` connected to the Freebox Server

Example request:

```
GET /api/v8/airmedia/receivers/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   success: true,
   result: [
      {
         capabilities: {
            photo: true,
            screen: false,
            audio: true,
            video: true
         },
         name: "Freebox Player",
         password_protected: true
      },
      {
         capabilities: {
            photo: false,
            screen: false,
            audio: true,
            video: false
         },
         name: "Freebox Server",
         password_protected: false
      }
   ]
}
```

Interacting with an AirMedia receiver

Once you have selected an available `AirMediaReceiver` you can start interacting with it by sending media with the following API.

AirMedia receiver request

`AirMediaReceiverRequest`

`action` `enum`

Action	Description
start	start playing a media
stop	stop playing a media

`media_type` `string`

Media Type	Description
photo	display a photo
video	display a video

`password` `string`
Optional receiver password.

`position` `int`
Start position for a video.

The start position is expressed in percent * 1000, for instance 50000 means 50% of the video

`media` `string`
The media to play.

- For video media, you have to specify the media URL, for instance http://anon.nasa-global.edgesuite.net/HD_downloads/GRAIL_launch_480.mov
- For photo media, you have to specify the file path on the Freebox Server (base64 encoded as returned in fs/ls call), for instance L0Rpc3F1ZSBkdXIvUGhvdG9zL1JvY2tldHMvRFNDXzM0OTEuanBn

Sending a new request to an AirMedia receiver

`POST /api/v8/airmedia/receviers/{receiver_name}/`

Example: display a photo on the Freebox Player:

```
POST /api/v8/airmedia/receivers/Freebox%20Player/ HTTP/1.1
Host: mafreebox.freebox.fr

{
```

```
   "action": "start",
   "media_type": "photo",
   "media": "L0Rpc3F1ZSBkdXIvUGhvdG9zL1JvY2tldHMvRFNDXzM0OTEuanBn",
   "password": "1111"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
}
```

Example: play a video the Freebox Player:

```
POST /api/v8/airmedia/receivers/Freebox%20Player/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "action": "start",
   "media_type": "video",
   "media": "http://anon.nasa-global.edgesuite.net/HD_downloads/GRAIL_launch_480.mov",
   "password": "1111"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
}
```

Example: stop the current AirMedia video on Freebox Player:

```
POST /api/v8/airmedia/receivers/Freebox%20Player/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "action": "stop",
   "media_type": "video"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
}
```

Calls / Contacts

Call

With the call API you access the Freebox call logs.

Call Errors

When attempting to access the call API, you may encounter the following errors:

error_code	Description
internal_error	Internal error
invalid_id	No call with this id
invalid_category	Invalid call category

Call Object

Call entries have the following properties

`CallEntry`

`id` `int` ***Read-only***
id

`type` `enum` ***Read-only***
The valid call types are:

Type	Description
missed	Missed incoming call
accepted	Incoming call
outgoing	Outgoing call

`datetime` `timestamp` ***Read-only***
Call creation timestamp.

`number` `string` ***Read-only***
Callee number for outgoing calls. Caller number for incoming calls.

`name` `string` ***Read-only***
Callee name for outgoing calls. Caller name for incoming calls.

For incoming call if the network does not provide a contact name, we try to use the contact database to find a suitable name

`duration` `int` ***Read-only***
Call duration in seconds.

`new` `bool`
Call entry has not been acknowledged yet.

`contact_id` `int` ***Read-only***
If the number matches an entry in the contact database, the id of the matching contact.

Call API

This is the call API

List every calls

`GET /api/v10/call/log/`
Returns the collection of all `CallEntry` call entries

Example request:

```
GET /api/v10/call/log/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
   success: true,
   result: [
      {
         number: "0102030405",
         type: "missed",
         id: 69,
         duration: 1,
         datetime: 1359546363,
         contact_id: 56,
         line_id: 0,
         name: "r0ro (Freebox)",
         new: true
      },
      {
         number: "**1",
         type: "outgoing",
         id: 68,
         duration: 5,
         datetime: 1359545960,
         contact_id: 0,
         line_id: 0,
         name: "**1",
         new: false
      }
   ]
}
```


Delete all calls

`POST /api/v10/call/log/delete_all/`

Remove all CallEntry call entries

Example request:

```
GET /api/v10/call/log/delete_all HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

Mark all calls as read

`POST /api/v10/call/log/mark_all_as_read/`

Mark all CallEntry call entries as read

Example request:

```
GET /api/v10/call/log/mark_all_as_read HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

Access a given call entry

`GET /api/v10/call/log/{id}`

Returns the CallEntry task with the given id

Example request:

```
GET /api/v10/call/log/69 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
   success: true,
   result: {
      number: "0102030405",
      type: "missed",
      id: 69,
      duration: 1,
      datetime: 1359546363,
      contact_id: 56,
      line_id: 0,
      name: "Romain Bureau",
      new: true
   }
}
```


Delete a call

`DELETE /api/v10/call/log/{id}`

Deletes the CallEntry with the given id.

Example request:

```
DELETE /api/v10/call/log/69 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Update a call entry

`PUT /api/v10/call/log/{id}`

Updates the CallEntry task with the given id

Example request:

```
PUT /api/v10/call/log/69 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "new": "false"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   success: true,
   result: {
      number: "0102030405",
      type: "missed",
      id: 69,
      duration: 1,
      datetime: 1359546363,
      contact_id: 56,
      line_id: 0,
      name: "Romain Bureau",
      new: false
   }
}
```

Account

The account API returns the phone number associated with the subscription.

`GET /api/v10/call/account`

Returns an object containing the phone number associated with the subscription.

Example request:

```
GET /api/v10/call/account/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
      "phone_number": "0999999999",
  }
}
```

Voicemail

The voicemail API lets one access voicemail messages.

Voicemail Errors

The following errors may be encountered with the voicemail API:

error_code	Description
internal_error	Internal error
invalid_id	No voicemail with this id

Voicemail Object

Voicemail entries have the following properties

`VoicemailEntry`

`id` string ***Read-only***
: id

`country_code` string ***Read-only***
: Country code part of the caller number. May be empty.

`phone_number` string ***Read-only***
: Caller number. May be empty.

`date` timestamp ***Read-only***
: Voicemail creation timestamp.

`read` bool
: Voicemail read status

`duration` int ***Read-only***
: Voicemail duration in seconds

Voicemail API

List voicemails

`GET /api/v10/call/voicemail/`

Returns a collection of all `VoicemailEntry` voicemail entries

Example request:

```
GET /api/v10/call/voicemail/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": [
    {
      "phone_number": "699999999",
      "read": false,
      "id": "20221215_154135_r0334371508.au",
      "duration": 8,
      "country_code": 33,
      "date": 1671115295
    }
  ]
}
```

Access a specific voicemail entry

`GET /api/v10/call/voicemail/{id}`

Returns the VoicemailEntry task with the given id

Example request:

```
GET /api/v10/call/voicemail/20221215_154135_r0334371508.au HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
      "phone_number": "699999999",
      "read": false,
      "id": "20221215_154135_r0334371508.au",
      "duration": 8,
      "country_code": 33,
      "date": 1671115295
  }
}
```

Delete a voicemail

`DELETE /api/v10/call/voicemail/{id}`

Deletes the VoicemailEntry with the given id.

Example request:

```
DELETE /api/v10/call/voicemail/20221215_154135_r0334371508.au HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Update a voicemail entry

`PUT /api/v10/call/voicemail/{id}`

Updates the VoicemailEntry with the given id

Example request:

```
PUT /api/v10/call/voicemail/20221215_154135_r0334371508.au HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "phone_number": "699999999",
  "read": true,
  "id": "20221215_154135_r0334371508.au",
  "duration": 8,
  "country_code": 33,
  "date": 1671115295
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
    "phone_number": "699999999",
    "read": true,
    "id": "20221215_154135_r0334371508.au",
    "duration": 8,
    "country_code": 33,
    "date": 1671115295
  }
}
```

Retrieve a voicemail

`GET /api/v10/call/voicemail/{id}/audio_file`

Download voicemail message in WAV format.

Example request:

```
GET /api/v10/call/voicemail/20221215_154135_r0334371508.au/audio_file HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: audio/wav; charset=utf-8
Content-Length: 60218
Content-Disposition: inline; filename="20221215_154135_r0334371508.wav"

/* binary data */
```

Contacts

The contact API allow to interact with the contact list stored on the Freebox

Contacts Errors

When attempting to access the contact API, you may encounter the following errors:

error_code	Description
noent	no entry with this id
exists	an entry already exists
no_match	no entry matched your request

Contact Objects

Contact Entry

Contact entries have the following properties

`ContactEntry`

`id` **int**
contact id

`display_name` **string**
contact display name

`first_name` **string**
contact first name

`last_name` **string**
contact last name

`company` **string**
contact company name

`photo_url` **string**
contact photo URL

NOTE the photo URL can be embedded (for instance " […]")

`last_update` **timestamp**
contact last modification timestamp

`notes` **string**
contact last modification timestamp

`addresses[]` **array of ContactAddress**
list of contact postal addresses

`emails[]` **array of ContactEmail**
list of contact email addresses

`numbers[]` **array of ContactNumber**
list of contact phone numbers

`urls[]` **array of ContactUrl**
list of contact URL

Contact Number

Contact number have the following properties

`ContactNumber`

`id` **int**
address id

`contact_id` **int**
id of the related contact

`type` **enum**
Type of number

Type	Description
fixed	fixed phone
mobile	mobile phone
work	work
fax	fax
other	other

`number` **string**

`is_default` **bool**
is this number the preferred contact phone number

`is_own` **bool**
is this number the Freebox owner number

Contact Address

Contact address have the following properties

`ContactAddress`

`id` **int**
address id

`contact_id` **int**
id of the related contact

`type` **enum**
Type of email

Type	Description
home	home address
work	work address
other	other

`number` **string**

`street` **string**

`street2` **string**

`city` **string**

`zipcode` **string**

`country` **string**

Contact Url

Contact URL have the following properties

`ContactUrl`

`id` **int**
address id

`contact_id` **int**
id of the related contact

`type` **enum**
Type of URL

Type	Description
profile	profile address
blog	blog address
site	website address

Type	Description
other	other

`url` `string`

URL address

Contact Email

Contact email have the following properties

`ContactEmail`

`id` `int`

address id

`contact_id` `int`

id of the related contact

`type` `enum`

Type of address

Type	Description
home	home address
work	work address
other	other

`email` `string`

email address

Contact API

Get a list of contacts

`GET /api/v8/contact/`

Returns the collection of all `ContactEntry`

Parameters

- **start** (*int*) – Offset
- **limit** (*int*) – Limit of contact to return (-1 means no limit)
- **group_id** (*int*) – Return only the contacts that belong to this group

Example request:

```
GET /api/v8/contact/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
        {
            "last_name": "Niel",
            "company": "Free",
            "photo_url": " [ ... ]",
            "id": 2,
            "birthday": "",
            "last_update": 1363964483,
            "display_name": "",
            "emails": [
                {
                    "id": 2,
                    "contact_id": 2,
                    "type": "home",
                    "email": "rocket@launchpad.free"
                }
            ],
            "urls": [
                {
                    "id": 1,
                    "contact_id": 2,
                    "url": "http://www.free.fr/",
                    "type": "site"
                }
            ],
            "notes": "",
            "first_name": "Xavier"
        },

        [ ... ],

        {
            "last_name": "Mamie",
            "first_name": "Kipic",
            "company": "",
            "photo_url": " [ ... ] ",
            "id": 1,
            "birthday": "",
            "numbers": [
                {
                    "number": "0612345678",
                    "type": "fixed",
                    "id": 1,
                    "contact_id": 1,
                    "is_default": false,
                    "is_own": false
                }
            ],
            "last_update": 1363973599,
            "display_name": "Mamie",
            "emails": [
                {
                    "id": 1,
                    "contact_id": 1,
                    "type": "home",
                    "email": "mamie@example.org"
                }
            ],
            "urls": [
                {
                    "id": 3,
```

```
                    "contact_id": 1,
                    "url": "ftp://free.fr",
                    "type": "site"
                }
            ],
            "addresses": [
                {
                    "street2": "",
                    "type": "home",
                    "country": "France",
                    "id": 1,
                    "street": "8 rue du pont",
                    "contact_id": 1,
                    "city": "Paris",
                    "zipcode": "75008",
                    "number": "11"
                }
            ],
            "notes": ""
        }
    ]
}
```


Access a given contact entry

`GET /api/v8/contact/{id}`

Returns the ContactEntry with the given id

Example request:

```
GET /api/v8/contact/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
      "last_name": "Mamie",
      "first_name": "Kipic",
      "company": "",
      "photo_url": " [ ... ] ",
      "id": 1,
      "birthday": "",
      "numbers": [
          {
              "number": "0612345678",
              "type": "fixed",
              "id": 1,
              "contact_id": 1,
              "is_default": false,
              "is_own": false
          }
      ],
      "last_update": 1363973599,
      "display_name": "Mamie",
      "emails": [
          {
              "id": 1,
              "contact_id": 1,
              "type": "home",
              "email": "mamie@example.org"
          }
      ],
      "urls": [
          {
              "id": 3,
              "contact_id": 1,
              "url": "ftp://free.fr",
              "type": "site"
          }
      ],
      "addresses": [
          {
              "street2": "",
              "type": "home",
              "country": "France",
              "id": 1,
              "street": "8 rue du pont",
              "contact_id": 1,
              "city": "Paris",
              "zipcode": "75008",
              "number": "11"
          }
      ],
      "notes": ""
    }
}
```


Create a contact

`POST /api/v8/contact/`

Creates a new ContactEntry

Example request:

```
POST /api/v8/contact/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "display_name": "Sandy Kilo",
   "first_name": "Sandy",
   "last_name":"Kilo"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "last_name": "Kilo",
        "company": "",
```

```
        "photo_url": "",
        "id": 10,
        "birthday": "",
        "last_update": 1372433423,
        "display_name": "Sandy Kilo",
        "notes": "",
        "first_name": "Sandy"
    }
}
```

Delete a contact

`DELETE /api/v8/contact/{id}`

Deletes the ContactEntry with the given id.

Example request:

```
DELETE /api/v8/contact/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Update a contact entry

`PUT /api/v8/contact/{id}`

Updates the ContactEntry with the given id

Example request:

```
PUT /api/v8/contact/4 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "company": "Freebox"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "last_name": "Anderson",
        "company": "Freebox",
        "photo_url": " [ ... ]",
        "id": 4,
        "birthday": "",
        "last_update": 1363977825,
        "display_name": "Thomas A. Anderson",
        "emails": [
            {
                "id": 3,
                "contact_id": 4,
                "type": "home",
                "email": "neo@matrix.com"
            }
        ],
        "notes": "",
        "first_name": "Thomas"
    }
}
```

Contact Related objects API

Contact related entries such as phone numbers, addresses, URLs and emails are all handled the same way.

Below we'll document the numbers API, you can use the same calls with addresses, URL and emails.

Get the list of numbers for a given contact

`GET /api/v8/contact/{contact_id}/[numbers|addresses|urls|emails]/`

Returns the collection of all ContactNumber for a given contact

Example request:

```
GET /api/v8/contact/4/numbers/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
        {
            "number": "+13374242",
            "type": "fixed",
            "id": 6,
            "contact_id": 4,
            "is_default": false,
            "is_own": false
        },
        {
            "number": "0611223344",
            "type": "mobile",
            "id": 5,
            "contact_id": 4,
            "is_default": false,
            "is_own": false
        }
    ]
}
```

Access a given contact number

`GET /api/v8/[number,address,url,email]/{id}`

Returns the ContactNumber with the given id

Example request:

```
GET /api/v8/number/6 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "number": "+13374242",
        "type": "fixed",
        "id": 6,
        "contact_id": 4,
        "is_default": false,
        "is_own": false
    }
}
```

Create a contact number

`POST /api/v8/[number,address,url,email]/`

Creates the ContactNumber

Example request:

```
POST /api/v8/number/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "contact_id":9,
   "number":"0144456789",
   "type":"fixed"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "number": "0144456789",
        "type": "fixed",
        "id": 18,
        "contact_id": 9,
        "is_default": false,
        "is_own": false
    }
}
```

Delete a contact number

`DELETE /api/v8/[number,address,url,email]/{id}`

Deletes the ContactNumber with the given id.

Example request:

```
DELETE /api/v8/number/6 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Update a contact number

`PUT /api/v8/[number,address,url,email]/{id}`

Updates the ContactNumber with the given id

Example request:

```
PUT /api/v8/number/5 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "number": "0655667788",
  "type": "mobile"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "number": "0655667788",
        "type": "mobile",
        "id": 5,
        "contact_id": 4,
        "is_default": false,
        "is_own": false
    }
}
```

Configuration

Connection API

This API provides Freebox connection settings information.

Connection Errors

When attempting to access the file connection API, you may encounter the following errors:

error_code	Description
inval	invalid request
nodev	no device found with this name
noent	no entity found with this name

error_code	Description
netdown	network is down
busy	device is busy
invalid_port	invalid port
insecure_password	the password is too weak to enable remote access
invalid_provider	invalid ddns provider name
invalid_next_hop	invalid next hop address (should be a link local address)

Connection status

Connection status object

`ConnectionStatus`

`state` `enum` ***Read-only***

State	Description
going_up	connection is initializing
up	connection is active
going_down	connection is about to become inactive
down	connection is inactive

`type` `enum` ***Read-only***

Type	Description
ethernet	FTTH/ethernet
rfc2684	xDSL (unbundled)
pppoatm	xDSL

`media` `enum` ***Read-only***

Media	Description
ftth	FTTH
ethernet	ethernet
xdsl	xDSL
backup_4g	Internet Backup

`ipv4` `string` ***Read-only***

Freebox IPv4 address

NOTE: this field is only available when connection state is up

`ipv6` `string` ***Read-only***

Freebox IPv6 address

NOTE: this field is only available when connection state is up

`rate_up` `int` ***Read-only***

current upload rate in byte/s

`rate_down` `int` ***Read-only***

current download rate in byte/s

`bandwidth_up` `int` ***Read-only***

available upload bandwidth in bit/s

`bandwidth_down` `int` ***Read-only***

available download bandwidth in bit/s

`bytes_up` `int` ***Read-only***

total uploaded bytes since last connection

`bytes_down` `int` ***Read-only***

total downloaded bytes since last connection

`ipv4_port_range` `int[2]` ***Read-only***

Some customers share the same IPv4 and each customer is then assigned a port range. The first value is the first port of the assigned range and the second value is the last port (inclusive).

All `PortForwardingConfig` must use ports in this range to be effective.

Get the current Connection status

`GET /api/v11/connection/`

Returns the current `ConnectionStatus`

Example request:

```
GET /api/v11/connection/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "type": "ethernet",
        "rate_down": 61,
        "bytes_up": 5489542,
        "rate_up": 0,
        "bandwidth_up": 100000000,
        "ipv4": "13.37.42.42",
        "ipv4_port_range": [
            0,
            65535
        ],
        "ipv6": "2a01:e30:d252:a2a0::1",
        "bandwidth_down": 100000000,
        "state": "up",
        "bytes_down": 13332830,
        "media": "ftth"
    }
}
```


Connection configuration

Connection configuration object

`ConnectionConfiguration`

`ping` **bool**
should the Freebox respond to external ping requests

`is_secure_pass` **bool** ***Read-only***
is the admin password secure enough to enable remote access

`remote_access` **bool**
enable/disable HTTP remote access

`remote_access_port` **int**
port number to use for remote HTTP access

`remote_access_min_port` **int** ***Read-only***
This field indicate the minimum possible value for remote_access_port (see `ConnectionStatus` ipv4_port_range)

`remote_access_max_port` **int** ***Read-only***
This field indicate the maximum possible value for remote_access_port (see `ConnectionStatus` ipv4_port_range)

`remote_access_ip` **string** ***Read-only***
IPv4 to use for remote access (can be missing if connection is down)

`api_remote_access` **bool** ***Read-only***
is remote access enabled for apps, or share link

`wol` **bool**
enable/disable Wake-on-lan proxy

`adblock` **bool**
is ads blocking feature enabled

`adblock_not_set` **bool** ***Read-only***
if set to true adblock setting has never been set by the user

`allow_token_request` **bool**
if false, user has disabled new token request. New apps can't request a new token. Apps that already have a token are still allowed

`sip_alg` **enum**

Status	Description
disabled	Fully disable SIP ALG
direct_media	Enable SIP ALG, RTP only allowed between SIP UA
any_media	Enable SIP ALG, RTP allowed between any host (dangerous for untrusted hosts)

Get the current Connection configuration

`GET /api/v11/connection/config/`

Returns the current `ConnectionConfiguration`

Example request:

```
GET /api/v11/connection/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "ping": true,
        "is_secure_pass": false,
        "remote_access_port": 80,
        "remote_access": false,
        "wol": false,
        "adblock": false,
        "adblock_not_set": false,
        "api_remote_access": true,
        "allow_token_request": true,
        "remote_access_ip": "312.13.37.42"
    }
}
```

Update the Connection configuration

`PUT /api/v11/connection/config/`

Updates the `ConnectionConfiguration`

Example request:

```
PUT /api/v11/connection/config/ HTTP/1.1
Host: mafreebox.freebox.fr

{
  "ping": true,
  "wol": false
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "ping": true,
        "is_secure_pass": false,
        "remote_access_port": 80,
        "remote_access": false,
        "wol": false,
        "adblock": false,
        "adblock_not_set": false,
        "api_remote_access": true,
        "allow_token_request": true,
        "remote_access_ip": "312.13.37.42"
    }
}
```

Connection IPv6 configuration

Connection IPv6 configuration object

`ConnectionIpv6Delegation`

`prefix` **string**

IPv6 prefix

next_hop **ipv6**
the next hop for the prefix

ConnectionIpv6Configuration

ipv6_enabled **bool**
is IPv6 enabled

ipv6_firewall **bool**
is IPv6 firewall enabled

ipv6_prefix_firewall **bool**
is IPv6 firewall enabled on secondary prefixes

ipv6ll **string** ***Read-only***
Freebox IPv6 link local address

ipv6_prefix_firewall **bool**
is IPv6 firewall enabled for delegated prefixes

delegations **ConnectionIpv6Delegation[8]**
list of IPv6 delegations

Get the current IPv6 Connection configuration

GET /api/v11/connection/ipv6/config/

Returns the current ConnectionIpv6Configuration

Example request:

```
GET /api/v11/connection/ipv6/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "ipv6_enabled": true,
        "ipv6_firewall": false,
        "ipv6_prefix_firewall": true,
        "delegations": [
            {
                "prefix": "2a01:e30:d252:a2a0::/64",
                "next_hop": ""
            },
            {
                "prefix": "2a01:e30:d252:a2a1::/64",
                "next_hop": ""
            },
            {
                "prefix": "2a01:e30:d252:a2a2::/64",
                "next_hop": ""
            },
            {
                "prefix": "2a01:e30:d252:a2a3::/64",
                "next_hop": ""
            },
            {
                "prefix": "2a01:e30:d252:a2a4::/64",
                "next_hop": ""
            },
            {
                "prefix": "2a01:e30:d252:a2a5::/64",
                "next_hop": ""
            },
            {
                "prefix": "2a01:e30:d252:a2a6::/64",
                "next_hop": ""
            },
            {
                "prefix": "2a01:e30:d252:a2a7::/64",
                "next_hop": ""
            }
        ]
    }
}
```


Update the IPv6 Connection configuration

PUT /api/v11/connection/ipv6/config/

Updates the ConnectionIpv6Configuration

Example request:

```
PUT /api/v11/connection/config/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "delegations": [
      {
         "prefix": "2a01:e30:d252:a2a2::/64",
         "next_hop": "fe80::be30:5bff:feb5:fcc7"
      }
   ]
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "ipv6_enabled": true,
        "ipv6_firewall": false,
        "ipv6_prefix_firewall": false,
        "ipv6ll": "fe80::224:d4ff:acac:ecec",
        "delegations": [
            {
                "prefix": "2a01:e30:d252:a2a0::/64",
                "next_hop": ""
            },
            {
```

```
            "prefix": "2a01:e30:d252:a2a1::/64",
            "next_hop": ""
        },
        {
            "prefix": "2a01:e30:d252:a2a2::/64",
            "next_hop": "fe80::d252:5bff:feb5:fcc7"
        },
        {
            "prefix": "2a01:e30:d252:a2a3::/64",
            "next_hop": ""
        },
        {
            "prefix": "2a01:e30:d252:a2a4::/64",
            "next_hop": ""
        },
        {
            "prefix": "2a01:e30:d252:a2a5::/64",
            "next_hop": ""
        },
        {
            "prefix": "2a01:e30:d252:a2a6::/64",
            "next_hop": ""
        },
        {
            "prefix": "2a01:e30:d252:a2a7::/64",
            "next_hop": ""
        }
    ]
  }
}
```


Connection xDSL status [UNSTABLE]

xDSL status object [UNSTABLE]

`XdslStatus`

`status` `enum` ***Read-only***

Status	Description
down	unsynchronized
training	synchronizing step 1/4
started	synchronizing step 2/4
chan_analysis	synchronizing step 3/4
msg_exchange	synchronizing step 4/4
showtime	Ready
disabled	Disabled

`protocol` `enum` ***Read-only***

Protocol	Description
t1413	T1.413
adsl1_a	ADSL
adsl2_a	ADSL2
adsl2plus_a	ADSL2+
readsl2	ReachDSL
adsl2_m	ADSL2 annex M
adsl2plus_m	ADSL2+ annex M
unknown	Unknown

`modulation` `enum` ***Read-only***

Protocol	Description
adsl	ADSL
vdsl	VDSL

`uptime` `int` ***Read-only***

uptime in seconds

xDSL stats object [UNSTABLE]

`XdslStats`

`maxrate` `int` ***Read-only***

ATM max rate in kbit/s

`rate` `int` ***Read-only***

ATM rate in kbit/s

`snr` `int` ***Read-only***

in dB

`attn` `int` ***Read-only***

in dB

`snr_10` `int` ***Read-only***

in dB/10

`attn_10` `int` ***Read-only***

in dB/10

`fec` `int` ***Read-only***

`crc` `int` ***Read-only***

`hec` `int` ***Read-only***

`es` `int` ***Read-only***

`ses` `int` ***Read-only***

`phyr` `bool` ***Read-only***

`ginp` `bool` ***Read-only***

`nitro` `bool` ***Read-only***

`rxmt` `int` ***Read-only***

only available when phyr is on

`rxmt_corr` `int` ***Read-only***

only available when phyr is on

`rxmt_uncorr` `int` ***Read-only***

only available when phyr is on

`rtx_tx` `int` ***Read-only***

only available when ginp is on

`rtx_c` `int` ***Read-only***

only available when ginp is on

`rtx_uc` `int` ***Read-only***

only available when ginp is on

xDSL infos object [UNSTABLE]

`XdslInfos`

`status` **XdslStatus**

`down` **XdslStats**

`up` **XdslStats**

Get the current xDSL infos

`GET /api/v11/connection/xdsl/`

Returns the current `XdslInfos`

Example request:

```
GET /api/v11/connection/xdsl/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "status": {
            "status": "showtime",
            "protocol": "adsl2plus_a",
            "uptime": 5017,
            "modulation": "adsl"
        },
        "down": {
            "es": 43,
            "phyr": true,
            "attn": 0,
            "snr": 7,
            "nitro": true,
            "rate": 28031,
            "hec": 0,
            "crc": 0,
            "rxmt_uncorr": 0,
            "rxmt_corr": 0,
            "ses": 43,
            "fec": 0,
            "maxrate": 30636,
            "rxmt": 0
        },
        "up": {
            "es": 0,
            "phyr": false,
            "attn": 23,
            "snr": 15,
            "nitro": true,
            "rate": 1022,
            "hec": 0,
            "crc": 0,
            "rxmt_uncorr": 0,
            "rxmt_corr": 0,
            "ses": 0,
            "fec": 0,
            "maxrate": 1022,
            "rxmt": 0
        }
    }
}
```


Connection LTE status [UNSTABLE]

LTE radio band object

`LteRadioBand`

`enabled` `bool`

`bandwidth` `int`

`rsrq` `int`

`rsrp` `int`

`rssi` `int`

`band` `int`

`pci` `int`

LTE radio object

`LteRadio`

`associated` `bool`

`plmn` `int`

`signal_level` `int`

`gcid` `string`

`bands` `[ro]`

`ue_active` `bool`

LTE network object

`LteNetwork`

`pdn_up` `bool`

`has_ipv6` `bool`

ipv6_dns string

ipv6 string

ipv6_netmask string

has_ipv4 bool

ipv4_dns string

ipv4 string

ipv4_netmask string

LTE sim object

LteSim

present bool

pin_locked bool

puk_remaining int

iccid string

puk_locked bool

pin_remaining int

LTE tunnel details object

LteTunnelDetails

connected bool

last_error string

tx_flows_rate int

tx_max_rate int

tx_used_rate int

rx_flows_rate int

rx_max_rate int

rx_used_rate int

LTE tunnel object

LteTunnel

lte LteTunnelDetails

xdsl LteTunnelDetails

LTE configuration object

LteConfiguration

enabled bool

radio LteRadio

state string

network LteNetwork

fsm_state string

sim LteSim

Get the current LTE infos

GET /api/v11/connection/lte/{id}

Returns the current LteConfiguration for the given id. Possible ids are:

- aggregation
- backup

Example request:

```
GET /api/v11/connection/lte/aggregation HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": true,
        "radio": {
            "associated": true,
            "plmn": 20202,
            "signal_level": 5,
            "gcid": "202020202020",
            "bands": [],
            "ue_active": false
        },
        "state": "connected",
        "network": {
            "ipv6_dns": "",
            "ipv6": "2a2a:e0e:beeb:eded::1",
            "ipv4_netmask": "0.0.0.0",
            "has_ipv6": true,
            "ipv4_dns": "0.0.0.0",
            "has_ipv4": alse,
            "pdn_up": true,
            "ipv6_netmask": "ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff00",
            "ipv4": "0.0.0.0"
        },
        "fsm_state": "poll_network",
        "sim": {
            "present": true,
            "pin_locked": alse,
            "puk_remaining": 10,
            "iccid": "1234567890123456789",
            "puk_locked":f alse,
            "pin_remaining": 3
        },
    }
}
```

Get the current xDSL/LTE aggregation infos

`GET /api/v11/connection/aggregation`

Returns the current `LteTunnel`

Example request:

```
GET /api/v11/connection/aggregation HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": true,
        "tunnel": {
            "lte": {
                "tx_flows_rate": 0,
                "connected": true,
                "last_error": "no_error",
                "rx_flows_rate": 0,
                "tx_max_rate": 0,
                "tx_used_rate": 0,
                "rx_max_rate": 0,
                "rx_used_rate": 0
            },
            "xdsl": {
                "tx_flows_rate": 0,
                "connected": true,
                "last_error": "no_error",
                "rx_flows_rate": 0,
                "tx_max_rate": 4428750,
                "tx_used_rate": 134,
                "rx_max_rate": 12502000,
                "rx_used_rate": 120
            }
        }
    }
}
```


Update the xDSL/LTE aggregation configuration

`PUT /api/v11/connection/aggregation`

Updates the `LteConfiguration`

Example request:

```
PUT /api/v11/connection/aggregation/ HTTP/1.1
Host: mafreebox.freebox.fr

{
  "enabled": true
}
```

Connection FTTH status [UNSTABLE]

FTTH status object [UNSTABLE]

`FtthStatus`

- `sfp_present` boolean ***Read-only***
- `sfp_alim_ok` boolean ***Read-only***
- `sfp_has_power_report` boolean ***Read-only***
- `sfp_has_signal` boolean ***Read-only***
- `link` boolean ***Read-only***
- `sfp_serial` string ***Read-only***
- `sfp_model` string ***Read-only***
- `sfp_vendor` string ***Read-only***
- `sfp_vendor` string ***Read-only***
- `sfp_pwr_tx` int ***Read-only***
  scaled by 100 (in dBm)
- `sfp_pwr_rx` int ***Read-only***
  scaled by 100 (in dBm)

Get the current FTTH status

`GET /api/v11/connection/ftth/`

Returns the current `FtthStatus`

Example request:

```
GET /api/v11/connection/ftth/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "sfp_has_power_report": true,
        "sfp_has_signal": false,
        "sfp_model": "SPBD-1250E4H2RDB",
        "sfp_vendor": "DELTA",
        "sfp_pwr_tx": -1172,
        "sfp_pwr_rx": -3698,
        "link": false,
        "sfp_alim_ok": true,
        "sfp_serial": "DE104900000471",
        "sfp_present": true
    }
}
```


Connection DynDNS status

DynDnsProvider status object

`DDNSStatus`

`status` `enum`

Status	Description
disabled	Disabled
ok	Ok
wait	Updating
reqfail	Request failed
authfail	Authentication error
nocredential	Invalid credential
ipinval	Invalid IP
hostinval	Invalid hostname
abuse	Blocked because of abuse
dnserror	DNS error
unavailable	Service unavailable
nowan	Unable to get wan IP
unknown	Unknown

`next_refresh` `int`

next refresh timestamp

`last_refresh` `int`

last refresh timestamp

`next_retry` `int`

next retry timestamp

`last_error` `int`

last error timestamp

Get the status of a DynDNS service

Right now the supported dynamic dns providers are:

- ovh
- dyndns
- noip

`GET /api/v11/connection/ddns/{provider}/status/`

Returns the current DDNSStatus

Example request:

```
GET /api/v11/connection/ddns/dyndns/status/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{

    "success": true,
    "result": {
        "last_error": 1354127350,
        "status": "hostinval",
        "next_refresh": 0,
        "last_refresh": 0,
        "next_retry": 0
    }
}
```


Connection DynDNS configuration

DynDns config object

`DDNSConfig`

`enabled` `bool`

`hostname` `string`

dns name to use to register

`password` `string` ***Write-only***

password to use to register

`user` `string`

username to use to register

Get the config of a DynDNS service

`GET /api/v11/connection/ddns/{provider}/`

Returns the current DDNSConfig

Example request:

```
GET /api/v11/connection/ddns/dyndns/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{

    "success": true,
    "result": {
        "enabled": true,
        "hostname": "test",
        "user": "test"
    }
}
```


Set the config of a DynDNS service

`PUT /api/v11/connection/ddns/{provider}/`

Set the DDNSConfig

Example request:

```
PUT /api/v11/connection/ddns/dyndns/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "enabled": false,
   "user": "test",
   "password": "ssss",
   "hostname": "ttt"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{

    "success": true,
    "result": {
        "enabled": false,
        "hostname": "ttt",
        "user": "test"
    }

}
```

Lan

With the LAN API you get information and modify the Freebox Server network configuration.

Lan Errors

When attempting to access the LAN API, you may encounter the following errors:

error_code	Description
noent	Invalid id
internal_error	Internal error
ioerror	Internal error
inval	Invalid parameter
inval_gateway_ip	Invalid Gateway IP

Lan Config

Lan config has the following attributes:

`LanConfig`

`ip` **string**
: Freebox Server IPv4 address

`name` **string**
: Freebox Server name

`name_dns` **string**
: Freebox Server DNS name

`name_mdns` **string**
: Freebox Server mDNS name

`name_netbios` **string**
: Freebox Server netbios name

`type` **enum**
: The valid LAN modes are:

Type	Description
router	The Freebox acts as a network router
bridge	The Freebox acts as a network bridge

NOTE: in bridge mode, most of Freebox services are disabled. It is recommended to use the router mode, and third party apps should not change this setting

Lan Config API

Get the current Lan configuration

`GET /api/v8/lan/config/`

Returns the current LanConfig

Example request:

```
GET /api/v8/lan/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "name_dns": "freebox-r0ro",
        "name_mdns": "Freebox-r0ro",
        "name": "Freebox r0ro",
        "mode": "router",
        "name_netbios": "Freebox_r0ro",
        "ip": "192.168.1.254"
    }
}
```

Update the current Lan configuration

`PUT /api/v8/lan/config/`

Update the current LanConfig

Example request:

```
PUT /api/v8/lan/config/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "mode":"router",
   "ip":"192.168.69.254",
   "name":"Freebox de r0ro",
   "name_dns":"freebox-de-r0ro",
   "name_mdns":"Freebox-de-r0ro",
```

```
   "name_netbios":"Freebox_de_r0ro"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success":true,
   "result": {
      "name_dns":"freebox-de-r0ro",
      "name_mdns":"Freebox-de-r0ro",
      "name":"Freebox de r0ro",
      "mode":"router",
      "name_netbios":"Freebox_de_r0ro",
      "ip":"192.168.69.254"
   }
}
```

Lan Browser

With the LAN browser API you get information on hosts on the Freebox Server local network.

Errors

When attempting to access the LAN browser API, you may encounter the following errors:

error_code	Description
inval	Invalid parameter
nodev	Invalid interface
nohost	Invalid host id
nomem	Internal error
netdown	Network is down

Lan Browser API

Lan browser API allow you to discover hosts on the local network

Getting the list of browsable LAN interfaces

`GET /api/v8/lan/browser/interfaces/`

Example request:

```
GET /api/v8/lan/browser/interfaces/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   success: true,
   result: [
      {
         name: "pub",
         host_count: 3
      }
   ]
}
```

Lan Host object

Lan Host has the following attributes:

`LanHost`

`id` string ***Read-only***
: Host id (unique on this interface)

`primary_name` string
: Host primary name (chosen from the list of available names, or manually set by user)

`host_type` enum
: When possible, the Freebox will try to guess the host_type, but you can manually override this to the correct value

Possible values are:

source	Description
workstation	Workstation
laptop	Laptop
smartphone	Smartphone
tablet	Tablet
printer	Printer
vg_console	Video game console
television	TV
nas	Nas
ip_camera	IP Camera
ip_phone	IP Phone
freebox_player	Freebox Player
freebox_hd	Freebox HD
freebox_crystal	Freebox Crystal
freebox_mini	Freebox Mini 4k
freebox_delta	Freebox Delta
freebox_one	Freebox One
freebox_wifi	Freebox Wi-Fi Pop
freebox_pop	Freebox Pop
networking_device	Networking device
multimedia_device	Multimedia device
car	Connected car

source	Description
watch	Smartwatch
light	Light
outlet	Connected outlet
appliances	Household appliances
thermostat	Thermostat
shutter	Electric shutter
other	Other

`primary_name_manual` **bool** ***Read-only***
If true the primary name has been set manually

`l2ident[]` **array of** **LanHostL2Ident** ***Read-only***
Layer 2 network id and its type

`vendor_name` **string** ***Read-only***
Host vendor name (from the mac address)

`persistent` **bool**
If true the host is always shown even if it has not been active since the Freebox startup

`reachable` **bool** ***Read-only***
If true the host can receive traffic from the Freebox

`last_time_reachable` **timestamp** ***Read-only***
Last time the host was reached

`active` **bool** ***Read-only***
If true the host sends traffic to the Freebox

`last_activity` **timestamp** ***Read-only***
Last time the host sent traffic

`first_activity` **timestamp** ***Read-only***
First time the host sent traffic, or 0 (Unix Epoch) if it wasn't seen before this field was added.

`names[]` **array of** **LanHostName** ***Read-only***
List of available names, and their source

`l3connectivities[]` **array of** **LanHostL3Connectivity** ***Read-only***
List of available layer 3 network connections

`network_control` **LanHostNetworkControl** ***Read-only***
If device is associated with a profile, contains profile summary.

`info` **dict** ***Read-only***
Contains detailed information that could be gathered about the device.

`LanHostName`

`name` **string** ***Read-only***
Host name

`source` **enum** ***Read-only***
source of the name

`LanHostL2Ident`

`id` **string** ***Read-only***
Layer 2 id

`type` **string** ***Read-only***
Type of layer 2 address

source	Description
dhcp	DHCP
netbios	Netbios
mdns	mDNS hostname
mdns_srv	mDNS service
upnp	UPnP
wsd	WS-Discovery

`LanHostL3Connectivity`

`addr` **string** ***Read-only***
Layer 3 address

`af` **enum** ***Read-only***

af	Description
ipv4	IPv4
ipv6	IPv6

`active` **bool** ***Read-only***
is the connection active

`reachable` **bool** ***Read-only***
is the connection reachable

`last_activity` **timestamp** ***Read-only***
last activity timestamp

`last_time_reachable` **timestamp** ***Read-only***
last reachable timestamp

`model` **string** ***Read-only***
device model if known

`LanHostNetworkControl`

`profile_id` **int** ***Read-only***
Id of profile this device is associated with.

`name` **string** ***Read-only***
Name of profile this device is associated with.

`current_mode` **enum** ***Read-only***
Mode described in Network Control Object

Getting the list of hosts on a given interface

`GET /api/v8/lan/browser/{interface}/`

Returns the list of LanHost on this interface

Example request:

```
GET /api/v8/lan/browser/pub/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "l2ident": {
                "id": "d0:23:db:36:15:aa",
                "type": "mac_address"
            },
            "active": true,
            "id": "ether-d0:23:db:36:15:aa",
            "last_time_reachable": 1360669498,
            "persistent": true,
            "names": [
                {
                    "name": "iPhone-r0ro",
                    "source": "dhcp"
                }
            ],
            "vendor_name": "Apple, Inc.",
            "l3connectivities": [
                {
                    "addr": "192.168.69.20",
                    "active": true,
                    "af": "ipv4",
                    "reachable": true,
                    "last_activity": 1360669498,
                    "last_time_reachable": 1360669498
                }
            ],
            "reachable": true,
            "last_activity": 1360669498,
            "primary_name_manual": true,
            "primary_name": "iPhone r0ro",
            "info": { }
        },
        {
            "l2ident": {
                "id": "00:24:d4:7e:00:4c",
                "type": "mac_address"
            },
            "active": true,
            "id": "ether-00:24:d4:7e:00:4c",
            "last_time_reachable": 1360669491,
            "persistent": false,
            "names": [
                {
                    "name": "Freebox Player",
                    "source": "dhcp"
                }
            ],
            "vendor_name": "FREEBOX SA",
            "l3connectivities": [
                {
                    "addr": "192.168.69.30",
                    "active": true,
                    "af": "ipv4",
                    "reachable": true,
                    "last_activity": 1360669491,
                    "last_time_reachable": 1360669491
                }
            ],
            "reachable": true,
            "last_activity": 1360669491,
            "primary_name_manual": false,
            "primary_name": "Freebox Player",
            "info": {
                "upnp": {
                    "modelName": "Freebox Player",
                    "friendlyName": "Freebox Player",
                    "manufacturer": "Freebox",
                    "service[0]": "urn:dial-multiscreen-org:serviceId:dial",
                    "deviceType": "urn:dial-multiscreen-org:device:dial:1"
                },
                "mdns": {
                    "Service: raop": "192.168.1.91:5000 (tcp)",
                    "Service: hid": "192.168.1.91:24322 (udp)",
                    "Service: airplay": "192.168.1.91:7000 (tcp)",
                    "Service: amzn-alexa": "192.168.1.91 (tcp)"
                },
                "dhcp": {
                    "Host Name": "Freebox Player"
                }
            }
        }
    ]
}
```


Getting an host information

`GET /api/v8/lan/browser/{interface}/{hostid}/`

Returns the requested LanHost properties

Example request:

```
GET /api/v8/lan/browser/pub/ether-00:24:d4:7e:00:4c/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "l2ident": {
```

```
            "id": "00:24:d4:7e:00:4c",
            "type": "mac_address"
        },
        "active": true,
        "id": "ether-00:24:d4:7e:00:4c",
        "last_time_reachable": 1360669611,
        "persistent": false,
        "names": [
            {
                "name": "Freebox Player",
                "source": "dhcp"
            }
        ],
        "vendor_name": "FREEBOX SA",
        "l3connectivities": [
            {
                "addr": "192.168.69.30",
                "active": true,
                "af": "ipv4",
                "reachable": true,
                "last_activity": 1360669611,
                "last_time_reachable": 1360669611
            }
        ],
        "reachable": true,
        "last_activity": 1360669611,
        "primary_name_manual": false,
        "primary_name": "Freebox Player",
        "info": {
            "upnp": {
                "modelName": "Freebox Player",
                "friendlyName": "Freebox Player",
                "manufacturer": "Freebox",
                "service[0]": "urn:dial-multiscreen-org:serviceId:dial",
                "deviceType": "urn:dial-multiscreen-org:device:dial:1"
            },
            "mdns": {
                "Service: raop": "192.168.1.91:5000 (tcp)",
                "Service: hid": "192.168.1.91:24322 (udp)",
                "Service: airplay": "192.168.1.91:7000 (tcp)",
                "Service: amzn-alexa": "192.168.1.91 (tcp)"
            },
            "dhcp": {
                "Host Name": "Freebox Player"
            }
        }
    }
}
```


Updating an host information

`PUT /api/v8/lan/browser/{interface}/{hostid}/`

Update a LanHost properties

Example request:

```
PUT /api/v8/lan/browser/pub/ether-00:24:d4:7e:00:4c/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "id":"ether-00:24:d4:7e:00:4c",
   "primary_name":"Freebox Tv"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "l2ident": {
            "id": "00:24:d4:7e:00:4c",
            "type": "mac_address"
        },
        "active": true,
        "id": "ether-00:24:d4:7e:00:4c",
        "last_time_reachable": 1360669851,
        "persistent": true,
        "names": [
            {
                "name": "Freebox Player",
                "source": "dhcp"
            }
        ],
        "vendor_name": "FREEBOX SA",
        "l3connectivities": [
            {
                "addr": "192.168.69.30",
                "active": true,
                "af": "ipv4",
                "reachable": true,
                "last_activity": 1360669851,
                "last_time_reachable": 1360669851
            }
        ],
        "reachable": true,
        "last_activity": 1360669851,
        "primary_name_manual": true,
        "primary_name": "Freebox Tv",
        "info": {
            "upnp": {
                "modelName": "Freebox Player",
                "friendlyName": "Freebox Player",
                "manufacturer": "Freebox",
                "service[0]": "urn:dial-multiscreen-org:serviceId:dial",
                "deviceType": "urn:dial-multiscreen-org:device:dial:1"
            },
            "mdns": {
                "Service: raop": "192.168.1.91:5000 (tcp)",
                "Service: hid": "192.168.1.91:24322 (udp)",
                "Service: airplay": "192.168.1.91:7000 (tcp)",
                "Service: amzn-alexa": "192.168.1.91 (tcp)"
            },
            "dhcp": {
```

```
                "Host Name": "Freebox Player"
            }
        }
    }
}
```

Getting available lan host types

`GET /api/v8/lan/browser/types/`

Get available LanHost types

Example request:

```
GET /api/v8/lan/browser/types/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
      {
          "icon": "/resources/images/lan/ic_device_computer.png",
          "type": "workstation",
          "name": "Ordinateur",
          "category": "personal_device"
      },
      {
          "icon": "/resources/images/lan/ic_device_printer.png",
          "type": "printer",
          "name": "Imprimante",
          "category": "network"
      },
      ...
    ]
}
```

Wake on LAN

Send Wake ok Lan packet to an host

`POST /api/v8/lan/wol/{interface}/`

Send a wake on LAN packet to the specified host with an optional password

Example request:

```
POST /api/v8/lan/wol/pub/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "mac": "00:24:d4:7e:00:4c",
   "password": ""
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Freeplug

The freeplug API allow you to list the freeplugs on the Freebox network and get stats

Freeplug Errors

When attempting to access the freeplug API, you may encounter the following errors:

error_code	Description
inval	Invalid request
nomem	Internal error
nosta	No freeplug with this id
nopeer	No freeplug with this id

Freeplug Network

FreeplugNetwork has the following attributes:

`FreeplugNetwork`

`id` `string` ***Read-only***
Network unique id

`members[]` `array of` Freeplug ***Read-only***
List of freeplugs member of this network

Freeplug Object

Freeplug has the following attributes:

`Freeplug`

`id` `string` ***Read-only***
Freeplug unique id

`local` `bool` ***Read-only***
if true the Freeplug is connected directly to the Freebox

`net_role` `enum` ***Read-only***
Freeplug network role

Type	Description
sta	Freeplug Station
pco	Freeplug proxy coordinator
cco	Central coordinator

`model` `string` ***Read-only***
Freebox Server netbios name

`eth_port_status` `enum` ***Read-only***

Type	Description
up	The ethernet port is up
down	The ethernet port is down
unknown	The ethernet port state is unknown

`eth_full_duplex` `bool` ***Read-only***
ethernet link is full duplex

`has_network` `bool` ***Read-only***
is connected to the network

`eth_speed` `int` ***Read-only***
ethernet port speed

`inactive` `int` ***Read-only***
seconds since last activity

`net_id` `string` ***Read-only***
network id

`rx_rate` `int` ***Read-only***
rx rate (from the freeplugs to the “cco” freeplug) (in Mb/s) -1 if not available

`tx_rate` `int` ***Read-only***
tx rate (from the “cco” freeplug to the freeplugs) (in Mb/s) -1 if not available

Freeplug API

Get the current Freeplugs networks

`GET /api/v8/freeplug/`

Returns the list of FreeplugNetwork

Example request:

```
GET /api/v8/freeplug/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "id": "c8:f7:b9:83:f5:10:01",
            "members": [
                {
                    "id": "00:24:D4:36:4C:CF",
                    "tx_rate": 148,
                    "eth_port_status": "up",
                    "rx_rate": 148,
                    "net_role": "sta",
                    "inactive": 1,
                    "net_id": "c8:f7:b9:83:f5:10:01",
                    "model": "int6400",
                    "eth_speed": 100,
                    "local": true,
                    "eth_full_duplex": true,
                    "has_network": true
                },
                {
                    "id": "F4:CA:E5:1D:46:AE",
                    "tx_rate": 149,
                    "eth_port_status": "up",
                    "rx_rate": 148,
                    "net_role": "sta",
                    "inactive": 1,
                    "net_id": "c8:f7:b9:83:f5:10:01",
                    "model": "int6400",
                    "eth_speed": 100,
                    "local": true,
                    "eth_full_duplex": true,
                    "has_network": true
                },
                {
                    "id": "00:24:D4:1B:15:D0",
                    "tx_rate": -1,
                    "eth_port_status": "up",
                    "rx_rate": -1,
                    "net_role": "cco",
                    "inactive": 1,
                    "net_id": "c8:f7:b9:83:f5:10:01",
                    "model": "int6400",
                    "eth_speed": 100,
                    "local": false,
                    "eth_full_duplex": true,
                    "has_network": true
                }
            ]
        }
    ]
}
```


Get a particular Freeplug information

`GET /api/v8/freeplug/{id}/`

Returns the list of Freeplug

Example request:

```
GET /api/v8/freeplug/F4:CA:E5:1D:46:AE/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "id": "00:24:D4:36:4C:CF",
        "tx_rate": -1,
```

```
        "eth_port_status": "up",
        "rx_rate": -1,
        "net_role": "sta",
        "inactive": 1,
        "net_id": "c8:f7:b9:83:f5:10:01",
        "model": "int6400",
        "eth_speed": 100,
        "local": true,
        "eth_full_duplex": true,
        "has_network": true
    }
}
```


Reset a Freeplug

`POST /api/v8/freeplug/{id}/reset/`

reset the given Freeplug

Example request:

```
POST /api/v8/freeplug/F4:CA:E5:1D:46:AE/reset/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
   "success":true,
}
```


DHCP

With the DHCP API you configure the Freebox dhcp server, and access its status.

DHCP Errors

When attempting to access the DHCP API, you may encounter the following errors:

error_code	Description
inval	invalid argument
inval_netmask	invalid netmask
inval_ip_range	invalid IP range
inval_ip_range_net	IP range & netmask mismatch
inval_gw_net	gateway & netmask mismatch
exist	already exists
nodev	no such device
noent	no such entry
netdown	network is down
busy	device or resource busy

DHCP Config Object

DHCP config has the following attributes:

`DhcpConfig`

- `enabled` **bool**
  Enable/Disable the DHCP server
- `sticky_assign` **bool**
  Always assign the same IP to a given host
- `gateway` **string** ***Read-only***
  Gateway IP address
- `netmask` **string** ***Read-only***
  Gateway subnet netmask
- `ip_range_start` **string**
  DHCP range start IP
- `ip_range_end` **string**
  DHCP range end IP
- `always_broadcast` **bool**
  Always broadcast DHCP responses
- `ignore_out_of_range_hint` **bool**
  Ignore requested address if it is outside of the DHCP range
- `boot_server` **string**
  Address of the TFTP server used when booting via TFTP.
- `boot_file` **string**
  Boot file to download from the TFTP server when booting via TFTP.
- `dns[]` **array of string**
  List of dns servers to include in DHCP reply
- `options[]` **array of** DhcpOption
  List of dns options to include in DHCP reply

DHCP Option Object

DHCP options have the following attributes

`DhcpOption`

- `id` **string** ***Read-only***
  Option identifier (as defined in RFC 2132)

  The valid option identifiers and types are:

Identifier	Type	Description
time_offset	s32	Time offset
time_server	ip_list	Time server
log_server	ip_list	Log server
cookie_server	ip_list	Cookie server

Identifier	Type	Description
lpr_server	ip_list	LPR server
impress_server	ip_list	Impress server
resource_location_server	ip_list	Resource location server
hostname	string	Hostname
merit_dump_file	string	Merit dump file
swap_server	ip_list	Swap server
root_path	string	Root path
extensions_path	string	Extensions path
ip_fwd	bool	IP forwarding
ip_fwd_non_local	bool	Non-local IP source routing
ip_max_reassembly_size	u16	Maximum IP reassembly size
ip_ttl	u8	Default IP TTL
ip_pmtu_timeout	u32	IP Path MTU timeout
mtu	u16	Interface MTU
local_subnets	bool	All subnets are local
mask_discovery	bool	Perform mask discovery
mask_supplier	bool	Mask supplier
perform_rd	bool	Perform router discovery
rs_address	ip	Router solicitation address
trailer_encapsulation	bool	Trailer encapsulation
arp_cache_timeout	u32	ARP cache timeout
eth_encapsulation	bool	Ethernet encapsulation
tcp_ttl	u8	Default TCP TTL
tcp_keepalive_interval	u32	TCP keepalive interval
tcp_keepalive_garbage	bool	TCP keepalive garbage
nis_domain	string	NIS domain
nis_server	ip_list	NIS server
ntp_server	ip_list	NTP server
vendor_specific	hexstring	Vendor specific information
nis_plus_domain	string	NIS+ domain
nis_plus_server	ip_list	NIS+ server
tftp_server_name	string	TFTP server name
bootfile_name	string	Bootfile name
mobile_ip_agent	ip_list	Mobile IP home agent
smtp_server	ip_list	SMTP server
pop3_server	ip_list	POP3 server
nntp_server	ip_list	NNTP server
www_server	ip_list	Default WWW server
finger_server	ip_list	Default Finger server
irc_server	ip_list	Default IRC server
streettalk_server	ip_list	StreetTalk server
stda_server	ip_list	StreetTalk directory assistance server
slp_directory_agent	ip_list	SLP directory agent
slp_service_scope	hexstring	SLP service scope
nds_servers	ip_list	NDS servers
nds_tree_name	string	NDS tree name
nds_context	string	NDS context
ldap_servers	ip_list	LDAP servers
timezone_posix	string	Timezone POSIX
timezone_database	string	Timezone database
name_service	hexstring	Name service
domain_search	hexstring	Domain search
classless_static_route	hexstring	Classless static route
capwap_ac	ip_list	CAPWAP access controller
tftp_server_address	ip_list	TFTP server address

`val` **string**

The value sent by the DHCP server when this option is requested by the client.

The formats depend on the option type:

- ip: A single IPv4 address (as described in RFC 791)
- ip_list: A comma-separated list of IPv4 addresses
- string: A string of ASCII characters
- hexstring: A string of ASCII hexadecimal characters [0-9a-fA-F] representing a binary value (example: C0A801FE)
- bool: one of ['true', 'false', '1', '0']
- s8, s16, s32: An n-bit signed integer value
- u8, u16, u32: An n-bit unsigned integer value

DHCP Configuration API

Get the current DHCP configuration

`GET /api/v8/dhcp/config/`

Returns the current `DhcpConfig`

Example request:

```
GET /api/v8/dhcp/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": true,
        "gateway": "192.168.1.254",
        "sticky_assign": true,
        "ip_range_end": "192.168.1.50",
        "netmask": "255.255.255.0",
        "boot_server": "",
        "boot_file": "",
        "dns": [
            "192.168.1.254",
            "",
            "",
            "",
            ""
        ],
        "always_broadcast": false,
        "ip_range_start": "192.168.1.2",
        "options": [
            {
               "id" : "ip_fwd",
               "val" : "true"
            },
            {
               "id" : "tcp_ttl",
               "val" : "64"
            },
            {
               "id" : "ntp_server",
               "val" : "192.168.1.38, 192.168.1.42"
            },
            {
               "id" : "log_server",
               "val" : "192.168.1.38"
            }
        ]
    }
}
```


Update the current DHCP configuration

`PUT /api/v8/dhcp/config/`

Update the current `DhcpConfig`

Example request:

```
PUT /api/v8/dhcp/config/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "enabled": false,
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": false,
        "gateway": "192.168.1.254",
        "sticky_assign": true,
        "ip_range_end": "192.168.1.50",
        "netmask": "255.255.255.0",
        "dns": [
            "192.168.1.254",
            "",
            "",
            "",
            ""
        ],
        "always_broadcast": false,
        "ip_range_start": "192.168.1.2",
        "options": [
            {
               "id" : "ip_fwd",
               "val" : "true"
            },
            {
               "id" : "tcp_ttl",
               "val" : "64"
            },
            {
               "id" : "ntp_server",
               "val" : "192.168.1.38, 192.168.1.42"
            },
            {
               "id" : "log_server",
               "val" : "192.168.1.38"
            }
        ]
    }
}
```


DHCP Static Lease Object

DHCP static lease have the following attributes

`DhcpStaticLease`

- `id` **string**
  DHCP static lease object id
- `mac` **string**
  Host mac address

`comment` **string**
an optional comment

`hostname` **string** ***Read-only***
hostname matching the mac address

`ip` **string**
IPv4 to assign to the host

`host` **LanHost** ***Read-only***
LAN host information from LAN browser (refer to `LanHost` documentation)

`options[]` **array of DhcpOption**
List of dns options to include in DHCP reply

DHCP Static Lease API

Get the list of DHCP static leases

You can get the list of `DhcpStaticLease` using this API

`GET /api/v8/dhcp/static_lease/`

Example request:

```
GET /api/v8/dhcp/static_lease/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "mac": "00:DE:AD:B0:0B:55",
            "comment": "",
            "hostname": "Pc de r0ro",
            "id": "00:DE:AD:B0:0B:55",
            "host": {
               [ ... ]
            },
            "ip": "192.168.1.1",
            "options": [
                {
                   "id" : "log_server",
                   "val" : "192.168.1.38"
                }
            ]
        },
        {
            "mac": "00:DE:AD:B0:0B:69",
            "comment": "",
            "hostname": "Imprimante",
            "id": "00:DE:AD:B0:0B:69",
            "host": {
               [ ... ]
            },
            "ip": "192.168.1.2",
            options: []
        }
    ]
}
```


Get a given DHCP static lease

You can get a specific `DhcpStaticLease` with its id

`GET /api/v8/dhcp/static_lease/{id}`

Example request:

```
GET /api/v8/dhcp/static_lease/00:DE:AD:B0:0B:55 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
         "mac": "00:DE:AD:B0:0B:55",
         "comment": "",
         "hostname": "Pc de r0ro",
         "id": "00:DE:AD:B0:0B:55",
         "host": {
            [ ... ]
         },
         "ip": "192.168.1.1",
         "options": [
             {
                "id" : "log_server",
                "val" : "192.168.1.38"
             }
         ]
     }
}
```


Update DHCP static lease

You can update a `DhcpStaticLease` with this method

`PUT /api/v8/dhcp/static_lease/{id}`

Example request:

```
PUT /api/v8/dhcp/static_lease/00:DE:AD:B0:0B:55 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "comment": "Mon PC"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
         "mac": "00:DE:AD:B0:0B:55",
         "comment": "Mon PC",
         "hostname": "Pc de r0ro",
         "id": "00:DE:AD:B0:0B:55",
         "host": {
            [ ... ]
         },
         "ip": "192.168.1.1",
         "options": [
             {
                "id" : "log_server",
                "val" : "192.168.1.38"
             }
         ]
     }
}
```


Delete a DHCP static lease

Deletes the DhcpStaticLease with this id

`DELETE /api/v8/dhcp/static_lease/{id}`

Example request:

```
DELETE /api/v8/dhcp/static_lease/00:DE:AD:B0:0B:55 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
}
```


Add a DHCP static lease

`POST /api/v8/dhcp/static_lease/`

Example request:

```
POST /api/v8/dhcp/static_lease/ HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
   "ip": "192.168.1.222",
   "mac": "00:00:00:11:11:11"
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "mac": "00:00:00:11:11:11",
        "comment": "",
        "hostname": "00:00:00:11:11:11",
        "id": "00:00:00:11:11:11",
        "ip": "192.168.1.222",
        "options": []
    }
}
```


DHCP Dynamic Lease Object

DHCP dynamic lease have the following attributes

`DhcpDynamicLease`

- `mac` string ***Read-only***
  Host mac address
- `hostname` string ***Read-only***
  hostname matching the mac address
- `ip` string ***Read-only***
  IPv4 assigned to the host
- `lease_remaining` int ***Read-only***
  time left before lease needs to be refreshed
- `assign_time` timestamp ***Read-only***
  timestamp of the lease first assignment
- `refresh_time` timestamp ***Read-only***
  timestamp of the last lease refresh
- `is_static` bool ***Read-only***
  is the lease static
- `host` LanHost ***Read-only***
  LAN host information from LAN browser (refer to `LanHost` documentation)

Get the list of DHCP dynamic leases

You can get the list of DhcpDynamicLease using this API

`GET /api/v8/dhcp/dynamic_lease/`

Example request:

```
GET /api/v8/dhcp/dynamic_lease/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "mac": "13:37:00:00:01:03",
            "host": {
               "l2ident": {
               "id": "13:37:00:00:01:03",
```

```
                "type": "mac_address"
            },
            "active": true,
            "id": "ether-13:37:00:00:01:03",
            "last_time_reachable": 1555555555,
            "persistent": false,
            "names": [],
            "vendor_name": "",
            "host_type": "",
            "primary_name": "",
            "l3connectivities": [
               {
                  "addr": "192.168.1.1",
                  "active": true,
                  "reachable": true,
                  "last_activity": 1555555555,
                  "af": "ipv4",
                  "last_time_reachable": 1555555555
               },
               {
                  "addr": "fe80::ffff:3333:eeee:eee",
                  "active": false,
                  "reachable": false,
                  "last_activity": 1555585108,
                  "af": "ipv6",
                  "last_time_reachable": 1555585103
               }
            ],
            "reachable": true,
            "last_activity": 1555555555,
            "primary_name_manual": false,
            "interface": "pub"
                         }
         "refresh_time": 1555555555,
         "hostname": "android r0ro",
         "assign_time": 1555555555,
         "lease_remaining": 123456,
         "is_static": false,
         "ip": "192.168.1.22",
         "options": [
             {
                "id" : "ip_fwd",
                "val" : "true"
             },
             {
                "id" : "tcp_ttl",
                "val" : "64"
             },
             {
                "id" : "ntp_server",
                "val" : "192.168.1.38, 192.168.1.42"
             },
             {
                "id" : "log_server",
                "val" : "192.168.1.38"
             }
         ]
      }
   ]
}
```


DHCPv6

With the DHCPv6 API you configure the Freebox DHCPv6 server, and access its status.

DHCPv6 Errors

When attempting to access the DHCPv6 API, you may encounter the following errors:

error_code	Description
inval	invalid parameter
noent	no such entry
nospc	too many entries
exist	already exists
conflict	conflict with another rule
nomem	internal error

DHCPv6 Config Object

DHCPv6 config has the following attributes:

DHCPv6Config

enabled **bool**

Enable/Disable the DHCPv6 server

NOTE: on some Android devices, enabling the DHCPv6 server may cause IPv6 to stop working on those devices

use_custom_dns **bool**

if set to true, the user provided IPv6 dns servers will be used instead of Free default IPv6 dns servers

NOTE: even if DHCPv6 server is disabled the custom dns can be used to replace Free dns in RA RDNSS

dns[] **array of ipv6** ***Read-only***

list of ipv6 dns servers to use instead of Free dns servers in case use_custom_dns is set to true

DHCPv6 Configuration API

Get the current DHCPv6 configuration

GET /api/v8/dhcpv6/config/

Returns the current DHCPv6Config

Example request:

```
GET /api/v8/dhcpv6/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success": true,
            "result": {
            "enabled": true,
            "use_custom_dns": false,
            "dns": [
                  "2620:0:ccc::a",
                  "2620:0:ccc::1"
            ]
      }
}
```


Update the current DHCPv6 configuration

`PUT /api/v8/dhcpv6/config/`

Update the current DHCPv6Config

Example request:


```
PUT /api/v8/dhcpv6/config/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "use_custom_dns": true,
}
```


Example response:


```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success": true,
            "result": {
            "enabled": true,
            "use_custom_dns": true,
            "dns": [
                  "2620:0:ccc::a",
                  "2620:0:ccc::1"
            ]
      }
}
```


Ftp

The FTP API allow you to control the Freebox ftp server settings

Ftp Errors

When attempting to access the FTP API, you may encounter the following errors:

error_code	Description
internal_error	Internal error
weak_password	Password is too weak for remote access

Ftp Config

FtpConfig has the following attributes:

`FtpConfig`

- `enabled` **bool**
  is the FTP server enabled
- `allow_anonymous` **bool**
  can anonymous user log in
- `allow_anonymous_write` **bool**
  can anonymous user write data
- `username` **string** ***Read-only***
  default user name to use. Cannot be changed
- `password` **string** ***Write-only***
  user password
- `allow_remote_access` **bool**
  enable ftp server remote access
  NOTE: to be able to enable the remote access the password must be strong enough
- `weak_password` **bool** ***Read-only***
  is the ftp password weak (in this case remote access is disabled)
- `port_ctrl` **int**
  ftp control port to use for remote access
- `port_data` **int**
  ftp data port to use for remote access
- `remote_domain` **string**
  domain name to use for remote access

Ftp config API

Get the current Ftp configuration

`GET /api/v8/ftp/config/`

Get the FtpConfig

Example request:


```
GET /api/v8/ftp/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```


Example response:


```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": false,
        "allow_anonymous": false,
        "allow_remote_access": false,
        "port_ctrl": 3615,
        "port_data": 1337,
        "weak_password": true,
```

```
        "allow_anonymous_write": false
    }
}
```

Update the FTP configuration

`PUT /api/v8/ftp/config/`

Update the FtpConfig

Example request:

```
PUT /api/v8/ftp/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "enabled": true
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": true,
        "allow_anonymous": false,
        "allow_anonymous_write": false
    }
}
```

TFTP

The TFTP API allow you to control the Freebox tftp server settings

TFTP Errors

When attempting to access the TFTP API, you may encounter the following errors:

error_code	Description
absolute	The path must be absolute

TFTP Config

TftpConfig has the following attributes:

`TftpConfig`

`enabled` **bool**
: is the TFTP server enabled

`root` **string**
: is the base64 encoded absolute path to the root directory exposed by the server. This path points to a folder inside the storage device (My Freebox).

TFTP Config API

Get the current TFTP configuration

`GET /api/v15/tftp/config/`

Get the TftpConfig

Example request:

```
GET /api/v15/tftp/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": false,
        "root": "/ssd2"
    }
}
```

Update the TFTP configuration

`PUT /api/latest/tftp/config/`

Update the TftpConfig

Example request:

```
PUT /api/v15/tftp/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "enabled": true
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": true,
        "root": "/ssd2"
    }
}
```

NAT

With the nat API you control port forwarding on your network

NAT Errors

When attempting to access the LAN API, you may encounter the following errors:

error_code	Description
noent	Invalid id
internal_error	Internal error

error_code	Description
exist	Conflict with an existing redirection

Dmz Config

Dmz config has the following attributes:

`DmzConfig`

`ip` `string`
dmz host IP

`enabled` `bool`
is dmz enabled

Dmz Config API

Get the current Dmz configuration

`GET /api/v8/fw/dmz/`

Returns the current `DmzConfig`

Example request:

```
GET /api/v8/fw/dmz/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": false,
        "ip": ""
    }
}
```

Update the current Dmz configuration

`PUT /api/v8/fw/dmz/`

Update the current `LanConfig`

Example request:

```
PUT /api/v8/lan/config/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "enabled": true,
   "ip": "192.168.1.42"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": true,
        "ip": "192.168.1.42"
    }
}
```

Port Forwarding

Port Forwarding Config

Port forwarding config has the following attributes:

`PortForwardingConfig`

`id` `int`
forwarding id

`enabled` `bool`
is forwarding enabled

`ip_proto` `enum`

ip_proto	Description
tcp	TCP
udp	UDP

`wan_port_start` `string`
forwarding range start

`wan_port_end` `int`
forwarding range end

`lan_ip` `string`
forwarding target on LAN

`lan_port` `int`
forwarding target start port on LAN, (last port is lan_port + wan_port_end - wan_port_start)

`hostname` `string` ***Read-only***
forwarding target host name

`host` `LanHost` ***Read-only***
forwarding target host information (see: `LanHost`)

`src_ip` `string`
if src_ip == 0.0.0.0 this rule will apply to any src ip otherwise it will only apply to the specified ip address

`comment` `string`
comment

Port Forwarding API

Getting the list of port forwarding

`GET /api/v8/fw/redir/`

Example request:

```
GET /api/v8/fw/redir/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "enabled": true,
            "comment": "",
            "id": 1,
            "host": {
                [ ... ]
            },
            "hostname": "android-c5fe44a2c27be1e2",
            "lan_port": 69,
            "wan_port_end": 69,
            "wan_port_start": 69,
            "lan_ip": "192.168.1.22",
            "ip_proto": "tcp",
            "src_ip": "8.8.8.8"
        },
        {
            "enabled": true,
            "comment": "",
            "id": 2,
            "host": {
                [ ... ]
            },
            "hostname": "android-c5fe44a2c27be1e2",
            "lan_port": 1337,
            "wan_port_end": 1340,
            "wan_port_start": 1337,
            "lan_ip": "192.168.1.22",
            "ip_proto": "udp",
            "src_ip": "0.0.0.0"
        }
    ]
}
```


Getting a specific port forwarding

GET /api/v8/fw/redir/{redir_id}

Returns the requested PortForwardingConfig properties

Example request:

```
GET /api/v8/fw/redir/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": true,
        "comment": "",
        "id": 1,
        "host": {
            [ ... ]
        },
        "hostname": "android-c5fe44a2c27be1e2",
        "lan_port": 69,
        "wan_port_end": 69,
        "wan_port_start": 69,
        "lan_ip": "192.168.1.22",
        "ip_proto": "tcp",
        "src_ip": "0.0.0.0"
    }
}
```


Updating a port forwarding

PUT /api/v8/fw/redir/{redir_id}

Update a PortForwardingConfig properties

Example request:

```
PUT /api/v8/fw/redir/1 HTTP/1.1
Host: mafreebox.freebox.fr

{
  "enabled": false
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": false,
        "comment": "",
        "id": 1,
        "host": {
            [ ... ]
        },
        "hostname": "android-c5fe44a2c27be1e2",
        "lan_port": 69,
        "wan_port_end": 69,
        "wan_port_start": 69,
        "lan_ip": "192.168.1.22",
        "ip_proto": "tcp",
        "src_ip": "0.0.0.0"
    }
}
```

Add a port forwarding

`POST /api/v8/fw/redir/`

Create a PortForwardingConfig

Example request:

```
POST /api/v8/fw/redir/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
    "enabled": true,
    "comment": "test",
    "lan_port": 4242,
    "wan_port_end": 4242,
    "wan_port_start": 4242,
    "lan_ip": "192.168.1.42",
    "ip_proto": "tcp",
    "src_ip": "0.0.0.0"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": true,
        "comment": "test",
        "id": 3,
        "host": {
            [ ... ]
        },
        "hostname": "Mac-mini-de-Romain",
        "lan_port": 4242,
        "wan_port_end": 4242,
        "wan_port_start": 4242,
        "lan_ip": "192.168.1.42",
        "ip_proto": "tcp",
        "src_ip": "0.0.0.0"
    }
}
```

Delete a port forwarding

`DELETE /api/v8/fw/redir/{redir_id}`

Delete a PortForwardingConfig

Example request:

```
DELETE /api/v8/fw/redir/3 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Incoming port configuration

Some services hosted on the Freebox Server need to listen to public ip address port. Incoming port api allow to enable/disable incoming port binding, and select the bind port to prevent conflit with your own nat port forwarding rules.

NOTE: you can't add or remove incoming ports, this ports are managed by Freebox services.

NOTE: in case of conflict with a nat port forwarding rule, this rule will have a higher priority and override the port forwarding rule.

Incoming port Config

Incoming port config has the following attributes:

`IncomingPortConfig`

`id` `string` ***Read-only***

incoming port id

id	Description
http	http port for remote access to Freebox OS
https	https port for tls remote access to Freebox OS
bittorrent-main	main bittorrent port for Freebox downloader
bittorrent-dht	bittorrent port for DHT
openvpn_routed	routed openvpn port
openvpn_bridge	bridged openvpn port
ipsec_ike	ipsec ikev2 vpn port
ipsec_nat	ipsec nat vpn port
pptp	pptp vpn server port
ftp	ftp control port for FTP remote access
ftp_pasv	ftp data port for FTP remote access

`enabled` `bool`

is the port binding allowed

`active` `bool` ***Read-only***

is the port binding currently active

`type` `enum` ***Read-only***

ip_proto	Description
tcp	TCP
udp	UDP
tcp_udp	both TCP and UDP

`in_port` `int`

binding port

`netns` `string` ***Read-only***

network namespace. The service may be running on a different namespace (for instance if the service uses the vpn client).

`in_port` `int`
binding port

`min_port` `int` ***Read-only***
This field indicate the minimum possible value for in_port (see `ConnectionStatus` ipv4_port_range)

`max_port` `int` ***Read-only***
This field indicate the maximum possible value for in_port (see `ConnectionStatus` ipv4_port_range)

`readonly` `bool` ***Read-only***
If set to true, the in_port field cannot be changed because of the underlying protocol does not allow it

Incoming port API

Getting the list of incoming ports

`GET /api/v8/fw/incoming/`

Example request:

```
GET /api/v8/fw/incoming/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "enabled": false,
            "type": "tcp",
            "in_port": 80,
            "id": "http",
            "netns": "init",
            "max_port": 65535,
            "min_port": 0
        },
        {
            "enabled": true,
            "type": "tcp",
            "in_port": 17591,
            "id": "bittorrent-main",
            "netns": "vpn",
            "max_port": 65535,
            "min_port": 0
        },
        {
            "enabled": true,
            "type": "udp",
            "in_port": 28946,
            "id": "bittorrent-dht",
            "netns": "vpn",
            "max_port": 65535,
            "min_port": 0
        }
    ]
}
```


Getting a specific incoming port

`GET /api/v8/fw/incoming/{port_id}`

Returns the requested `IncomingPortConfig` properties

Example request:

```
GET /api/v8/fw/incoming/bittorrent-main HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": true,
        "type": "tcp",
        "in_port": 17591,
        "id": "bittorrent-main",
        "netns": "vpn",
        "max_port": 65535,
        "min_port": 0
    }
}
```


Updating an incoming port

`PUT /api/v8/fw/incoming/{port_id}`

Update a `IncomingPortConfig` properties

Example request:

```
PUT /api/v8/lan/fw/incoming/bittorrent-main HTTP/1.1
Host: mafreebox.freebox.fr

{
  "in_port": 3615
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": true,
        "type": "tcp",
        "in_port": 3615,
        "id": "bittorrent-main",
        "netns": "vpn",
        "max_port": 65535,
        "min_port": 0
```

```
    }
}
```

UPnP IGD

The UPnP IGD API allow you to control the settings of the Universal Plug n' Play Internet Gateway Device service. This service allow hosts on your local network to manage nat redirections.

UPnP IGD Errors

When attempting to access the UPnP IGD API, you may encounter the following errors:

error_code	Description
disabled	the service is disabled
noent	invalid rule id

UPnP IGD Config

UPnPIGDConfig has the following attributes:

UPnPIGDConfig

- **enabled** bool
  is the UPnP IGD service enabled
- **version** int
  UPnP IGD protocol version Supported values are 1 / 2

UPnP IGD config API

Get the current UPnP IGD configuration

GET /api/v8/upnpigd/config/

Get the UPnPIGDConfig

Example request:

```
GET /api/v8/upnpigd/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": false,
        "version": 1
    }
}
```

Update the UPnP IGD configuration

PUT /api/v8/upnpigd/config/

Update the UPnPIGDConfig

Example request:

```
PUT /api/v8/upnpigd/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "enabled": true,
   "version": 2
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": true,
        "version": 2
    }
}
```

UPnP IGD Redirection

UPnPRedir has the following attributes:

UPnPRedir

- **id** string ***Read-only***
  the redirection id
- **enabled** bool ***Read-only***
  is the redirection enabled
- **ext_src_ip** string ***Read-only***
  source IP
- **ext_port** int ***Read-only***
  external port
- **int_ip** string ***Read-only***
  the target IP on your LAN
- **int_port** int ***Read-only***
  the target port on your LAN
- **proto** string ***Read-only***
  the IP protocol to redirect
- **desc** string ***Read-only***
  a description
- **remaining** int ***Read-only***
  seconds remaining before redirection expire
- **host** LanHost ***Read-only***
  lan host if available

UPnP IGD Redirection API

Get the list of current redirection

`GET /api/v8/upnpigd/redir/`

Get the list of UPnPRedir redirections

Example request:

```
GET /api/v8/upnpigd/redir/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "enabled": true,
            "proto": "udp",
            "id": "0.0.0.0-53644-udp",
            "desc": "iC53644",
            "remaining": 0,
            "ext_src_ip": "0.0.0.0",
            "int_port": 16402,
            "int_ip": "192.168.1.44",
            "ext_port": 53644
        }
    ]
}
```


Delete a redirection

`DELETE /api/v8/upnpigd/redir/{id}`

Deletes the given UPnPRedir

Example request:

```
GET /api/v8/upnpigd/redir/0.0.0.0-53644-udp HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


LCD

The lcd API allow you to control the Freebox lcd screen settings

LCD Errors

When attempting to access the lcd API, you may encounter the following errors:

error_code	Description
inval	Invalid parameters
no_panel	No screen detected
setup	Unable to setup screen
notsup	Operation is not supported

LCD Config

LcdConfig has the following attributes:

`LcdConfig`

- `brightness` **int**
  the screen brightness (range from 0 to 100)
- `orientation_forced` **bool**
  is the screen orientation forced
- `orientation` **int**
  the screen orientation angle
- `hide_wifi_key` **bool**
  hide wifi key information (including qrcode) - optional
- `led_strip_enabled` **bool**
  enable/disable led strip brightness - optional
- `led_strip_brightness` **int**
  led strip brightness (range from 0 to 100) - optional
- `led_strip_animation` **enum**
  led strip animation - optional
- `available_led_strip_animations[]` **array of enum** ***Read-only***
  array containing what LED strip animations can be configured
- `hide_status_led` **bool**
  hide status LED (on supported Freebox models) - optional

LCD config API

Get the current LCD configuration

`GET /api/v8/lcd/config/`

Get the LcdConfig

Example request:

```
GET /api/v8/lcd/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "brightness": 100,
        "orientation": 0,
```

```
        "orientation_forced": false,
        "hide_wifi_key": false,
        "hide_led": false
    }
}
```


Update the lcd configuration

`PUT /api/v8/lcd/config/`

Update the `LcdConfig`

Example request:

```
PUT /api/v8/lcd/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
  "brightness": 50
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "brightness": 50,
        "orientation": 0,
        "orientation_forced": false
        "hide_wifi_key": false,
        "hide_led": false
    }
}
```


Ledstrip

This API allows ledstrip scheduling on boxes that have has_led_strip to true in their `SystemConfig` information.

Ledstrip errors

When attempting to access the ledstrip API, you may encounter the following errors

error_code	Description
inval	Invalid parameters

Ledstrip planning object

Ledstrip planning object have the following properties:

`LedstripPlanning`

`use_planning` **bool**
: is the planning enabled

`planning_mode` **enum**
: current planning mode

Type	Description
ledstrip_off	ledstrip disabled

`resolution` **int** ***Read-only***
: planning resolution (number of slots per day)

`mapping[]` **array of bool**
: mapping for planning : true or false

mapping[0] is monday at 0:0

mapping[7 * resolution - 1] is sunday last slot

(each slot has a duration of 60 * 24 / resolution minutes)

The boolean value indicates whether the planning is in effect (i.e: ledstrip disabled)

Ledstrip status object

Ledstrip status object has the following properties:

`LedstripStatus`

`use_planning` **bool** ***Read-only***
: is the planning enabled

`next_change` **timestamp** ***Read-only***
: timestamp of the scheduled next change, according to planning

Ledstrip API

Get ledstrip status

`GET /api/v15/ledstrip/status`

Returns the `Ledstrip status object`

Example request:

```
GET /api/v15/ledstrip/status HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true,
  "result": {
    "use_planning": true,
    "next_change": 1651135474996,
  }
}
```


Get ledstrip planning

Get the `LedstripPlanning`

Example request:

```
GET /api/v15/ledstrip/planning/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true,
  "result": {
    "use_planning": false,
    "planning_mode": "ledstrip_off",
    "mapping": [
      false,
      false,
      false,
      false,

      [ ... ]

      false,
      false,
      false,
      false
    ],
    "resolution": 48
  }
}
```


Update ledstrip planning

`PUT /api/v15/ledstrip/planning`

Example request:

```
PUT /api/v15/ledstrip/planning/ HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
  "use_planning": true,
  "planning_mode": "ledstrip_off",
  "mapping": [
    false,
    false,
    false,
    false,

    [ ... ],

    false,
    false,
    false,
    false
  ],
  "resolution": 48
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true,
  "result": {
    "use_planning": false,
    "planning_mode": "ledstrip_off",
    "mapping": [
      false,
      false,
      false,
      false,
      false,

      [ ... ]

      false,
      false,
      false,
      false
    ],
    "resolution": 48
  }
}
```


Network Share

The network share API allow you to control the file sharing services running on the Freebox.

Network Share Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
invalid_workgroup_name	Invalid workgroup name
invalid_logon_user	Invalid samba user name
invalid_logon_password	Invalid samba user password
invalid_afp_login_name	Invalid AFP user name
invalid_afp_login_password	Invalid AFP user password

Samba Config

SambaConfig has the following attributes:

`SambaConfig`

- `file_share_enabled` `bool`
  is file sharing enabled
- `print_share_enabled` `bool`
  is printer sharing enabled
- `logon_enabled` `bool`
  is login/password required to access shares
- `logon_user` `string`
  samba user name

`logon_password` `string` ***Write-only***
samba user password

`workgroup` `string`
name of the workgroup

`smbv2_enabled` `bool`
Set to true to enable SMBv2/v3

Samba config API

Get the current Samba configuration

`GET /api/v8/netshare/samba/`

Get the SambaConfig

Example request:

```
GET /api/v8/netshare/samba/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "workgroup": "WORKGROUP",
        "print_share_enabled": true,
        "file_share_enabled": true,
        "logon_enabled": false,
        "logon_user": "freebox"
    }
}
```

Update the Samba configuration

`PUT /api/v8/netshare/samba/`

Update the SambaConfig

Example request:

```
PUT /api/v8/netshare/samba/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "print_share_enabled": false
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "workgroup": "WORKGROUP",
        "print_share_enabled": false,
        "file_share_enabled": true,
        "logon_enabled": false,
        "logon_user": "freebox"
    }
}
```

Afp Config

AfpConfig has the following attributes:

`AfpConfig`

`enabled` `bool`
is afp service enabled

`guest_allow` `bool`
allow guest to access shared files

`server_type` `enum`
Afp server type (to display proper icon) in MacOS
valid server types are:

server_type	
powerbook	
powermac	
macmini	
imac	
macbook	
macbookpro	
macbookair	
macpro	
appletv	
airport	
xserve	

`login_name` `string`
Afp user name

`login_password` `string` ***Write-only***
Afp user password

Afp config API

Get the current Afp configuration

`GET /api/v8/netshare/afp/`

Get the AfpConfig

Example request:

```
GET /api/v8/netshare/afp/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": false,
        "guest_allow": true,
        "login_name": "freebox",
        "server_type": "airport"
    }
}
```

Update the Afp configuration

`PUT /api/v8/netshare/afp/`

Update the AfpConfig

Example request:

```
PUT /api/v8/netshare/afp/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "guest_allow": false
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": false,
        "guest_allow": false,
        "login_name": "freebox",
        "server_type": "airport"
    }
}
```

UPnP AV

The UPnP AV API allow you to control the settings of the Freebox UPnP AV service.

UPnP AV Errors

When attempting to access the UPnP AV API, you may encounter the following errors:

error_code	Description
internal_error	internal error

UPnP AV Config

UPnPAVConfig has the following attributes:

`UPnPAVConfig`

- `enabled` **bool**
  is the UPnP AV service enabled

UPnP AV config API

Get the current UPnP AV configuration

`GET /api/v8/upnpav/config/`

Get the UPnPAVConfig

Example request:

```
GET /api/v8/upnpav/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": true
    }
}
```

Update the UPnP AV configuration

`PUT /api/v8/upnpav/config/`

Update the UPnPAVConfig

Example request:

```
PUT /api/v8/upnpigd/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "enabled": false
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": false
    }
}
```

Switch

The Switch API allow you to control the settings of the Freebox integrated switch.

Switch Errors

When attempting to access the switch API, you may encounter the following errors:

error_code	Description
bad_port	invalid port number
bad_speed	unable to set port speed
bad_link	unable to set port link mode
bad_mac_entry_type	invalid mac entry type

Switch Port Status Object

SwitchPortStatus has the following attributes:

`SwitchPortStatus`

`id` `int` ***Read-only***
switch port id

`link` `enum` ***Read-only***

link	Description
up	port is up
down	port is down

`duplex` `enum`

duplex	Description
half	force in half duplex mode
full	force in full duplex mode

`speed` `enum`

duplex	Description
10	10Base-T
100	100Base-TX
1000	1000Base-T

`mode` `string` ***Read-only***
display form of speed and duplex mode

`mac_list[]` `array of object` ***Read-only***
list of { mac, name } of hosts connected to this port

Switch Port Configuration Object

SwitchPortConfig has the following attributes:

`SwitchPortConfig`

`id` `int` ***Read-only***
switch port id

`duplex` `enum`

duplex	Description
auto	auto negotiate duplex mode
half	force in half duplex mode
full	force in full duplex mode

`speed` `enum`

duplex	Description
auto	auto negotiate speed
10	10Base-T
100	100Base-TX
1000	1000Base-T

Switch Port Stats Object [UNSTABLE]

SwitchPortStats has the following attributes:

`SwitchPortStats`

`rx_bad_bytes` `int` ***Read-only***

`rx_broadcast_packets` `int` ***Read-only***

`rx_bytes_rate` `int` ***Read-only***

`rx_err_packets` `int` ***Read-only***

`rx_fcs_packets` `int` ***Read-only***

`rx_fragments_packets` `int` ***Read-only***

`rx_good_bytes` `int` ***Read-only***

`rx_good_packets` `int` ***Read-only***

`rx_jabber_packets` `int` ***Read-only***

`rx_multicast_packets` `int` ***Read-only***

`rx_oversize_packets` `int` ***Read-only***

`rx_packets_rate` `int` ***Read-only***

`rx_pause` `int` ***Read-only***

`rx_undersize_packets` `int` ***Read-only***

`rx_unicast_packets` `int` ***Read-only***

`tx_broadcast_packets` `int` ***Read-only***

`tx_bytes` `int` ***Read-only***

`tx_bytes_rate` `int` ***Read-only***

`tx_collisions` `int` ***Read-only***

`tx_deferred` `int` ***Read-only***

`tx_excessive` `int` ***Read-only***

`tx_fcs` `int` ***Read-only***

`tx_late` int ***Read-only***

`tx_multicast_packets` int ***Read-only***

`tx_multiple` int ***Read-only***

`tx_packets` int ***Read-only***

`tx_packets_rate` int ***Read-only***

`tx_pause` int ***Read-only***

`tx_single` int ***Read-only***

`tx_unicast_packets` int ***Read-only***

Switch API

Get the current switch status

`GET /api/v8/switch/status/`

Return the list of swith port status SwitchPortStatus

Example request:

```
GET /api/v8/switch/status/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "duplex": "half",
            "link": "down",
            "id": 3,
            "mode": "10BaseT-HD",
            "speed": "10"
        },
        {
            "duplex": "full",
            "link": "up",
            "id": 1,
            "mode": "1000BaseT-FD",
            "speed": "1000"
        },
        {
            "duplex": "half",
            "link": "down",
            "id": 2,
            "mode": "10BaseT-HD",
            "speed": "10"
        },
        {
            "duplex": "full",
            "mac_list": [
                {
                    "mac": "00:24:D4:7E:00:4C",
                    "hostname": "r0ro's player"
                }
            ],
            "link": "up",
            "id": 4,
            "mode": "1000BaseT-FD",
            "speed": "1000"
        }
    ]
}
```


Get a port configuration

`GET /api/v8/switch/port/{id}`

Get the SwitchPortConfig for the given port id

Example request:

```
GET /api/v8/switch/port/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "id": 1,
        "speed": "auto",
        "duplex": "auto"
    }
}
```


Update a port configuration

`PUT /api/v8/switch/port/{id}`

Update the SwitchPortConfig for the given port id

Example request:

```
PUT /api/v8/switch/port/1 HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
   "speed": "10"
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "id": 4,
        "speed": "10",
        "duplex": "auto"
```

```
    }
}
```

Get a port stats

`GET /api/v8/switch/port/{id}/stats`

Get the SwitchPortStats for the given port id

Example request:

```
GET /api/v8/switch/port/4/stats HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "rx_packets_rate": 4,
        "rx_good_bytes": 20018805,
        "rx_oversize_packets": 0,
        "rx_unicast_packets": 113034,
        "tx_bytes_rate": 736,
        "tx_unicast_packets": 112409,
        "rx_bytes_rate": 608,
        "tx_packets": 166266,
        "tx_collisions": 0,
        "tx_packets_rate": 6,
        "tx_fcs": 0,
        "tx_bytes": 25316860,
        "rx_jabber_packets": 0,
        "tx_single": 0,
        "tx_excessive": 0,
        "rx_pause": 0,
        "rx_multicast_packets": 1217,
        "tx_pause": 0,
        "rx_good_packets": 114296,
        "rx_broadcast_packets": 45,
        "tx_multiple": 0,
        "tx_deferred": 0,
        "tx_late": 0,
        "tx_multicast_packets": 27962,
        "rx_fcs_packets": 0,
        "tx_broadcast_packets": 25895,
        "rx_err_packets": 0,
        "rx_fragments_packets": 0,
        "rx_bad_bytes": 0,
        "rx_undersize_packets": 0
    }
}
```


Wi-Fi

The Wi-Fi API allow you to control the settings of the Freebox Wi-Fi.

Wi-Fi Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
inval	invalid parameters
exist	entry already exists
nospc	maximum entry count reached
nodev	invalid device id
noent	invalid id
busy	device busy
inval_band	invalid wifi band
inval_ssid	invalid ssid
inval_freq	invalid wifi frequency
inval_cipher	invalid cipher mod
inval_key_len	invalid key length
inval_key	invalid key
inval_ht_needs_wmm	wmm must be enabled for 802.11n
inval_ac_needs_ht	invalid configuration 802.11ac need ht support
inval_ac_not_2d4g	invalid configuration 802.11ac is not supported on 2.4G band
inval_wps_needs_ccmp	wps need WPA2/AES to be enabled
inval_wps_macfilter	wps cannot work when mac filter is enabled
inval_wps_hidden_ssid	wps cannot work with hidden ssid
inval_eht_needs_he	802.11ax must be enabled for 802.11be
inval_ht_needs_ht	802.11n must be enabled for 802.11ax on 2.4G band
inval_ht_needs_vht	802.11ac must be enabled for 802.11ax on 6G band
inval_6g_needs_he	6G band requires 802.11ax

Wi-Fi Global Config

Global config gives quick access to major configuration settings (eg: toggle Wi-Fi)

WifiGlobalConfig has the following attributes:

`WifiGlobalConfig`

`enabled` **bool**

is wifi enabled

`mac_filter_state` **enum**

mac_filter_state	Description
disabled	mac filter is disabled
whitelist	mac filter is enabled, using a whitelist

mac_filter_state	Description
blacklist	mac filter is enabled, using a blacklist

Wi-Fi global config API

Get the current Wi-Fi global configuration

`GET /api/v9/wifi/config/`

Get the WifiGlobalConfig

Example request:

```
GET /api/v9/wifi/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": true,
        "mac_filter_state": "blacklist"
    }
}
```

Update the Wi-Fi global configuration

`PUT /api/v9/wifi/config/`

Update the WifiGlobalConfig

Example request:

```
PUT /api/v9/wifi/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "enabled": false
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "enabled": false,
        "mac_filter_state": "blacklist"
    }
}
```

Wi-Fi global state

Wi-Fi global state object

`WifiGlobalState`

`state` enum ***Read-only***

wifi global state

state	Description
enabled	Wifi is enabled
disabled	Wi-Fi is disabled
disabled_planning	Wi-Fi is disabled by planning

`expected_phys[]` array of ExpectedPhy ***Read-only***

expected wifi cards

`ExpectedPhy`

`band` enum ***Read-only***

state	Description
2d4g	2.4GHz band
5g	5GHz band
6g	6 GHz band
60g	60GHz band

`phy_id` int ***Read-only***

id of the phy

`detected` bool ***Read-only***

true if the wifi card is detected

Wi-Fi global state API

Get the global wifi state

`GET /api/v10/wifi/state/`

Get the global wifi state WifiGlobalState

Example request:

```
GET /api/v10/wifi/state/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
        {
            "state": "enabled",
            "expected_phys": [
                {
                    "band": "2d4g",
                    "phy_id": 0,
                    "detected": true
                },
                {
```

```
                    "band": "5g",
                    "phy_id": 1,
                    "detected": true
                }
            ],
        }
    ]
}
```


Wi-Fi Access Point

Wi-Fi AP objects

The Freebox may have one or more access points, you can configure each access point with this api.

WifiAp

- `id` int ***Read-only***
  wifi ap id
- `name` string ***Read-only***
  wifi ap name
- `status` WifiApStatus ***Read-only***
  ap status
- `capabilities` WifiApCapabilities ***Read-only***
  ap capabilities
- `config` WifiApConfig
  ap configuration

WifiApStatus

- `state` enum ***Read-only***

state	Description
scanning	Ap is probing wifi channels
no_param	Ap is not configured
bad_param	Ap has an invalid configuration
disabled	Ap is permanently disabled
disabled_planning	Ap is currently disabled according to planning
disabled_power_saving	Ap is currently disabled according to power save
disabled_temp	Ap is currently disabled temporarily
no_active_bss	Ap has no active BSS
starting	Ap is starting
starting	Ap is stopping
acs	Ap is selecting the best available channel
ht_scan	Ap is scanning for other access point
dfs	Ap is performing dynamic frequency selection
active	Ap is active
failed	Ap has failed to start

- `channel_width` int ***Read-only***
  effective channel width (in MHz)
- `primary_channel` int ***Read-only***
  effective primary channel
- `secondary_channel` int ***Read-only***
  effective secondary channel
- `dfs_cac_remaining_time` int ***Read-only***
  time left in dfs state
- `dfs_disabled` bool ***Read-only***
  Indicates if DFS channels are unavailable regardless of how the WifiApConfig is configured for this phy. This is enabled when your freebox is in compatibility mode for other Freebox wifi products.
- `temp_disable_remaining_time` int ***Read-only***
  Optional remaining time this access point is temporarily disabled.

WifiApCapabilities

[UNSTABLE]

- `2d4g` int ***Read-only***
  map of capabilities in 2.4 GHz band
- `5g` int ***Read-only***
  map of capabilities in 5 GHz band
- `6g` int ***Read-only***
  map of capabilities in 6 GHz band
- `60g` int ***Read-only***
  map of capabilities in 60 GHz band

NOTE: before enabling some feature in ap config, you should ensure that AP supports the feature using its provided capabilities.

WifiApHtConfig

- `ac_enabled` bool
  enable 802.11ac
- `ht_enabled` bool
  enable 802.11n

  [UNSTABLE]

WifiApHeConfig

- `enabled` bool
  enable 802.11ax (HE)

  [UNSTABLE]

WifiApConfig

- `band` enum

band	Description
2d4g	2.4 GHz
5g	5 GHz

band	Description
6g	6 GHz
60g	60 GHz

`channel_width` **int**

wanted channel width (in MHz) :

- 20 MHz
- 40 MHz
- 80 MHz
- 160 MHz

`primary_channel` **int**

wanted primary channel, value of 0 means automatic selection

`secondary_channel` **int**

wanted secondary channel, value of 0 means automatic selection

`dfs_enabled` **bool**

enable channels that require DFS

`ht` **WifiApHtConfig**

wifi ht config

`he` **WifiApHeConfig**

wifi HE config

`WifiApChannelSurveyData`

`timestamp` **int**

timestamp at which the survey data was retrieved

`busy_percent` **int**

percentage of time the channel was sensed busy

`tx_percent` **int**

percentage of time spent sending on the channel

`rx_percent` **int**

percentage of time spent receiving Wi-Fi traffic on the channel

`rx_bss_percent` **int**

percentage of time spent receiving Wi-Fi traffic for a local BSS

Wi-Fi AP API

Get the ap list

`GET /api/v9/wifi/ap/`

Get the list of Freebox Access Points WifiAp

Example request:

```
GET /api/v9/wifi/ap/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "capabilities": {
                "2d4g": {
                    "shortgi20": true,
                    "vht_rx_ldpc": false,

                     [ ... ]

                    "shortgi40": true,
                },
                "60g": {
                     [ ... ]
                },
                "5g": {
                     [ ... ]
                }
            },
            "name": "2.4G",
            "id": 0,
            "config": {
                "channel_width": "40",
                "ht": {
                    "ht_enabled": true,
                    "ac_enabled": false,

                    [ ... ]
                },
                "dfs_enabled": false,
                "band": "2d4g",
                "secondary_channel": 13,
                "primary_channel": 9
            },
            "status": {
                "channel_width": "20",
                "primary_channel": 9,
                "dfs_cac_remaining_time": 0,
                "secondary_channel": 0,
                "state": "active"
            }
        }
    ]
}
```


Get a particular AP

`GET /api/v9/wifi/ap/{id}`

Get the WifiAp with the requested id

Example request:

```
GET /api/v9/wifi/ap/0 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "capabilities": {
            "2d4g": {
                "shortgi20": true,
                "vht_rx_ldpc": false,

                 [ ... ]

                "shortgi40": true,
            },
            "60g": {
                [ ... ]
            },
            "5g": {
                [ ... ]
            }
        },
        "name": "2.4G",
        "id": 0,
        "config": {
            "channel_width": "40",
            "ht": {
                "ht_enabled": true,
                "ac_enabled": false,

                [ ... ]
            },
            "dfs_enabled": false,
            "band": "2d4g",
            "secondary_channel": 13,
            "primary_channel": 9
        },
        "status": {
            "channel_width": "20",
            "primary_channel": 9,
            "dfs_cac_remaining_time": 0,
            "secondary_channel": 0,
            "state": "active"
        }
    }
}
```


Update an AP

PUT /api/v9/wifi/ap/{id}

Update the WifiAp

Example request:

```
PUT /api/v9/wifi/ap/0 HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
  "config": {
    "channel_width": "20",
    "ht": {
        "ht_enabled": false
    },
    "primary_channel": 0,
    "secondary_channel": 0
  }
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "capabilities": [ ... ],
        "name": "2.4G",
        "id": 0,
        "config": {
            "channel_width": "20",
            "ht": {
                "ht_enabled": false,
                "ac_enabled": false

                [ ... ]
            },
            "dfs_enabled": false,
            "band": "2d4g",
            "secondary_channel": 0,
            "primary_channel": 0
        },
        "status": {
            "channel_width": "20",
            "primary_channel": 0,
            "dfs_cac_remaining_time": 0,
            "secondary_channel": 0,
            "state": "scanning"
        }
    }
}
```


Wi-Fi AP allowed channels

To be able to allow user to pick a valid channel combination for a given AP you should use the following api to retrieve the list of allowed channel combination.

WifiAllowedComb

band enum ***Read-only***

the band for which the combination can be used

band	Description
2d4g	2.4 GHz
5g	5 GHz
60g	60 GHz

channel_width string ***Read-only***

the channel_width for which the combination can be used

`need_dfs` `bool` ***Read-only***

does this combination requires DFS.

You should only allow this combination if ap has allowed dfs.

`dfs_cac_time` `int` ***Read-only***

time required in dfs state before being able to start the AP.

`psc` `bool` ***Read-only***

is this using a PSC channel as primary.

Some phones/PCs can only see 6GHz APs when their primary channel is a Preferred Scanning Channel (PSC).

`primary` `int` ***Read-only***

primary channel

`secondary` `int` ***Read-only***

secondary channel (zero means that secondary channel will not be used)

`GET /api/v9/wifi/ap/{id}/allowed_channel_comb`

Get the `WifiAllowedComb` for the given ap id

Example request:

```
GET /api/v9/wifi/ap/0/allowed_channel_comb HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "channel_width": "20",
            "dfs_cac_time": 0,
            "need_dfs": false,
            "primary": 1,
            "band": "2d4g",
            "secondary": 0
        },

        [ ... ]

        {
            "channel_width": "20",
            "dfs_cac_time": 0,
            "need_dfs": false,
            "primary": 13,
            "band": "2d4g",
            "secondary": 0
        },
        {
            "channel_width": "40",
            "dfs_cac_time": 0,
            "need_dfs": false,
            "primary": 1,
            "band": "2d4g",
            "secondary": 5
        },

        [ ... ]

        {
            "channel_width": "40",
            "dfs_cac_time": 0,
            "need_dfs": false,
            "primary": 13,
            "band": "2d4g",
            "secondary": 9
        }
    ]
}
```


Wi-Fi AP stations

Wi-Fi AP Stations objects
WifiStation has the following attributes:

`WifiStation`

`id` `string` ***Read-only***

station id

`mac` `string` ***Read-only***

client MAC address

`bssid` `string` ***Read-only***

bssid on which the client is associated

`hostname` `string` ***Read-only***

client host name

`host` `LanHost` ***Read-only***

client host information

`state` `enum` ***Read-only***

state	Description
associated	station is associated
authenticated	station is authenticated

`inactive` `int` ***Read-only***

inactive duration (in seconds)

`conn_duration` `int` ***Read-only***

connection duration (in seconds)

`rx_bytes` `int` ***Read-only***

received bytes (from station to Freebox)

`tx_bytes` `int` ***Read-only***

transmitted bytes (from Freebox to station)

`tx_rate` `int` ***Read-only***

reception data rate (in bytes/s)

`rx_rate` `int` ***Read-only***
transmission data rate (in bytes/s)

`signal` `int` ***Read-only***
signal attenuation (in dB)

`flags` `WifiStationFlags` ***Read-only***
station flags

`last_rx` `WifiStationStats` ***Read-only***
last rx stats

`last_tx` `WifiStationStats` ***Read-only***
last tx stats

`WifiStationFlags`
[UNSTABLE]

`legacy` `bool` ***Read-only***
does station uses legacy wifi (802.11a, 802.11b)

`ht` `bool` ***Read-only***
does station support ht (802.11n)

`vht` `bool` ***Read-only***
does station support vht (802.11ac)

`he` `bool` ***Read-only***
does station support he (802.11ax)

`authorized` `bool` ***Read-only***
is the station authenticated

`WifiStationStats`
[UNSTABLE]

`bitrate` `int` ***Read-only***
physical link rate (in 1/10th of MBit/s), -1 if unknown

`mcs` `int` ***Read-only***
current link mcs, -1 if not used

`vht_mcs` `int` ***Read-only***
current link vht mcs, -1 if not used

`width` `string` ***Read-only***
current channel width

`shortgi` `bool` ***Read-only***
is shortgi enabled

Get Wi-Fi Stations List

`GET /api/v9/wifi/ap/{id}/stations/`

Get the list of `WifiStation` associated to the AP

Example request:

```
GET /api/v9/wifi/ap/0/stations/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "mac": "18:AF:36:15:69:42",
            "last_rx": {
                "bitrate": 110,
                "mcs": -1,
                "shortgi": false,
                "vht_mcs": -1,
                "width": "20"
            },
            "tx_bytes": 2651,
            "last_tx": {
                "bitrate": 360,
                "mcs": -1,
                "shortgi": false,
                "vht_mcs": -1,
                "width": "20"
            },
            "id": "00:24:D4:AC:DC:88-18:AF:36:15:69:42",
            "bssid": "00:24:D4:AC:DC:88",
            "flags": {
                "vht": false,
                "legacy": false,
                "authorized": true,
                "ht": false
            },
            "tx_rate": 0,
            "host": {
                [ ... ]
            },
            "inactive": 168,
            "conn_duration": 263,
            "hostname": "iPhone-de-r0ro",
            "state": "authenticated",
            "rx_bytes": 781,
            "rx_rate": 0,
            "signal": -38
        }
    ]
}
```


Get Wi-Fi Station

`GET /api/v9/wifi/ap/{id}/stations/{mac}`

Get a `WifiStation` associated to the AP

Example request:

```
GET /api/v9/wifi/ap/0/stations/18:AF:36:15:69:42 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "mac": "18:AF:36:15:69:42",
        "last_rx": {
            "bitrate": 110,
            "mcs": -1,
            "shortgi": false,
            "vht_mcs": -1,
            "width": "20"
        },
        "tx_bytes": 2651,
        "last_tx": {
            "bitrate": 360,
            "mcs": -1,
            "shortgi": false,
            "vht_mcs": -1,
            "width": "20"
        },
        "id": "00:24:D4:AC:DC:88-18:AF:36:15:69:42",
        "bssid": "00:24:D4:AC:DC:88",
        "flags": {
            "vht": false,
            "legacy": false,
            "authorized": true,
            "ht": false
        },
        "tx_rate": 0,
        "host": {
            [ ... ]
        },
        "inactive": 168,
        "conn_duration": 263,
        "hostname": "iPhone-de-r0ro",
        "state": "authenticated",
        "rx_bytes": 781,
        "rx_rate": 0,
        "signal": -38
    }
}
```


Wi-Fi AP channel survey history

Retrieve survey data for the channel the AP is operating on, starting from a given timestamp.

Get survey data history

`GET /api/v9/wifi/ap/{id}/channel_survey_history/{timestamp}`

Get an array of WifiApChannelSurveyData

Example request:

```
GET /api/v9/wifi/ap/0/channel_survey_history/1651135474000 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
      "success": true,
      "result": [
            {
                  "busy_percent": 65,
                  "tx_percent": 2,
                  "timestamp": 1651135474996,
                  "rx_bss_percent": 0,
                  "rx_percent": 56
            },
            {
                  "busy_percent": 70,
                  "tx_percent": 3,
                  "timestamp": 1651135475796,
                  "rx_bss_percent": 0,
                  "rx_percent": 58
            },
            {
                  "busy_percent": 71,
                  "tx_percent": 3,
                  "timestamp": 1651135475896,
                  "rx_bss_percent": 0,
                  "rx_percent": 58
            },
            {
                  "busy_percent": 73,
                  "tx_percent": 4,
                  "timestamp": 1651135475998,
                  "rx_bss_percent": 0,
                  "rx_percent": 59
            }
      ]
}
```


Restart an AP

WARNING during the restart the AP will be unavailable. You may not receive the response if you restart the Wifi card you are using to call the api

This will restart an AP, this is useful when an AP is in failed state. This is the same as disabling/re-enabling the BSS on an AP.

`POST /api/v9/wifi/ap/{id}/restart`

Restarts the AP

Example request:

```
POST /api/v9/wifi/ap/0/restart HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Wi-Fi BSS

Each AP can manage a set of BSS, with this api you can manage BSS settings

Wi-Fi BSS objects

`WifiBss`

`id` `int` ***Read-only***
bss id

`phy_id` `string` ***Read-only***
associated AP id

`status` `WifiBssStatus` ***Read-only***
bss status

`use_shared_params` `bool`
if set to True the bss will use the shared parameters stored under shared_bss_params

if not the bss will use a configuration specific to this bss stored under bss_params

when you want to edit the bss config you should change the config values using values from bss_params or shared_bss_params as a source and update use_shared_params accordingly.

`config` `WifiBssConfig`
bss configuration (use this field for editing)

`bss_params` `WifiBssConfig` ***Read-only***
current configuration specific to this bss

`shared_bss_params` `WifiBssConfig` ***Read-only***
current configuration for shared bss config

`disable_wep` `bool` ***Read-only***
Whether or not this BSS can work with wep encryption or not

`WifiBssStatus`

`state` `enum` ***Read-only***

state	Description
phy_stopped	associated AP is stopped
no_param	bss is missing config
bad_param	bss has an invalid config
disabled	bss is disabled
temp_disabled	bss has been temporary disabled
starting	bss is starting
active	bss is active
failed	bss has failed to start

`sta_count` `int` ***Read-only***
number of stations for this bss

`authorized_sta_count` `int` ***Read-only***
number of authenticated stations for this bss

`custom_key_ssid` `string` ***Read-only***
SSID to use with custom keys

`is_main_bss` `bool` ***Deprecated***
this as been replaced by use_shared_params in `WifiBss`

`partners` `[int]` ***Read-only***
The currently active MLO partners's AP for this BSS. Can be empty if MLO is disabled. See the MLO chapter for more info

`WifiBssConfig`

`enabled` `bool`
enable this BSS. Note that if you want the AP to completely stop emitting wifi you should use `WifiGlobalConfig` enabled attribute.

`use_default_config` `bool` ***Deprecated***
this as been replaced by use_shared_params in `WifiBss`

`ssid` `str`
bss displayed name

`hide_ssid` `str`
don't show bss in bss list

`gcmp256` `str`
Whether or not to use GCMP-256 (only in WPA3 & for box that supports 802.11-be)

`encryption` `enum`

encryption	Description
wep	wep (should not use)
wpa_psk_auto	wpa1 CCMP+TKIP (should not use)
wpa_psk_tkip	wpa1 TKIP (should not use)
wpa_psk_ccmp	wpa1 CCMP (should not use)
wpa12_psk_auto	wpa1+wpa2 CCMP+TKIP (should not use)
wpa2_psk_auto	wpa2 CCMP+TKIP (should not use)
wpa2_psk_tkip	wpa2 TKIP (should not use)
wpa2_psk_ccmp	wpa2 CCMP
wpa23_psk_ccmp	wpa2+wpa3 CCMP WPA3-personal transition mode
wpa23_psk_ccmp_mrsno	wpa2+wpa3 CCMP WPA3-personal compatibility mode
wpa3_psk_ccmp	wpa3 CCMP WPA3-personal only mode

`key` `string` ***Write-only***
wifi key

`eapol_version` `int` ***Read-only***
eapol version

Wi-Fi BSS API

Get the bss list

`GET /api/v9/wifi/bss/`

Get the list of Freebox Access Points `WifiBss`

Example request:

```
GET /api/v9/wifi/bss/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "id": "00:24:D4:AA:BB:CC",
            "phy_id": 0,
            "use_shared_params": false,
            "config": {
                  "enabled": true,
                  "ssid": "r0ro 2.4",
                  "encryption": "wpa2_psk_ccmp",
                  "use_default_config": false,
                  "hide_ssid": false,
                  "eapol_version": 2,
                  "wps_enabled": true,
                  "wps_uuid": "37f5c24a-4d8f-4dfc-9321-c40c42e588c0",
                  "key": "jesaispasdevine!"
            },
            "bss_params": {
                  "enabled": true,
                  "ssid": "r0ro 2.4",
                  "encryption": "wpa2_psk_ccmp",
                  "hide_ssid": false,
                  "eapol_version": 2,
                  "wps_enabled": true,
                  "wps_uuid": "37f5c24a-4d8f-4dfc-9321-c40c42e588c0",
                  "key": "jesaispasdevine!"
            },
            "shared_bss_params": {
                  "enabled": true,
                  "ssid": "r0ro",
                  "encryption": "wpa2_psk_ccmp",
                  "hide_ssid": false,
                  "eapol_version": 2,
                  "wps_enabled": true,
                  "wps_uuid": "37f5c24a-4d8f-4dfc-9321-c40c42e588c0",
                  "key": "lav7lav7!"
            },
            "status": {
                "state": "active",
                "sta_count": 1,
                "authorized_sta_count": 1,
                "is_main_bss": true
            }
        },

        [ ... ]
    ]
}
```


Get a particular BSS

GET /api/v9/wifi/bss/{id}

Get the WifiBss with the requested id

Example request:

```
GET /api/v9/wifi/bss/00:24:D4:AA:BB:CC HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
      "id": "00:24:D4:AA:BB:CC",
      "phy_id": 0,
      "use_shared_params": false,
      "config": {
            "enabled": true,
            "ssid": "r0ro 2.4",
            "encryption": "wpa2_psk_ccmp",
            "use_default_config": false,
            "hide_ssid": false,
            "eapol_version": 2,
            "wps_enabled": true,
            "wps_uuid": "37f5c24a-4d8f-4dfc-9321-c40c42e588c0",
            "key": "jesaispasdevine!"
      },
      "bss_params": {
            "enabled": true,
            "ssid": "r0ro 2.4",
            "encryption": "wpa2_psk_ccmp",
            "hide_ssid": false,
            "eapol_version": 2,
            "wps_enabled": true,
            "wps_uuid": "37f5c24a-4d8f-4dfc-9321-c40c42e588c0",
            "key": "jesaispasdevine!"
      },
      "shared_bss_params": {
            "enabled": true,
            "ssid": "r0ro",
            "encryption": "wpa2_psk_ccmp",
            "hide_ssid": false,
            "eapol_version": 2,
            "wps_enabled": true,
            "wps_uuid": "37f5c24a-4d8f-4dfc-9321-c40c42e588c0",
            "key": "lav7lav7!"
      },
      "status": {
          "state": "active",
          "sta_count": 1,
          "authorized_sta_count": 1,
          "is_main_bss": true
      }
```

```
    }
}
```

Update an BSS

`PUT /api/v9/wifi/bss/{id}`

Update the WifiAp

Example request:

```
PUT /api/v9/wifi//bss/00:24:D4:AA:BB:CC HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "config": {
    "key": "c'était trop facile"
  }
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
      "id": "00:24:D4:AA:BB:CC",
      "phy_id": 0,
      "use_shared_params": false,
      "config": {
            "enabled": true,
            "ssid": "r0ro 2.4",
            "encryption": "wpa2_psk_ccmp",
            "use_default_config": false,
            "hide_ssid": false,
            "eapol_version": 2,
            "wps_enabled": true,
            "wps_uuid": "37f5c24a-4d8f-4dfc-9321-c40c42e588c0",
            "key": "jesaispasdevine!"
      },
      "bss_params": {
            "enabled": true,
            "ssid": "r0ro 2.4",
            "encryption": "wpa2_psk_ccmp",
            "hide_ssid": false,
            "eapol_version": 2,
            "wps_enabled": true,
            "wps_uuid": "37f5c24a-4d8f-4dfc-9321-c40c42e588c0",
            "key": "c'était trop facile"
      },
      "shared_bss_params": {
            "enabled": true,
            "ssid": "r0ro",
            "encryption": "wpa2_psk_ccmp",
            "hide_ssid": false,
            "eapol_version": 2,
            "wps_enabled": true,
            "wps_uuid": "37f5c24a-4d8f-4dfc-9321-c40c42e588c0",
            "key": "lav7lav7!"
      },
      "status": {
          "state": "active",
          "sta_count": 1,
          "authorized_sta_count": 1,
          "is_main_bss": true
      }
    }
}
```


Wi-Fi Radar

With this api you can list the surrounding Wi-Fi access points, and Wi-fi channel usage.

This a new feature introduced in firmware 2.1.0 (api v2).

A scan is automatically done at AP startup, if you need to refresh the information you can use the scan api

Wi-Fi Neighbor Object

WifiNeighbor has the following attributes:

`WifiNeighbor`

- `bssid` `string` ***Read-only***
  neighbor bssid
- `ssid` `string` ***Read-only***
  neighbor ssid
- `band` `enum` ***Read-only***
  the band for which the combination can be used

band	Description
2d4g	2.4 GHz
5g	5 GHz
60g	60 GHz

- `channel_width` `int` ***Read-only***
  neighbor channel_width
- `channel` `int` ***Read-only***
  neighbor primary channel
- `secondary_channel` `int` ***Read-only***
  neighbor secondary channel (0 for unused)
- `signal` `int` ***Read-only***
  signal attenuation in dB
- `capabilities` WifiNeighborCap ***Read-only***
  neighbor capabilities

`WifiNeighborCap`

- `legacy` `bool` ***Read-only***
  neighbor uses legacy wifi (802.11a, 802.11b)
- `ht` `bool` ***Read-only***
  neighbor supports ht (802.11n)

`vht` `bool` ***Read-only***

neighbor supports vht (802.11ac)

List AP neighbors

`GET /api/v9/wifi/ap/{id}/neighbors/`

Get the list of `WifiNeighbor` seen by the AP

Example request:

```
GET /api/v9/wifi/ap/0/neighbors/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "channel_width": "20",
            "capabilities": {
                "legacy": false,
                "vht": false,
                "ht": true
            },
            "ssid": "Freebox-future",
            "channel": 1,
            "band": "2d4g",
            "bssid": "00:24:D4:BA:BB:EE",
            "secondary_channel": 0,
            "signal": -27
        },

        [ ... ]

        {
            "channel_width": "20",
            "capabilities": {
                "legacy": false,
                "vht": false,
                "ht": true
            },
            "ssid": "Encore une freebox",
            "channel": 1,
            "band": "2d4g",
            "bssid": "F4:CA:E5:5E:AC:4F",
            "secondary_channel": 0,
            "signal": -33
        },
        {
            "channel_width": "20",
            "capabilities": {
                "legacy": false,
                "vht": false,
                "ht": true
            },
            "ssid": "lav6-140c76670212",
            "channel": 1,
            "band": "2d4g",
            "bssid": "00:07:CB:00:00:FD",
            "secondary_channel": 0,
            "signal": -33
        }
    ]
}
```


Wi-Fi Channel usage Object

`WifiChannelUsage`

`channel` `int` ***Read-only***

channel number

`band` `enum` ***Read-only***

band	Description
2d4g	2.4 GHz
5g	5 GHz
60g	60 GHz

`noise_level` `int` ***Read-only***

noise level on channel in dB

`rx_busy_percent` `int` ***Read-only***

rx channel busy time percentage

List Wi-Fi channels usage

`GET /api/v9/wifi/ap/{id}/channel_usage/`

Get the list of `WifiChannelUsage` for the given AP

Example request:

```
GET /api/v9/wifi/ap/0/channel_usage/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": "result": [
       {
            "band": "2d4g",
            "noise_level": -66,
            "rx_busy_percent": 35,
            "channel": 1
       },

       [ ... ]

       {
            "band": "2d4g",
```

```
            "noise_level": -58,
            "rx_busy_percent": 46,
            "channel": 13
        }
    ]
}
```

Refresh radar informations

WARNING during the scan the AP will be unavailable. Therefore, you should ask for user confirmation prior to launching a scan.

Once launched you should wait until the ap state comes back from scanning to get updated info.

`POST /api/v9/wifi/ap/{id}/neighbors/scan`

Launch a wifi scan on given ap

Example request:

```
POST /api/v9/wifi/ap/0/neighbors/scan HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Wi-Fi Planning

With api v2 you can now specify time range when you want to enable your wifi.

Wi-Fi Planning Object

`WifiPlanning`

- `use_planning` **bool**
  is the planning enabled
- `resolution` **int** ***Read-only***
  planning resolution (number of slots per day)
- `mapping[]` **array of str**
  mapping for planning : “on” or “off” mapping[0] is monday at 0:0 mapping[7 * resolution - 1] is sunday last slot
  (each slot has a duration of 60 * 24 / resolution minutes)

Get Wi-Fi Planning

`GET /api/v9/wifi/planning/`

Get the current `WifiPlanning`

Example request:

```
GET /api/v9/wifi/planning/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "use_planning": false,
        "resolution": 48,
        "mapping": [
            "on",
            "on",
            "on",
            "on",

            [ ... ]

            "on",
            "on",
            "on",
            "on"
        ]
    }
}
```

Update Wi-Fi Planning

`PUT /api/v9/wifi/planning/`

Update the `WifiPlanning`

Example request:

```
PUT /api/v9/wifi/planning/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "use_planning": true
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "use_planning": true,
        "resolution": 48,
        "mapping": [
            "on",
            "on",
            "on",
            "on",

            [ ... ]

            "on",
            "on",
            "on",
            "on"
        ]
```

```
    }
}
```

Wi-Fi MAC Filter API

Wi-Fi MAC Filter object

WifiMacFilter has the following attributes:

`WifiMacFilter`

`id` `string` ***Read-only***
filter id

`mac` `string` ***Read-only***
MAC address to filter

`comment` `string`
comment

`type` `enum`

type	Description
whitelist	if mac_filter is set to whitelist this station will be allowed
blacklist	if mac_filter is set to blacklist this station will be rejected

`hostname` `string` ***Read-only***
host name when available

`host` `LanHost` ***Read-only***
host information when available

Get the MAC filter list

`GET /api/v9/wifi/mac_filter/`

Get the list of `WifiMacFilter`

Example request:

```
GET /api/v9/wifi/mac_filter/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
        {
            "mac": "00:07:CB:01:02:03",
            "type": "whitelist",
            "comment": "test",
            "hostname": "00:07:CB:01:02:03",
            "id": "00:07:CB:01:02:03"
        },
        {
            "mac": "00:24:D4:00:00:69",
            "type": "blacklist",
            "comment": "plop",
            "hostname": "r0ro's iPad",
            "id": "00:24:D4:00:00:69",
            "host": {
               [ ... ]
            }
        }
    ]
}
```

Getting a particular MAC filter

`GET /api/v9/wifi/mac_filter/{filter_id}`

Returns the requested `WifiMacFilter` properties

Example request:

```
GET /api/v9/wifi/mac_filter/00:07:CB:01:02:03 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "mac": "00:07:CB:01:02:03",
        "type": "whitelist",
        "comment": "test",
        "hostname": "00:07:CB:01:02:03",
        "id": "00:07:CB:01:02:03"
    }
}
```

Updating a MAC filter

`PUT /api/v9/wifi/mac_filter/{filter_id}`

Update a `WifiMacFilter` properties

Example request:

```
PUT /api/v9/wifi/mac_filter/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "comment": "filtre de test",
   "type": "blacklist"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "mac": "00:07:CB:01:02:03",
        "type": "blacklist",
```

```
        "comment": "filtre de test",
        "hostname": "00:07:CB:01:02:03",
        "id": "00:07:CB:01:02:03"
    }
}
```


Delete a MAC filter

`DELETE /api/v9/wifi/mac_filter/{filter_id}`

Delete the WifiMacFilter with the given id

Example request:

```
DELETE /api/v9/wifi/mac_filter/00:07:CB:01:02:03 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Create a new MAC filter

`POST /api/v9/wifi/mac_filter/`

Crate a new the WifiMacFilter

Example request:

```
POST /api/v9/wifi/mac_filter/00:07:CB:01:02:03 HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
   "comment": "filtre de test",
   "type": "blacklist",
   "mac": "00:07:CB:CB:07:00"
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "mac": "00:07:CB:CB:07:00",
        "type": "blacklist",
        "comment": "filtre de test",
        "hostname": "00:07:CB:CB:07:00",
        "id": "00:07:CB:CB:07:00"
    }
}
```


Wifi Config reset

Global reset

You can reset Wifi to default configuration with this api

`POST /api/v9/wifi/config/reset/`

Create a new the WifiMacFilter

Example request:

```
POST /api/v9/wifi/config/reset/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Config reset value of an AP

You can get the default config value of a given AP.

`GET /api/v9/wifi/ap/{id}/default`

Get the WifiApConfig with the requested id

Example request:

```
GET /api/v9/wifi/ap/0/default HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true,
  "result": {
    "channel_width": "20",
    "ht": {
      [ ... ]
    },
    "dfs_enabled": false,
    "band": "2d4g",
    "secondary_channel": 0,
    "primary_channel": 0
  }
}
```


Config reset value of a BSS

You can get the default config value for a given BSS.

`GET /api/v9/wifi/bss/{id}/default`

Get the WifiBssConfig with the requested bssid

Example request:

```
GET /api/v9/wifi/bss/02:00:00:00:00:00/default HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true,
  "result": {
    "enabled": true,
    "wps_uuid": "7ace9cb4-3aec-486e-b487-28df4998ff46",
    "ssid": "super_ssid",
    "encryption": "wpa2_psk_ccmp",
    "wps_enabled": true,
    "hide_ssid": false,
    "eapol_version": 2,
    "key": "motdepasse"
  }
}
```


Config reset value (bulk)

This api gets the same data as the per AP/BSS ones but in one call only

GET /api/v9/wifi/default

Get the WifiBssConfig or WifiApConfig of all cards

Example request:

```
GET /api/v9/wifi/default HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true,
  "result": {
    "aps": [
      {
        "params": {
          "channel_width": "20",
          "ht": { ... },
          "dfs_enabled": false,
          "band": "2d4g",
          "secondary_channel": 0,
          "primary_channel": 0
        },
        "ap_id": 0
      },
      {
        "params": {
          "channel_width": "80",
          "ht": { ... },
          "dfs_enabled": true,
          "band": "5g",
          "secondary_channel": 0,
          "primary_channel": 0
        },
        "ap_id": 1
      }
    ],
    "bsss": [
      {
        "params": {
          "enabled": true,
          "wps_uuid": "cbf5826c-25b2-4795-a7c7-cbd8f9454431",
          "ssid": "super_ssid",
          "encryption": "wpa2_psk_ccmp",
          "wps_enabled": true,
          "hide_ssid": false,
          "eapol_version": 2,
          "key": "lolzme"
        },
        "bssid": "00:00:00:00:00:08"
      },
      {
        "params": {
          "enabled": true,
          "wps_uuid": "1d77f4c0-9544-4478-a8f0-cccb77031b94",
          "ssid": "super_ssid",
          "encryption": "wpa2_psk_ccmp",
          "wps_enabled": true,
          "hide_ssid": false,
          "eapol_version": 2,
          "key": "lolzme"
        },
        "bssid": "00:00:00:00:00:0C"
      }
    ]
  }
}
```


Diagnostic API

This API is intended to simplify detecting problems or suboptimal configs on bsss or aps. This API is articulated around the *WifiDiagItem*

WifiDiagItem

ap_id **int**

When this item relates to an AP, this indicates the AP's index When this item relates to a BSS, this field is unset

bssid **str**

When this item relates to a BSS, this field indicates the bss's id When this item relates to an AP, this field is unset

code **enum**

The code identifying which param is faulty/suboptimal

Code	Description
all	This is a the same as doing a full reset of this AP/BSS
network_disabled	This changes the 'enabled' field in WifiBssConfig
network_security	This changes the 'encryption' field in WifiBssConfig
network_visibility	This changes the 'hide_ssid' field in WifiBssConfig
channel_width	This changes the 'channel_width' field in WifiApConfig
channel_value	This changes the 'channel' & 'secondary_channel' fields in WifiApConfig

severity enum

Severity	Description
minor	minor problems don't have performance/compatibility implications
major	major problems do

Global diagnostic

The global diagnostics evaluates/works on all AP/BSS at once. This is good for bulk access

GET /api/v9/wifi/diag

Get the WifiDiagItem for the box

Example request:

```
GET /api/v9/wifi/diag HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
    "aps": [
      {
        "severity": "minor",
        "ap_id": 0,
        "code": "channel_width"
      },
      {
        "severity": "major",
        "ap_id": 1,
        "code": "channel_value"
      }
    ],
    "bsss": [
      {
        "severity": "major",
        "bssid": "02:00:00:00:00:08",
        "code": "network_security"
      },
      {
        "severity": "major",
        "bssid": "02:00:00:00:00:0C",
        "code": "network_visibility"
      }
    ]
  }
}
```

POST /api/v9/wifi/diag

Fix a few of the WifiDiagItem at once. 'aps' & 'bsss' are arrays in which you can put any items. You can also omit 'aps' and/or 'bsss'

Example request:

```
POST /api/v9/wifi/diag HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "aps": [
    {
      "ap_id": 0,
      "code": "channel_width"
    },
    [ ... ]
  ],
  "bsss": [
    {
      "bssid": "02:00:00:00:00:08",
      "code": "all"
    },
    [ ... ]
  ]
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success": true
}
```

Per AP/BSS diagnostic

Same as the global API there also is a per AP/BSS api to get/fix the problems.

GET /api/v9/wifi/ap/{id}/diag & /api/v9/wifi/bss/{id}/diag

Get the WifiDiagItem for the AP/BSS

Example request:

```
GET /api/v9/wifi/ap/0/bss HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": [
    {
      "severity": "minor",
      "ap_id": 0,
      "code": "channel_width"
    },
    {
      "severity": "major",
      "ap_id": 0,
      "code": "channel_value"
    },
  ]
}
```

POST /api/v9/wifi/ap/{id}/diag & /api/v9/wifi/bss/{id}/diag

Fix a few of the WifiDiagItem at once for a given AP/BSS

Example request:

```
POST /api/v9/wifi/bss/02:00:00:00:00:08/diag HTTP/1.1
Host: mafreebox.freebox.fr
```

```
[ "network_visibility", "network_visibility", ... ]
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success": true
}
```

Wifi WPS API

This api lets you open wps sessions on wifi a bss to allow a device to connect to Wifi using WPS

To be able to open wps session, you first need to make sure that the bss is properly configured (with WifiBssConfig field 'wps_enabled' set to true)

Note that wps_enabled requires the encryption to either be wpa2_psk_ccmp or wpa2_psk_auto

You should call the WifiWpsCandidate api help to check which bss can be used for wps

Also, only one WPS session can be active at a given time

Wifi Wps Candidate object

WifiWpsCandidate has the following attributes:

`WifiWpsCandidate`

`bssid` `string` ***Read-only***
bss id

`ssid` `string` ***Read-only***
wifi network name

`bss_uuid` `string` ***Read-only***
bss uuid for wps

`band` `string` ***Read-only***

band	Description
2d4g	2.4 GHz
5g	5 GHz
60g	60 GHz

`encryption` `enum` ***Read-only***
currently configured encryption mode see WifiBssConfig encryption field

`wps_enabled` `bool` ***Read-only***
is wps enabled for this bss

`state` `enum` ***Read-only***
the current state of the associated ap see WifiBssStatus state

Enable/disable WPS on all Wi-Fi cards

`GET /api/v9/wifi/wps/config/`

Get the global WPS state. WPS is globally enabled if at least one BSS has WPS enabled.

Example request:

```
GET /api/v9/wifi/wps/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success": true,
      "result": {
            "enabled": true
      }
}
```

`PUT /api/v9/wifi/wps/config/`

Set the global WPS state. It will update each BSS config with the provided state.

Example request:

```
PUT /api/v9/wifi/wps/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
      "enabled": false
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success": true,
      "result": {
            "enabled": false
      }
}
```

Wifi WPS Session object

WifiWpsSession has the following attributes:

`WifiWpsSession`

`id` `int` ***Read-only***
wps session id

`bss_uuid` `string` ***Read-only***
bss wps uuid

`ssid` `string` ***Read-only***
ssid

`active` `bool` ***Read-only***
is the session active

`result` `enum` ***Read-only***

result of the wps session

result	Description
success	success
user_canceled	canceled by user
self_canceled	canceled by restart of bss
failed_timeout	timeout while waiting for station
failed_overlap	another wps session was active
failed_unknown	unknown failure

`start_date` `int` ***Read-only***

session start date (timestamp)

`end_date` `enum` ***Read-only***

session end date (timestamp)

`mac` `string` ***Read-only***

mac of the associated client (in case of success)

Start a Wps session on a bss

`POST /api/v9/wifi/wps/start/`

Once you identified a `WifiWpsCandidate` eligible for wps you can start a `WifiWpsSession` on the associated bss. In return you'll get the id of the created session.

Example request:

```
POST /api/v9/wifi/wps/start/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
      "bssid":"14:0C:76:87:04:38"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success": true,
      "result": 1
}
```

Stop a Wps session

This lets you close an open session

Example request:

```
POST /api/v9/wifi/wps/stop/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
      "session_id": 1
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success": true
}
```

List the Wps session

`GET /api/v9/wifi/wps/sessions/`

Get the list of `WifiWpsSession`

Example request:

```
GET /api/v9/wifi/wps/sessions/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success": true,
      "result": [
            {
                  "mac": "00:00:00:00:00:00",
                  "end_date": 1516012651,
                  "ssid": "r0ro 5G",
                  "active": false,
                  "id": 1,
                  "start_date": 1516012531,
                  "result": "failed_timeout",
                  "bss_uuid": "6a55ea3d-29fa-4bd9-b1e3-22a49a3ca134"
            }
      ]
}
```

Clear all Wps Sessions

`DELETE /api/v9/wifi/wps/sessions/`

Clear all the existing wps sessions

Example request:

```
DELETE /api/v9/wifi/wps/sessions/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Wifi guest

This api lets you create “custom key” (guest Wi-Fi access) that can be used on your existing bss to allow someone to connect to your Wi-Fi network without knowing your actual Wi-Fi password.

When creating a “custom key” you can select if the associated access should be restricted to WAN only access, or if the guest can also access your local network. You can also define how long the access should be available.

A dedicated Wi-Fi network is created for guest usage, and the SSID can be configured. Note that network will only be running when you have wifi running and a custom key created.

Wifi Custom Key config

WifiCustomKeyConfig

- **ssid** string
  The name of the dedicated wifi network
- **ssid_read_only** bool ***Read-only***
  When true, the SSID name cannot be changed.
- **hide_ssid** bool ***Read-only***
  When true, the SSID used for guest network is hidden.
- **encryption** enum ***Read-only***
  Encryption used for guest Wi-Fi network.

Get or change the dedicated ap config

GET /api/v14/wifi/custom_keys/config/

Get the dedicated guest config as a WifiCustomKeyConfig

Example request:

```
GET /api/v14/wifi/custom_keys/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success":true,
      "result": {
            "ssid":"Freebox-C0001B-guest",
            "ssid_read_only":false,
            "hide_ssid":false,
            "encryption":"wpa2_psk"
      }
}
```

PUT /api/v14/wifi/custom_keys/config/

Set the dedicated guest AP config. Only SSID or global enabled switch.

Example request:

```
PUT /api/v9/wifi/custom_keys/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
      "ssid": "my-guest-network-ssid"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
      "success":true,
      "result": {
            "ssid":"my-guest-network-ssid",
            "ssid_read_only":true,
            "hide_ssid":false,
            "encryption":"wpa2_psk"
      }
}
```

Wifi Custom Key object

WifiCustomKey has the following attributes:

WifiCustomKeyHost

- **hostname** string ***Read-only***
  host name
- **host** LanHost ***Read-only***
  optional host information from Lan Browser (if available)

WifiCustomKeyParams

- **description** string
  description of the custom key
- **key** string
  Wi-Fi password for this custom access
- **max_use_count** int
  Number of different hosts that can connect to this network (maximum 127) 0 has special meaning, it means unlimited number of users.
- **duration** int
  Number of seconds before the custom access is revoked
- **access_type** enum

access_type	Description
full	stations will get full access to local network + internet
net_only	stations connected using this custom key will be isolated and won’t have access to local network devices

WifiCustomKey

- **id** int ***Read-only***
  custom key id
- **remaining** int ***Read-only***
  time remaining before the access (seconds) if 0 then it does not expire
- **params** WifiCustomKeyParams
  custom key parameters
- **users[]** array of WifiCustomKeyHost ***Read-only***
  list of hosts that used the custom key

Get the list of wifi custom key

GET /api/v9/wifi/custom_key/

Get the list of WifiCustomKey

Example request:

```
GET /api/v9/wifi/custom_key/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
        "success": true,
        "result": [
                {
                        "id": 8,
                        "remaining": 86376,
                        "params": {
                                "max_use_count": 100,
                                "description": "soirée mario kart",
                                "duration": 86400,
                                "access_type": "full",
                                "key": "YY5Sg74W3VNxrmfwAz7aCY7OVqRVG2JN"
                        }
                }
        ]

}
```


Getting a particular wifi custom key

GET /api/v9/wifi/custom_key/{key_id}

Returns the requested WifiCustomKey properties

Example request:

```
GET /api/v9/wifi/custom_key/8 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
        "success": true,
        "result": {
                "id": 8,
                "remaining": 86376,
                "params": {
                        "max_use_count": 100,
                        "description": "soirée mario kart",
                        "duration": 86400,
                        "access_type": "full",
                        "key": "YY5Sg74W3VNxrmfwAz7aCY7OVqRVG2JN"
                }
        }
}
```


Delete a wifi custom key

DELETE /api/v9/wifi/custom_key/{key_id}

Delete the WifiCustomKey with the given id It will automatically disconnect any connected stations using this custom key

Example request:

```
DELETE /api/v9/wifi/custom_key/8 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Create a new wifi custom key

POST /api/v9/wifi/custom_key/

Create a new the WifiCustomKey Post the parameters of the custom key

Example request:

```
POST /api/v9/wifi/custom_key HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
      "description": "zuper",
      "key": "rzR18eLeh6D8B7n1DtMbeDxwo2d4O9fB",
      "max_use_count": "100",
      "duration":86400,
      "access_type":"net_only"
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
      "success":true,
      "result": {
            "id": 11,
            "remaining": 86399,
            "params": {
                  "max_use_count": 100,
                  "description": "zuper",
                  "duration": 86400,
                  "access_type": "full",
                  "key":"rzR18eLeh6D8B7n1DtMbeDxwo2d4O9fB"
            }
      }
}
```


Temporary disabling Wifi

This API lets you disable some wifi bands for a given amount of time. This is useful to pair IOT devices that only supports some bands.

Temporary disable object

TemporaryWifiDisable has the following attributes:

`TemporaryWifiDisable`

`duration` `int` ***Write-only***

temporary disable duration

`keep` `enum` ***Write-only***

specify a wifi band to keep active

keep	Description
2d4g	keep only 2,4Ghz band active
5g	keep only 5GHz bands active
6g	keep only 6GHz band active

`remaining` `int` ***Read-only***

remaining seconds the wifi is temporarily disabled. Set to 0 to stop the temporary wifi disabling period.

Get temporary disable state

`GET /api/v13/wifi/temp_disable`

Get the state of temporary wifi disable.

Example request:

```
GET /api/v13/wifi/temp_disable HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "remaining": 267
    }
}
```

`POST /api/v13/wifi/temp_disable`

Start or stop a temporary wifi disabling period

Example request:

```
POST /api/v13/wifi/temp_disable HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
    "duration": 1200,
    "keep": "2d4g"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Multi Link Operation (MLO)

For a given BSS you can configure with which bands it will try to participate in an MLD. Whatever the configuration is, the operational state may be different if the BSS on the partner AP is unavailable (disabled or no EHT) or does not have the right parameters (not using shared params or wrong security)

Available partner

To get the available AP partner of a BSS use the mlo/allowed_comb api to return a list of possible combinations:

`GET /api/v14/wifi/bss/{id}/mlo/allowed_comb`

Get the allowed phy combination for a BSS

Example request:

```
GET /api/v14/wifi/bss/02:00:00:00:00:00/mlo/allowed_comb HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```

```

{

“success”: true, “result”: [

[0, 1], [0]

]

}

MLO configuration object

`WifiMLOConfiguration`

`partners` `[int]`

List of phys participating in the MLD for the BSS An empty array means MLO is disabled An array with only the BSS's AP index in it means SLO (single link mode) The allowed combinations are retrieved by the mlo/allowed_comb api.

Getting the MLO config

To get the currently configured partners of a BSS mlo/config. It will return the current `WifiMLOConfiguration` for this BSS

`GET /api/v14/wifi/bss/{id}/mlo/config`

Get the current `WifiMLOConfiguration` for the BSS

Example request:

```
GET /api/v14/wifi/bss/02:00:00:00:00:00/mlo/config HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true,
   "result": {
      partners: [ 0, 1 ]
```

```
   }
}
```

Changing the MLO config

To update the MLO confuguration put a new `WifiMLOConfiguration` at mlo/config. Please note that only combinations from mlo/allowed_comb can be used for the 'partners' field

PUT /api/v9/wifi/config/

Update the `WifiGlobalConfig`

Example request:

```
PUT /api/v14/wifi/bss/02:00:00:00:00:00/mlo/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "partners": [ 0, 1 ]
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
       partners: [ 0, 1 ]
    }
}
```

System

System Config

SystemConfig has the following attributes:

SystemConfig

- **firmware_version** string ***Read-only***
  freebox firmware version
- **mac** string ***Read-only***
  freebox mac address
- **serial** string ***Read-only***
  freebox serial number
- **uptime** string ***Read-only***
  readable freebox uptime
- **uptime_val** int ***Read-only***
  freebox uptime (in seconds)
- **board_name** string ***Read-only***
  freebox hardware revision
- **box_authenticated** bool ***Read-only***
  is the box authenticated ("étape 6")
- **disk_status** enum ***Read-only***
  the internal disk status

Value	Description
not_detected	The disk as not been detected
disabled	The disk is disabled
initializing	The disk is initializing
error	The disk failed to mount
active	The disk is ready

- **usb3_enable** bool
  enable USB3 (on supported platforms)
- **user_main_storage** string
  The label of the storage partition to use for user data. (Matches the label of the `DiskPartition`) In case of 'light' box flavor, it must be set by to a permanently attached external storage
- **user_storage_powered** bool ***Read-only***
  Indicate whether the user storage is powered or not
- **expansions[]** array of **SystemConfigSensor** ***Read-only***
  List of thermal sensors on the system
- **model_info** **SystemModelInfo** ***Read-only***
  Device informations
- **fans[]** array of **SystemConfigFan** ***Read-only***
  List of fans on the system
- **expansions[]** array of **SystemConfigExpansion** ***Read-only***
  List of expansions slots modules

SystemModelInfo

- **name** enum ***Read-only***

name	Description
fbxgw-r1/full	Freebox Server (v6) revision 1
fbxgw-r2/full	Freebox Server (v6) revision 2
fbxgw-r1/mini	Freebox Mini revision 1
fbxgw-r2/mini	Freebox Mini revision 2
fbxgw-r1/one	Freebox One revision 1
fbxgw-r2/one	Freebox One revision 2
fbxgw7-r1/full	Freebox v7 revision 1
fbxgw8-r1/full	Freebox v8 revision 1
fbxgw9-r1/full	Freebox v9 revision 1

- **pretty_name** string ***Read-only***
  Display name for the box model
- **has_expansions** bool ***Read-only***
  if present and true, the box has expansions
- **has_lan_sfp** bool ***Read-only***

if present and true, the box has an SFP port for lan

`has_dect` `bool` ***Read-only***
if present and true, the box has a DECT base station

`has_home_automation` `bool` ***Read-only***
if present and true, the box has a Home automation module

`has_femtocell_exp` `bool` ***Read-only***
if present and true, the box has a femtocell expansion slot

`has_fixed_femtocell` `bool` ***Read-only***
if present and true, the box has an internal femtocell

`has_vm` `bool` ***Read-only***
if present and true, the box supports virtual machines

`has_dsl` `bool` ***Read-only***
if present and true, the box supports DSL

`has_standby` `bool` ***Read-only***
if present and true, the box supports standby

`has_eco_wifi` `bool` ***Read-only***
if present and true, the box supports Eco-WiFi

`has_wop` `bool` ***Read-only***
if present and true, the box supports Wake-On-PON

`has_led_strip` `bool` ***Read-only***
if present and true, the box has a LED strip

`has_status_led` `bool` ***Read-only***
if present and true, the box has a status LED

`has_usb3_enable` `bool` ***Read-only***
if present and true, the box supports disabling USB3

`SystemConfigSensor`

`id` `string` ***Read-only***
sensor id

`name` `string` ***Read-only***
sensor display name

`value` `int` ***Read-only***
sensor current value (in celsius degree)

`SystemConfigFan`

`id` `string` ***Read-only***
fan id

`name` `string` ***Read-only***
fan display name

`value` `int` ***Read-only***
fan current speed (RPM)

`SystemConfigExpansion`

`slot` `int` ***Read-only***
expansion slot id

`probe_done` `bool` ***Read-only***
has the module presence been probed yet

`present` `bool` ***Read-only***
has an expansion module been detected in the slot

`supported` `bool` ***Read-only***
is the module supported in this slot

`bundle` `string` ***Read-only***
module serial number

`type` `enum` ***Read-only***
module type

Value	Description
unknown	unknown module
dsl_lte	xDSL + LTE
dsl_lte_external_antennas	xDSL + LTE with external antennas switch
ftth_p2p	FTTH P2P
ftth_pon	FTTH PON
security	Security module

System Config V5 (DEPRECATED)

SystemConfigV5 has the following attributes:

`SystemConfigV5`

`firmware_version` `string` ***Read-only***
freebox firmware version

`mac` `string` ***Read-only***
freebox mac address

`serial` `string` ***Read-only***
freebox serial number

`uptime` `string` ***Read-only***
readable freebox uptime

`uptime_val` `int` ***Read-only***
freebox uptime (in seconds)

`board_name` `string` ***Read-only***
freebox hardware revision

`temp_cpum` `int` ***Read-only***
temp cpum (°C)

`temp_sw` `int` ***Read-only***
temp sw (°C)

`temp_cpub` `int` ***Read-only***

temp cpub (°C)

`fan_rpm` `int` ***Read-only***

fan rpm

`box_authenticated` `bool` ***Read-only***

is the box authenticated (“étape 6”)

`disk_status` `enum` ***Read-only***

the internal disk status

Value	Description
not_detected	The disk as not been detected
disabled	The disk is disabled
initializing	The disk is initializing
error	The disk failed to mount
active	The disk is ready

`box_flavor` `enum` ***Read-only***

the box ‘flavor’ for a given model

Value	Description
full	The box has an internal storage
light	The box has no internal storage

`user_main_storage` `string`

The label of the storage partition to use for user data. (Matches the label of the `DiskPartition`) In case of 'light' box flavor, it must be set by to a permanently attached external storage

System API

Get the current system info [UNSTABLE]

Current version (api >= v6)

`GET /api/v8/system/`

Get the `SystemConfig`

Example request:

```
GET /api/v8/system/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "mac": "34:27:92:60:0B:9E",
        "sensors": [{
                        "id": "t2",
                        "name": "Température 2",
                        "value": 47
                },
                {
                        "id": "t1",
                        "name": "Température 1",
                        "value": 45
                },
                {
                        "id": "t3",
                        "name": "Température 3",
                        "value": 42
                },
                {
                        "id": "cpu_cp_slave",
                        "name": "Température CPU CP Slave",
                        "value": 72
                },
                {
                        "id": "cpu_cp_master",
                        "name": "Température CPU CP Master",
                        "value": 72
                },
                {
                        "id": "cpu_ap",
                        "name": "Température CPU",
                        "value": 64
                }
        ],
        "model_info": {
                "pretty_name": "Freebox v7 (r1)",
                "has_expansions": true,
                "name": "fbxgw7-r1/full",
                "has_lan_sfp": true,
                "has_dect": true,
                "internal_hdd_size": 0,
                "has_home_automation": true,
                "wifi_type": "2d4_5g_5g"
        },
        "fans": [{
                        "id": "secondary-fan",
                        "name": "Ventilateur 2",
                        "value": 1725
                },
                {
                        "id": "main",
                        "name": "Ventilateur 1",
                        "value": 1739
                }
        ],
        "expansions": [{
                        "type": "security",
                        "present": true,
                        "slot": 1,
                        "probe_done": true,
                        "supported": true,
                        "bundle": "985700J183900112"
                },
                {
                        "type": "ftth_p2p",
                        "present": true,
```

```
                        "slot": 2,
                        "probe_done": true,
                        "supported": true,
                        "bundle": "959300V181500003"
                    }
            ],
            "box_authenticated": true,
            "disk_status": "active",
            "uptime": "2 heures 11 minutes 32 secondes",
            "uptime_val": 7892,
            "user_main_storage": "Disque 1",
            "board_name": "fbxgw7r",
            "serial": "95760J1J83400107",
            "firmware_version": "6.6.6"
        }
    }
```


Old version (api < v5)

`GET /api/v8/system/`

Get the SystemConfigV5

Example request:

```
GET /api/v8/system/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
      "success": true,
      "result": {
            "mac": "F4:CA:E5:5C:EA:14",
            "box_flavor": "light",
            "temp_cpub": 63,
            "disk_status": "active",
            "box_authenticated": true,
            "board_name": "fbxgw1r",
            "fan_rpm": 1832,
            "temp_sw": 52,
            "uptime": "6 jours 22 heures 9 minutes 46 secondes",
            "uptime_val": 598186,
            "user_main_storage": "Disque 1",
            "temp_cpum": 62,
            "serial": "805400T144100853",
            "firmware_version": "6.6.6"
      }
}
```


Reboot the system

`POST /api/v8/system/reboot/`

Reboot the Freebox

Example request:

```
POST /api/v8/system/reboot/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Shutdown the system

`POST /api/v11/system/shutdown/`

Shutdown the Freebox

Example request:

```
POST /api/v11/system/shutdown/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


VPN Server [UNSTABLE]

The VPN Server API allows you to control the Freebox VPN Server

VPN Server Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
inval	invalid parameters
exist	entry already exists
noent	invalid id
nomem	internal error
unsupp	not supported
inuse	resource in use
busy	resource is busy
ioerror	internal error
size	too many elements

VPN Server List

VPN Server Object

`VPNServer`

VPNServer has the following attributes:

`name` `string` ***Read-only***
VPN server name (id)

`type` `enum` ***Read-only***
VPN server type

type	Description
ipsec	IPsec IKEv2 server
pptp	PPTP VPN server
openvpn	OpenVPN server
wireguard	WireGuard server

`state` `enum` ***Read-only***
server state

state	
stopped	
starting	
started	
stopping	
error	

`connection_count` `int` ***Read-only***
number of active connections

`auth_connection_count` `int` ***Read-only***
number of active connections that have passed authentication

VPN Server List API

`GET /api/v8/vpn/`
Get the list of VPNServer

Example request:

```
GET /api/v8/vpn/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "state": "stopped",
            "type": "pptp",
            "name": "pptp",
            "connection_count": 0,
            "auth_connection_count": 0
        },
        {
            "state": "stopped",
            "type": "openvpn",
            "name": "openvpn_routed",
            "connection_count": 0,
            "auth_connection_count": 0
        },
        {
            "state": "stopped",
            "type": "openvpn",
            "name": "openvpn_bridge",
            "connection_count": 0,
            "auth_connection_count": 0
        },
        {
            "state": "stopped",
            "type": "wireguard",
            "name": "wireguard",
            "connection_count": 0,
            "auth_connection_count": 0
        }
    ]
}
```


VPN Server Config

`VPNPPTPConfig`
VPNServerConfig has the following attributes:

`mppe` `enum`

mppe	Description
disable	disable mppe
require	require mppe
require_128	require 128 bits mppe

`allowed_auth` `dict`
allowed authentication methods dictionnary with following entries:

- pap
- chap
- mschapv2

values are booleans.

`VPNOpenVpnConfig`

`cipher` `enum`

cipher	
blowfish	
aes128	
aes256	

cipher	
chacha20poly1305	

`disable_fragment` **bool**
disable fragment configuration option

`use_tcp` **bool**
use TCP instead of UDP

VPNWireGuardConfig

`mtu` **int**
wireguard device MTU. Value must be between 512 and 1420.

VPNIPSecAuthMode

`id_source` **enum**
source of the connection id

id_source	
custom	

`id_custom` **string**
value of the source id when id_source is custom

VPNIPSecConfig

`ike_version` **int** ***Read-only***
IKE protocol version

`auth_modes[]` **array of** VPNIPSecAuthMode ***Read-only***
map of supported auth modes, currently only psk is supported

VPNServerConfig

`id` **string** ***Read-only***
VPN server id

`type` **enum** ***Read-only***
VPN server type

type	Description
pptp	PPTP VPN server
openvpn	OpenVPN server
ipsec	IPsec IKEv2 server
wireguard	WireGuard server

`enabled` **bool**
is the VPN server enabled

`enable_ipv4` **bool**
enable IPv4 on this server

NOTE: Not relevant for openvpn_bridge, pptp and wireguard

`enable_ipv6` **bool**
enable IPv6 on this server

NOTE: Not relevant for openvpn_bridge, pptp and wireguard

`port` **int**
the server port

NOTE: you can only edit the server port when type is openvpn or wireguard

`min_port` **int** ***Read-only***
This field indicate the minimum possible value for port (see ConnectionStatus ipv4_port_range)

`max_port` **int** ***Read-only***
This field indicate the maximum possible value for port (see ConnectionStatus ipv4_port_range)

`port_ike` **int**
IPSec ike server port

NOTE: only present for ipsec server

`port_nat` **int**
IPSec nat server port

NOTE: only present for ipsec server

`conf_pptp` VPNPPTPConfig
only available when type is PPTP

`conf_openvpn` VPNOpenVpnConfig
only available when type is OpenVPN

`conf_ipsec` VPNIPSecConfig
only available when type is IPsec

`conf_wireguard` VPNWireGuardConfig
only available when type is WireGuard

`ip_start` **string** ***Read-only***
start of the IP range that will be used to give clients an IP

`ip_end` **string** ***Read-only***
end of the IP range that will be used to give clients an IP

`ip6_start` **string** ***Read-only***
start of the IPv6 range that will be used to give clients an IPv6

`ip6_end` **string** ***Read-only***
end of the IPv6 range that will be used to give clients an IPv6

VPN Server Config API

Get a VPN config

`GET /api/v8/vpn/{vpn_id}/config/`

Get the VPNServerConfig

Example request:

```
GET /api/v8/vpn/openvpn_routed/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": false,
        "port": 1194,
        "conf_openvpn": {
            "cipher": "aes128"
        },
        "id": "openvpn_routed",
        "ip_start": "192.168.27.65",
        "ip_end": "192.168.27.95",
        "type": "openvpn"
    }
}
```


Update the VPN configuration

`PUT /api/v8/vpn/openvpn_routed/config/`

Update the VPNServerConfig

Example request:

```
PUT /api/v8/vpn/openvpn_routed/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
   "conf_openvpn": {
      "cipher": "blowfish"
    }
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": false,
        "port": 1194,
        "conf_openvpn": {
            "cipher": "blowfish"
        },
        "id": "openvpn_routed",
        "ip_start": "192.168.27.65",
        "ip_end": "192.168.27.95",
        "type": "openvpn"
    }
}
```


VPN Server User API

VPN users are common to all VPN servers.

VPN Server User Object

`VPNUser`

VPNUser has the following attributes:

`login` **string**

VPN user login

`type` **enum**

VPN user type

type	
standard	
wireguard	

`password` **string** ***Write-only***

VPN user password (length must be between 8 and 32)

`password_set` **bool** ***Read-only***

True if a password was provided for this user

`ip_reservation` **ipv4**

You can specify the IP you want to assign to this user. If you don't want to use a specific IP pass an empty string or omit this property. This field is required if the type property is set to 'wireguard'.

The IP must be in the VPN range (see ip_start, ip_end).

`conf_wireguard`

This field is present only if the type property is set to 'wireguard'.

`keepalive` **int**

Interval in seconds at which keepalive packets are sent.

`psk` **bool**

Enable optional preshared-key.

VPN Server User List

`GET /api/v8/vpn/user/`

Get the list of VPNUser

Example request:

```
GET /api/v8/vpn/user/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "ip_reservation": "",
            "type": "standard",
            "login": "test-1392677633-np",
            "password_set": false
        },
        {
            "ip_reservation": "",
```

```
            "type": "standard",
            "login": "test-1392677633",
            "password_set": true
        },
        {
            "ip_reservation": "192.168.27.68",
            "type": "wireguard",
            "login": "test-1392677633-wg",
            "password_set": false,
            "conf_wireguard": {
                "keepalive": 10,
                "psk": false
            }
        }
    ]
}
```


Get a VPN user

`GET /api/v8/vpn/user/{login}`

Gets the VPNUser with the given login

Example request:

```
GET /api/v8/vpn/user/test-1392677633-np HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "ip_reservation": "",
        "login": "test-1392677633-np",
        "type": "standard",
        "password_set": false
    }
}
```


Add a VPN User

`POST /api/v8/vpn/user/`

Creates a new VPNUser.

Example request:

```
POST /api/v8/vpn/user/ HTTP/1.1
Host: mafreebox.freebox.fr

{
  "login": "vpnuser01",
  "type": "standard",
  "password": "thisisasecret",
  "ip_reservation": "192.168.27.69"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "ip_reservation": "192.168.27.69",
        "login": "vpnuser01",
        "password_set": true
    }
}
```


Delete a VPN User

`DELETE /api/v8/vpn/user/{login}`

Deletes the VPNUser

Example request:

```
DELETE /api/v8/vpn/user/vpnuser01 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Update a VPN User

`PUT /api/v8/vpn/user/{login}`

Updates the VPNUser task with the given login

Example request:

```
PUT /api/v8/vpn/user/test-1392677633-np HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
    "password": "donttellanyone"
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "ip_reservation": "",
        "login": "test-1392677633-np",
        "password_set": true
    }
}
```


VPN IP Pool

Get the VPN server IP pool reservations

`GET /api/v8/vpn/ip_pool/`

Gets the VPNUser with the given login

Example request:

```
GET /api/v8/vpn/ip_pool/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "ip_start": "192.168.27.65",
        "ip_end": "192.168.27.95",
        "reservations": [
            {
                "login": "test",
                "ip": "192.168.27.69"
            }
        ]
    }
}
```

VPN Server Connection API

This API allows listing the active connections to the VPN server

VPN Connection Object

`VPNConnection`

VPNConnection has the following attributes:

`id` `string` ***Read-only***
connection id

`vpn` `strong` ***Read-only***
related VPN server id

`user` `string` ***Read-only***
user login

`authenticated` `bool` ***Read-only***
is the connection authenticated

`auth_time` `int` ***Read-only***
timestamp of the authentication

`src_ip` `ipv4` ***Read-only***
connection source IP address

`src_port` `int` ***Read-only***
connection source port

`local_ip` `int` ***Read-only***
attributed IP address from VPN adress pool

`rx_bytes` `int` ***Read-only***
rx bytes

`tx_bytes` `int` ***Read-only***
tx bytes

Get the list of connections

`GET /api/v8/vpn/connection/`

Get the list of VPNUser

Example request:

```
GET /api/v8/vpn/user/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
        {
            "rx_bytes": 94,
            "authenticated": true,
            "tx_bytes": 94,
            "user": "test",
            "id": "pptp-2",
            "vpn": "pptp",
            "src_ip": "93.184.216.119",
            "auth_time": 1392895603,
            "local_ip": "192.168.27.65"
        }
    ]
}
```

Close a given connection

`DELETE /api/v8/vpn/connection/{id}`

Deletes the VPNUser

Example request:

```
DELETE /api/v8/vpn/connection/pptp-2 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

VPN User configuration file API

For OpenVPN and WireGuard servers, you can download a configuration file that will be used to configure the VPN client

Donwload a user configuration file

`GET /api/v8/vpn/download_config/{server_name}/{login}/{fmt}`

Download an configuration file for the given server and login The “fmt” field must be set to either “plain” or “json”.

WARNING: each time you download a new OpenVPN configuration file for a given user, you invalidate previous configuration file emitted for this user

Example request:

```
GET /api/v8/vpn/download_config/openvpn_routed/test/plain HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Date: Thu, 20 Feb 2014 13:14:01 GMT
Server: nginx
Content-Type: application/x-openvpn-profile
Content-Disposition: attachment; filename="config_openvpn_routed_test.ovpn"
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Transfer-Encoding: chunked

[ ... ]
```

VPN Client [UNSTABLE]

The VPN Client API allows you to control the Freebox VPN Client

VPN Client Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
inval	invalid parameters
nomem	internal error
ioerror	internal error
nodev	invalid device
noent	invalid id
netdown	network is not available
exist	entry already exists
busy	resource is busy

VPN Client Configuration

VPN Client Configuration Object

`VPNClientConfig`

VPNClientConfig has the following attributes:

`id` `string` ***Read-only***
VPN config id

`description` `string`
VPN description

`type` `enum`
VPN server type

type	Description
pptp	PPTP VPN server
openvpn	OpenVPN server
wireguard	WireGuard server

`active` `bool`
is this configuration active. Only one configuration is active at a time.

`conf_pptp` VPNClientConfigPPTP
only available when type is PPTP

`conf_wireguard` VPNClientConfigWireGuard
only available when type is WireGuard

`VPNClientConfigPPTP`

VPNClientConfigPPTP has the following attributes:

`remote_host` `string`
remote host IP or name

`username` `string`
VPN username

`password` `string` ***Write-only***
VPN password

`mppe` `enum`

mppe	Description
disable	disable mppe
require	require mppe
require_128	require 128 bits mppe

`allowed_auth` `dict`
allowed authentication methods dictionary with following keys:

- eap
- pap
- chap
- mschap
- mschapv2

values are booleans.

`VPNClientConfigWireGuard`

VPNClientConfigWireGuard has the following attributes:

`remote_addr` `string`
remote host IP

`remote_port` `int`

remote host port

remote_public_key string
remote host public key

remote_preshared_key string
optional preshared key

local_priv_key string
local private key

local_addr[] array of **VPNClientConfigWireGuardIP**
IPs to assign to the local interface.

dns[] array of string
list of strings containing IPs of DNS servers to use. Both IPv4 and IPv6 are supported.

VPNClientConfigWireGuardIP

ip string
string representation of an IPv4 or IPv6 address

len int
prefix length associated with the IP address

Get VPN Client configuration list

GET /api/v8/vpn_client/config/

Get the list of VPNClientConfig

Example request:

```
GET /api/v8/vpn_client/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "type": "pptp",
            "description": "test vpn2",
            "active": true,
            "id": "vpn0",
            "conf_pptp": {
                "mppe": "require",
                "username": "freeuser",
                "remote_host": "vpnhost.example.org",
                "allowed_auth": {
                    "eap": false,
                    "mschap": false,
                    "mschapv2": true,
                    "chap": false,
                    "pap": false
                }
            }
        },
        {
            "type": "pptp",
            "description": "test vpn1",
            "active": false,
            "id": "vpn1",
            "conf_pptp": {
                "mppe": "require",
                "username": "testuser",
                "remote_host": "example.org",
                "allowed_auth": {
                    "eap": false,
                    "mschap": false,
                    "mschapv2": true,
                    "chap": false,
                    "pap": false
                }
            }
        }
        {
            "type": "wireguard",
            "description": "test vpn2",
            "active": false,
            "id": "vpn2",
            "conf_wireguard": {
                "local_addr": [{"ip":"198.51.100.10", "len":24}],
                "local_priv_key": "TdbS1Y0RHZ6rRNSxlEUssD/pnRDfrHMFfJPLl5icvQg=",
                "dns": ["198.51.100.53", "2001:db8:100::53"],
                "mtu": 1420,
                "remote_public_key": "QZnLR0TYPbPbhfVWeLVRf1zsPC0JXG/woVmsmEkgsw8=",
                "remote_addr": "192.0.2.1",
                "remote_port": 51820,
                "remote_preshared_key": ""
            }
        }
    ]
}
```


Get a VPN client config

GET /api/v8/vpn_client/config/{id}

Get the VPNClientConfig

Example request:

```
GET /api/v8/vpn_client/config/vpn0 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "type": "pptp",
        "description": "test vpn2",
        "active": true,
        "id": "vpn0",
```

```
        "conf_pptp": {
            "mppe": "require",
            "username": "freeuser",
            "remote_host": "vpnhost.example.org",
            "allowed_auth": {
                "eap": false,
                "mschap": false,
                "mschapv2": true,
                "chap": false,
                "pap": false
            }
        }
    }
}
```


Add a VPN client configuration

POST /api/v8/vpn_client/config/

Creates a new VPNClientConfig.

Example request:

```
POST /api/v8/vpn_client/config/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "type": "pptp",
   "description": "test pptp",
   "active": false,
   "conf_pptp": {
      "mppe": "require",
      "username": "fbxtest",
      "password": "",
      "remote_host": "test.example.org",
      "allowed_auth": {
         "mschapv2": true
      }
   }
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "type": "pptp",
        "description": "test pptp",
        "active": false,
        "id": "vpn2",
        "conf_pptp": {
            "password": "",
            "mppe": "require",
            "username": "fbxtest",
            "remote_host": "test.example.org",
            "allowed_auth": {
                "eap": false,
                "mschap": false,
                "mschapv2": true,
                "chap": false,
                "pap": false
            }
        }
    }
}
```


Delete a VPN client Configuration

DELETE /api/v8/vpn_client/config/{id}

Deletes the VPNClientConfig

Example request:

```
DELETE /api/v8/vpn_client/config/vpn2 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Update the VPN client configuration

PUT /api/v8/vpn_client/config/{id}

Update the VPNServerConfig

Example request:

```
PUT /api/v8/vpn_client/config/vpn0 HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
   "active": false
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "type": "pptp",
        "description": "test vpn2",
        "active": false,
        "id": "vpn0",
        "conf_pptp": {
            "mppe": "require",
            "username": "freeuser",
            "remote_host": "vpnhost.example.org",
            "allowed_auth": {
                "eap": false,
```

```
                "mschap": false,
                "mschapv2": true,
                "chap": false,
                "pap": false
            }
        }
    }
}
```

VPN Client Status

VPN Client Status Object

`VPNClientStatus`

VPNClientStatus has the following attributes:

`enabled` `bool` ***Read-only***
is VPN client enabled

`active_vpn` `string` ***Read-only***
active VPN id

`active_vpn_description` `string` ***Read-only***
active VPN description

`type` `enum` ***Read-only***
active VPN type

type	Description
pptp	PPTP VPN server
openvpn	OpenVPN server
wireguard	WireGuard server

`state` `enum` ***Read-only***

state	Description
waiting_wan	waiting for wan connection
going_up	connecting
up	connected
going_down	disconnecting
down	disconnected

`last_up` `int` ***Read-only***
timestamp of last successful connection

`last_try` `int` ***Read-only***
timestamp of last connection attempt

`next_try` `int` ***Read-only***
seconds left until next connection attempt

`last_error` `enum` ***Read-only***

last_error	Description
none	no error
internal	internal error
authentication_failed	wrong credentials
auth_failed	wrong credentials
resolv_failed	invalid host name
connect_timeout	connection timeout
connect_failed	connection failed
setup_control_failed	PPTP session negotiation failure
setup_call_failed	PPTP session failure
protocol	protocol error
remote_terminated	connection closed by remote peer
remote_disconnect	connection closed by remote peer

`stats` `VpnClientStats` ***Read-only***
connection statistics

`IPv4` `VpnClientIpInfo` ***Read-only***
connection IPv4 information

`VpnClientStats`

`rate_up` `int` ***Read-only***
current upload rate (in byte/s)

`rate_down` `int` ***Read-only***
current download rate (in byte/s)

`bytes_up` `int` ***Read-only***
total bytes uploaded

`bytes_down` `int` ***Read-only***
total bytes downloaded

`VpnClientIpInfo`

`config_valid` `bool` ***Read-only***
is the configuration valid

`ip_mask` `dict` ***Read-only***
assigned IP and netmask

`domain` `string` ***Read-only***
provided domain

`gateway` `IPv4` ***Read-only***
provided gateway

`dns[]` `array of ipv4` ***Read-only***
list of dns servers

`provider` `enum` ***Read-only***
ip_mask source

provider	Description
none	none
static	static IP configuration
ppp	ppp
dhcp	DHCP server

`routes` `list` ***Read-only***

list of provided routes

`dhcp` `dict` ***Read-only***

DHCP status information

Get the VPN client status

`GET /api/v8/vpn_client/status`

Get the VPNClientStatus

Example request:

```
GET /api/v8/vpn_client/status HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "enabled": true,
        "type": "pptp",
        "last_error": "none",
        "active_vpn_description": "test vpn",
        "last_try": 1392904509,
        "state": "up",
        "stats": {
            "rate_up": 0,
            "bytes_down": 94,
            "bytes_up": 94,
            "rate_down": 0
        },
        "active_vpn": "vpn1",
        "next_try": 0,
        "last_up": 1392904510,
        "ipv4": {
            "routes": { },
            "config_valid": true,
            "ip_mask": {
                "ip": "192.168.27.65",
                "mask": "255.255.255.255"
            },
            "provider": "ppp",
            "dhcp": {
                "state": "down",
                "renew_remaining": 0,
                "dhcp_options": { },
                "lease_remaining": 0,
                "lease_time": 0,
                "rebind_remaining": 0,
                "server_id": 0
            },
            "dns": [
                "212.27.38.253"
            ],
            "domain": "",
            "gateway": "212.27.38.253"
        }
    }
}
```


Get the VPN client logs

`GET /api/v8/vpn_client/log`

Example request:

```
GET /api/v8/vpn_client/log HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": "2014-02-20 14:55:10 dbg: ppp: pppd: sent [ ... ] "
}
```


Diagnostics

Slowness

The slowness API allow you to execute diagnostics on a selected host to detect a potential causes of degradation of the throughput.

Slowness Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
inval	invalid parameters
nodev	invalid device id
nohost	invalid host GID or not found
noconn	WAN connection is down
netdown	link with host is down
erunning	API is already running
internal	system internal error

Slowness API

Get the last result of a given host

Downloads

Download

With the download API you can control the download queue of the Freebox. The Freebox supports downloads from HTTP, FTP, Magnet link, `.torrent` files and newsgroups (NNTP). Each download task is represented by a `Download` object.

Download Errors

When attempting to access the download API, you may encounter the following errors:

error_code	Description
task_not_found	No task was found with the given id
invalid_operation	Attempt to perform an invalid operation
invalid_file	Error with the download file (invalid format ?)
invalid_url	URL is invalid
not_implemented	Method not implemented
out_of_memory	No more memory available to perform the requested action
invalid_task_type	The task type is invalid
hibernating	The downloader is hibernating
need_bt_stopped_done	This action is only valid for Bittorrent task in stopped or done state
bt_tracker_not_found	Attempt to access an invalid tracker object
too_many_tasks	Too many tasks
invalid_address	Invalid peer address
port_conflict	Port conflict when setting config
invalid_priority	Invalid priority
internal_error	Internal error
ctx_file_error	Failed to initialize task context file (need to check disk)
exists	Same task already exists
port_outside_range	Incoming port is not available for this customer (see `ConnectionStatus` ipv4_port_range)

Download Task / TaskFile Errors

Each download task can encounter one of the following errors:

Error	Description
none	No error
internal	Internal error
disk_full	The disk is full
unknown	Unknown error
parse_error	Parse error
http_301	HTTP 301 error
http_400	HTTP 400 error
http_401	
http_402	
http_403	
http_404	
http_405	
http_406	
http_407	
http_408	
http_409	
http_410	
http_411	
http_412	[…]
http_413	
http_414	
http_415	
http_416	
http_417	
http_422	
http_423	
http_424	
http_425	
http_426	
http_427	
http_428	
http_429	
http_430	
http_431	
http_4xx	Other 4xx HTTP errors
http_500	HTTP 500 error
http_501	
http_502	

Error	Description
http_503	
http_504	
http_505	
http_506	[…]
http_507	
http_508	
http_509	
http_510	
http_511	
http_5xx	Other 5xx HTTP errors
http_redirections_exceeded	Too many HTTP redirections
nzb_no_group	Cannot find the requested group on server
nzb_not_found	Article not fount on the server
nzb_invalid_crc	Invalid article CRC
nzb_invalid_size	Invalid article size
nzb_invalid_filename	Invalid filename
nzb_open_failed	Error opening
nzb_write_failed	Error writing
nzb_missing_size	Missing article size
nzb_decode_error	Article decoding error
nzb_missing_segments	Missing article segments
nzb_error	Other nzb error
unknown_host	Unknown host
timeout	Timeout
bad_authentication	Invalid credentials
connection_refused	Remote host refused connection
nzb_authentication_required	Nzb server need authentication
bt_tracker_error	Unable to announce on tracker
bt_missing_files	Missing torrent files
bt_file_error	Error accessing torrent files
missing_ctx_file	Error accessing task context file

Download object

Download objects have the following attributes:

`Download`

`id` `int` ***Read-only***
id

`type` `enum` ***Read-only***
The valid download types are:

Type	Description
bt	bittorrent download
nzb	newsgroup download
http	HTTP download
ftp	FTP download

`name` `string` ***Read-only***

`status` `enum`
The valid download status are:

Status	Description
stopped	task is stopped, can be resumed by setting the status to downloading
queued	task will start when a new download slot is available the queue position is stored in queue_pos attribute
starting	task is preparing to start download
downloading	
stopping	task is gracefully stopping
error	there was a problem with the download, you can get an error code in the error field
done	the download is over. For bt you can resume seeding setting the status to seeding if the ratio is not reached yet
checking	(only valid for nzb) download is over, the downloaded files are being checked using par2
repairing	(only valid for nzb) download is over, the downloaded files are being repaired using par2
extracting	(only valid for nzb) download is over, the downloaded files are being extracted
seeding	(only valid for bt) download is over, the content is Change to being shared to other users. The task will automatically stop once the seed ratio has been reached
retry	You can set a task status to ‘retry’ to restart the download task.

`size` `int` ***Read-only***
download size (in Bytes)

`queue_pos` `int`
position in download queue (0 if not queued)

`io_priority` `enum`
The valid download priorities are:

Priority	Description
low	low

Priority	Description
normal	normal
high	high

`tx_bytes` `int` ***Read-only***
transmitted bytes (including protocol overhead)

`rx_bytes` `int` ***Read-only***
received bytes (including protocol overhead)

`tx_rate` `int` ***Read-only***
current transmit rate (in byte/s)

`rx_rate` `int` ***Read-only***
current receive rate (in byte/s)

`tx_pct` `int` ***Read-only***
transmit percentage (without protocol overhead)

To improve precision the value as been scaled by 100 so that a tx_pct of 123 means 1.23%

`rx_pct` `int` ***Read-only***
received percentage (without protocol overhead)

To improve precision the value as been scaled by 100 so that a tx_pct of 123 means 1.23%

`error` `enum` ***Read-only***
An error code

`created_ts` `timestamp` ***Read-only***
timestamp of the download creation time

`eta` `int` ***Read-only***
estimated remaining download time (in seconds)

`download_dir` `string` ***Read-only***
directory where the file(s) will be saved (base64 encoded)

`stop_ratio` `int` ***Read-only***
Only relevant for bittorrent tasks. Once the transmit ration has been reached the task will stop seeding.

The ratio is scaled by 100 to improve resolution.

A stop_ratio of 150 means that the task will stop seeding once tx_bytes = 1.5 * rx_bytes.

`archive_password` `string`
(**only relevant for nzb**) password for extracting downloaded archives

`info_hash` `string`
(**only relevant for bt**) torrent info_hash encoded in hexa

`piece_length` `int`
(**only relevant for bt**) torrent piece length in bytes

Download API

Retrieve a Download task

`GET /api/v8/downloads/`
Returns the collection of all Download tasks

Example request:

```
GET /api/v8/downloads/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
      "success": true,
      "result": {
            "rx_bytes": 147450,
            "tx_bytes": 3460,
            "download_dir": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMv",
            "archive_password": "",
            "eta": 60290,
            "status": "downloading",
            "io_priority": "normal",
            "type": "bt",
            "piece_length": 524288,
            "queue_pos": 2,
            "id": 1273,
            "info_hash": "A7055D06E5A8F7F816EC01AC7F7F5243D3CB008F",
            "created_ts": 1485513882,
            "stop_ratio": 150,
            "tx_rate": 202,
            "name": "debian-8.7.1-amd64-CD-1.iso",
            "tx_pct": 0,
            "rx_pct": 0,
            "rx_rate": 10950,
            "error": "none",
            "size": 660600000
      }
}
```


`GET /api/v8/downloads/{id}`
Returns the Download task with the given id

Example request:

```
GET /api/v8/downloads/16 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "rx_bytes": 688005364,
        "tx_bytes": 3232055279,
        "download_dir": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMv", /* /Disque dur/Téléchargements/ */
        "archive_password": "",
        "eta": 331896,
        "status": "seeding",
```

```
        "io_priority": "high",
        "size": 678428672,
        "type": "bt",
        "error": "none",
        "queue_pos": 0,
        "id": 14,
        "created_ts": 1349786169,
        "tx_rate": 0,
        "name": "debian-6.0.6-amd64-CD-1.iso",
        "rx_pct": 10000,
        "rx_rate": 0,
        "tx_pct": 0
    }
}
```

Delete a Download task

DELETE /api/v8/downloads/{id}

Deletes the Download task with the given id, **without** erasing the downloaded files If the task was not done it is stopped

You can call this method to remove done tasks from the task list.

Example request:

```
DELETE /api/v8/downloads/16 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

DELETE /api/v8/downloads/{id}/erase

Same as previous, but **erases** the downloaded files

Update a Download task

PUT /api/v8/downloads/{id}

Updates the Download task with the given id

Example request:

```
PUT /api/v8/downloads/16 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "io_priority": "high",
   "status": "stopped"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "rx_bytes": 683407058,
        "tx_bytes": 17866436,
        "download_dir": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMv", /* /Disque dur/Téléchargements/ */
        "eta": 1075260392,
        "status": "stopping",
        "io_priority": "high",
        "size": 678428672,
        "type": "bt",
        "error": "none",
        "queue_pos": 0,
        "id": 14,
        "created_ts": 1349786169,
        "tx_rate": 0,
        "name": "debian-6.0.6-amd64-CD-1.iso",
        "stop_ratio": 55936,
        "rx_pct": 10000,
        "rx_rate": 0,
        "tx_pct": 4
    }
}
```

Get download log

GET /api/v8/downloads/{id}/log

Get the log.

Example request:

```
GET /api/v8/downloads/16/log HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": "log line\nanother log line\n"
}
```

Adding a new Download task

Adding by URL
Supported URL scheme are `http://`, `ftp://`, `magnet:`

You can start a recursive download by setting the recursive parameter. The downloader will then extract links from each donwloaded html page and continue downloading files on the same domain and on the same root path. This can be used to download all the files on a directory index.

You can add multiple downloads at once by passing a list of URL (separated by a new line delimiter) in download_url_list instead of using download_url.

/!NOTE: for this API the request arguments must be encoded using “application/x-www-form-urlencoded” (or “multipart/form-data” for file upload) instead of “application/json”

POST /api/v8/downloads/add

Parameters

- **download_url** (*string*) – The URL
- **download_url_list** (*string*) – A list of URL separated by a new line delimiter (use download_url or download_url_list)
- **download_dir** (*string*) – The download destination directory (optional: will use the configuration download_dir by default)

- **filename** (*string*) – Override the name of the destination file. Only valid with one, non-recursive download_url.
- **hash** (*string*) – Verify the hash of the downloaded file. The format is sha256:xxxxxx or sha512:xxxxxx; or the URL of a SHA256SUMS, SHA512SUMS, -CHECKSUM or .sha256 file. Only valid with one, non-recursive download_url.
- **recursive** (*bool*) – If true the download will be recursive
- **username** (*string*) – Auth username (optional)
- **password** (*string*) – Auth password (optional)
- **archive_password** (*string*) – The password required to extract downloaded content (only relevant for nzb)
- **cookies** (*string*) – The http cookies (to be able to pass session cookies along with url). This is the content of the HTTP Cookie header, for example: cookie1=value1; cookie2=value2

NOTE: instead of passing password and username you can include them in the URL.

Example request : Single download add:

```
POST /api/v8/downloads/add HTTP/1.1
Host: mafreebox.freebox.fr

download_url=http%3A%2F%2Fcdimage.debian.org%2Fdebian-cd%2F6.0.6%2Famd64%2Fbt-cd%2Fdebian-6.0.6-amd64-CD-1.iso.torrent
&download_dir=L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMv
```

Example response:

On success you'll get the id of the new download task.

```
{
   "result": {
      "id": 23
    },
   "success": true
}
```

Example request : Multiple downloads at once:

```
POST /api/v8/downloads/add HTTP/1.1
Host: mafreebox.freebox.fr

download_url_list=ftp%3A%2F%2Ftest-debit.free.fr%2F1024.rnd
        %0Ahttp%3A%2F%2Ftest-debit.free.fr%2F4096.rnd
        %0Ahttp%3A%2F%2Ftest-debit.free.fr%2F32768.rnd
&download_dir=L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMv
```

Example response:

On success you'll get the list of id of the new download tasks.

```
{
   "result": {
       "id": [
           32,
           33,
           34
       ]
    },
    "success": true
}
```

Adding by file upload
Supported files are .torrent, .nzb,

`POST /api/v8/downloads/add`

Parameters

- **download_file** (*string*) – The download file (must be uploaded using multipart/form-data)
- **download_dir** (*string*) – The download destination directory (optional: will use the configuration download_dir by default)
- **archive_password** (*string*) – The password required to extract downloaded content (only relevant for nzb)

Example request:

```
POST /api/v8/downloads/add HTTP/1.1
Host: mafreebox.freebox.fr
Content-Type: multipart/form-data; boundary=---------------------------176791920111939857911845395343
Content-Length: 26651

---------------------------176791920111939857911845395343
Content-Disposition: form-data; name="download_dir"

L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMv
---------------------------176791920111939857911845395343
Content-Disposition: form-data; name="archive_password"


---------------------------176791920111939857911845395343
Content-Disposition: form-data; name="download_file"; filename="debian-6.0.6-amd64-CD-1.iso.torrent"
Content-Type: application/x-bittorrent

d8:announce41:http://bttracker.debian.org:6969/announce7:comment [ ... ]
```

Example response:

```
{
   "result": {
      "id": 42
   },
   "success": true
}
```

Download Stats

If you just want to display synthetic information about downloader this is the method to use.

Download Nzb configuration status Object

`NzbConfigStatus`

`status` `enum` ***Read-only***

The valid config status are:

Type	Description
not_checked	config has not been checked yet
checking	test in progress
error	config is invalid, see error
ok	config is ok

`error` `enum` ***Read-only***

The valid config status are:

Type	Description
none	test is ok
nzb_authentication_required	authentication is required
bad_authentication	incorrect credentials
connection_refused	unable to connect to NNTP server

Download DHT stats Object

`DhtStats`

- `enabled` bool ***Read-only***
  is the dht enabled
- `node_count` int ***Read-only***
  number of active nodes
- `enabled_ipv6` bool ***Read-only***
  is the dht enabled on IPv6
- `node_count_ipv6` int ***Read-only***
  number of active nodes on IPv6

Download Stats Object

`DownloadStats`

- `nb_tasks` int ***Read-only***
  total number of tasks
- `nb_tasks_stopped` int ***Read-only***
  number of stopped tasks
- `nb_tasks_checking` int ***Read-only***
  number of checking tasks
- `nb_tasks_queued` int ***Read-only***
  number of queued tasks
- `nb_tasks_extracting` int ***Read-only***
  number of extracting tasks
- `nb_tasks_done` int ***Read-only***
  number of done tasks
- `nb_tasks_repairing` int ***Read-only***
  number of repairing tasks
- `nb_tasks_seeding` int ***Read-only***
  number of seeding tasks
- `nb_tasks_downloading` int ***Read-only***
  number of downloading tasks
- `nb_tasks_error` int ***Read-only***
  number of error tasks
- `nb_tasks_stopping` int ***Read-only***
  number of stopping tasks
- `nb_tasks_active` int ***Read-only***
  number of active tasks (checking + queued + extracting + repairing + seeding + downloading)
- `nb_rss` int ***Read-only***
  number of RSS feed subscriptions
- `nb_rss_items_unread` int ***Read-only***
  number of unread RSS items
- `rx_rate` int ***Read-only***
  current receive rate in bytes / second
- `tx_rate` int ***Read-only***
  current transmit rate in bytes / second
- `throttling_mode` enum ***Read-only***
  active throttling_mode (see `DlThrottlingConfig`)
- `throttling_is_scheduled` bool ***Read-only***
  if true, the current throttling mode has been computed using the throttling schedule

  if false, the current throttling mode has been manually forced
- `throttling_rate` :json:object:`DlRate` ***Read-only***
  current rate for throttling
- `nzb_config_status` :json:object:`NzbConfigStatus` ***Read-only***
  current nzb configuration status
- `conn_ready` bool ***Read-only***
  is the connection ready
- `nb_peer` int ***Read-only***
  number of bittorrent peers
- `blocklist_entries` int ***Read-only***
  number of rules in blocklist
- `blocklist_hits` int ***Read-only***
  number of hits in blocklist
- `dht_stats` :json:object:`DhtStats` ***Read-only***
  dht stats

Get the Download Stats

`GET /api/v8/downloads/stats`

Example request:

```
GET /api/v8/downloads/stats HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
{
    "success": true,
    "result": {
        "throttling_rate": {
            "rx_rate": 0,
            "tx_rate": 0
```

```
        },
        "nb_tasks_stopped": 1,
        "nb_tasks_checking": 0,
        "nb_tasks_queued": 0,
        "nb_tasks_extracting": 4,
        "nb_tasks_done": 1,
        "nb_tasks_repairing": 0,
        "throttling_mode": "normal",
        "nb_tasks_active": 11,
        "tx_rate": 4294,
        "nb_tasks_downloading": 4,
        "throttling_is_scheduled": true,
        "nb_tasks": 13,
        "nb_tasks_error": 0,
        "nb_tasks_stopping": 0,
        "nb_rss_items_unread": 5,
        "rx_rate": 14222,
        "nb_tasks_seeding": 3
    }
}
```


Download Files

Download Files Object

Each Download has one or more DownloadFile.

DownloadFile

id string ***Read-only***
opaque id

task_id int ***Read-only***
id of the download task

path string ***Read-only***
[DEPRECATED]

filepath string ***Read-only***
full filepath on the disk (encoded as in file system api)

name string ***Read-only***
file name

mimetype string ***Read-only***
file mimetype

size int ***Read-only***
file size in bytes

rx int ***Read-only***
received bytes

status enum ***Read-only***
file download status

Status	Description
queued	file is queued for download
error	there was a problem with this file, see error to get the error code
done	file download is completed

error enum ***Read-only***
file error code in case status is error

priority string
file download priority inside the download task

Priority	Description
no_dl	this file will not be downloaded
low	low priority
normal	default priority
high	high priority

preview_url string ***Read-only***
url to preview downloaded file (only available for bittorrent) as a share link, this url can be use without requiring any form of authentication so that it can be passed as-is to any software.

Download Files API

Get the list of files for a given Download

GET /api/v8/downloads/{task_id}/files

Example request:

```
GET /api/v8/downloads/37/files HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:


```
{
    "success": true,
    "result": [
      {
          "path": "/Disque dur/Téléchargements//test-debit.free.fr.html",
          "id": "5-1",
          "task_id": "5",
          "filepath": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMvL3Rlc3QtZGViaXQuZnJlZS5mci5odG1s",
          "mimetype": "text/html",
          "name": "test-debit.free.fr.html",
          "rx": 0,
          "status": "done",
          "priority": "normal",
          "error": "none",
          "size": 0

      },
      {
          "path": "/Disque dur/Téléchargements//test-debit.free.fr/1024.rnd",
          "id": "5-7",
          "task_id": "5",
          "filepath": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMvL3Rlc3QtZGViaXQuZnJlZS5mci8xMDI0LnJuZA==",
          "mimetype": "application/octet-stream",
          "name": "1024.rnd",
          "rx": 1048576,
          "status": "done",
```

```
            "priority": "low",
            "error": "none",
            "size": 1048576
        },

          [ ... ]

        {
            "path": "/Disque dur/Téléchargements//test-debit.free.fr/image.iso",
            "id": "5-16",
            "task_id": "5",
            "filepath": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMvL3Rlc3QtZGViaXQuZnJlZS5mci9pbWFnZS5pc28=",
            "mimetype": "application/x-cd-image",
            "name": "image.iso",
            "rx": 678428672,
            "status": "done",
            "priority": "low",
            "error": "none",
            "size": 678428672
        }
    ]

}
```

Change the priority of a Download File

`PUT /api/v8/downloads/{task_id}/files/{file_id}`

Parameters

- **task_id** (*string*) – The download task id
- **path** (*string*) – The file_id
- **priority** (*string*) – The new file download priority

Example request:

```
PUT /api/v8/downloads/37/files/37-4 HTTP/1.1
Host: mafreebox.freebox.fr

{
  "priority" : "high"
}
```

Example response:

```
{
   "success": true
}
```

Download Trackers [UNSTABLE]

Download Tracker Object

Each torrent `Download` task has one or more `DownloadTracker`.

Each tracker is identified by its announce URL.

`DownloadTracker`

`announce` `string` ***Read-only***
tracker announce URL

`is_backup` `bool` ***Read-only***
true if the tracker is a backup tracker (the downloader won't connect to this tracker unless the primary tracker fails)

`status` `enum` ***Read-only***
tracker status

Status	Description
unannounced	not announced
announcing	announcing
announce_failed	an error occurred while trying to announce
announced	announced

`interval` `int` ***Read-only***
desired interval between two announces (in seconds)

`min_interval` `int` ***Read-only***
minimum interval between two announces (in seconds)

`reannounce_in` `int` ***Read-only***
time left before reannounce (in seconds)

`nseeders` `int` ***Read-only***
number of seeders announced on tracker

`nleechers` `int` ***Read-only***
number of leechers announced on tracker

`is_enabled` `bool`
is the tracker enabled

Download Tracker API

Get the list of trackers for a given Download

Attempting to call this method on a download other than bittorent will fail

`GET /api/v8/downloads/{task_id}/trackers`

Example request:

```
GET /api/v8/downloads/35/tracker HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
{
    "success": true,
    "result": [
        {
            "nseeders": 0,
            "nleechers": 0,
            "reannounce_in": 790,
            "is_backup": false,
            "interval": 900,
            "min_interval": 60,
            "announce": "http://bttracker.debian.org:6969/announce",
            "status": "announced"
```

```
        }
    ]
}
```

Add a new tracker

Attempting to call this method on a download other than bittorent will fail

`POST /api/v8/downloads/{task_id}/trackers`

Example request:

```
POST /api/v8/downloads/35/tracker HTTP/1.1
Host: mafreebox.freebox.fr

{
  "announce": "udp://tracker.openbittorrent.com:80"
}
```

Example response:

```
{
    "success": true
}
```

Remove a tracker

`DELETE /api/v8/downloads/{task_id}/trackers/{announce}`

Example request:

```
DELETE /api/v8/downloads/35/tracker/udp%3A%2F%2Ftracker.openbittorrent.com%3A80 HTTP/1.1
Host: mafreebox.freebox.fr

{
  "announce": "udp://tracker.openbittorrent.com:80"
}
```

Example response:

```
{
    "success": true
}
```

Update a tracker

`PUT /api/v8/downloads/{task_id}/trackers/{announce}`

Example request:

```
PUT /api/v8/downloads/35/tracker/udp%3A%2F%2Ftracker.openbittorrent.com%3A80 HTTP/1.1
Host: mafreebox.freebox.fr

{
  "announce": "udp://tracker.openbittorrent.com:80",
  "is_enabled": true
}
```

Example response:

```
{
    "success": true
}
```

Download Peers [UNSTABLE]

Download Peer Object

Each torrent `Download` task has one or more `DownloadPeer`.

`DownloadPeer`

`host` `string` ***Read-only***

peer IP

`port` `int` ***Read-only***

peer port

`state` `enum` ***Read-only***

peer state

State	Description
disconnected	not connected
connecting	trying to connect to the peer
handshaking	connected to the peer, negotiating capabilities
ready	ready to exchange data

`origin` `enum` ***Read-only***

peer origin

Origin	Description
tracker	got the peer from the tracker
incoming	incoming peer
dht	got the peer from DHT
pex	got the peer from Peer exchange protocol
user	manually added peer

`protocol` `enum` ***Read-only***

Protocol	Description
tcp	TCP
tcp_obfuscated	Obfuscated TCP
udp	UDP

`client` `string` ***Read-only***

Bittorrent client name

`country_code` `string` ***Read-only***

Peer country code (iso 3166)

If country code is not available it will have the value “??”

`tx` `int` ***Read-only***

transmitted bytes

`rx` `int` ***Read-only***

received bytes

`tx_rate` `int` ***Read-only***

current transmit rate in byte/s

`rx_rate` `int` ***Read-only***

current receive rate in byte/s

`progress` `int` ***Read-only***

peer current download progress

`requests[]` `array of int` ***Read-only***

current requested pieces

Get the list of peers for a given Download

Attempting to call this method on a download other than bittorent will fail

`GET /api/v8/downloads/{task_id}/peers`

Example request:

```
GET /api/v8/downloads/42/peers HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
{
    "success": true,
    "result": [
        {
            "protocol": "tcp_obfuscated",
            "origin": "tracker",
            "progress": 91,
            "remote_choke": true,
            "requests": { },
            "host": "186.213.200.201",
            "port": 0,
            "client": "Azureus 4.7.2.0",
            "country_code": "BR",
            "local_interest": false,
            "state": "ready",
            "rx": 1617,
            "tx": 836670,
            "remote_interest": true,
            "tx_rate": 0,
            "rx_rate": 0,
            "local_choke": false
        },

        [ ... ]

        {
          "protocol": "tcp",
          "origin": "tracker",
          "progress": 11,
          "remote_choke": true,
          "requests": { },
          "host": "208.127.4.60",
          "port": 0,
          "client": "Transmission 2.51",
          "country_code": "US",
          "local_interest": false,
          "state": "ready",
          "rx": 8929,
          "tx": 7592234,
          "remote_interest": true,
          "tx_rate": 0,
          "rx_rate": 0,
          "local_choke": false
        }
    ]
}
```

Download Pieces

Each Torrent is split in 'pieces' of fixed size. The Download Piece Api allow tracking the download state of each pieces of a Torrent

Get the pieces status a given download

The result value is a string, with each character representing a piece status. Piece status can be:

Status	Description
X	piece is complete
•	piece is currently downloading
.	piece is wanted but not downloading yet
•	piece is not wanted and will not be downloaded
/	piece is downloading with high priority as it is needed for file preview
U	piece is scheduled with high priority as it is needed for file preview

`GET /api/v8/downloads/{task_id}/pieces`

Example request:

```
GET /api/v8/downloads/5/pieces HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
{
    "success": true,
    "result": "XXXXX//++....-- [ ... ] XXX"
}
```

Download Blacklist [UNSTABLE]

For bittorrent downloads, we use a blacklist to store information about "useless" or broken peers. For instance if a peer is complete and we are trying to seed data, there is no use attempting to connect to this peer again.

The download blacklist api allow you to retrieve information about this blacklist, and remove, or add peers to the blacklist.

Each `DownloadBlacklistEntry` can be specific to a torrent, or "global" and apply to any torrent.

Download Blacklist Object

`DownloadBlacklistEntry`

`host` `string` ***Read-only***

entry ip

`reason` `enum` ***Read-only***

blacklist reason

State	Description
not_blacklisted	
crypto_not_supported	peer does not support encrypted connection
connect_fail	failed to connect
hs_timeout	handshake timeout
hs_failed	handshake failed
hs_crypt_failed	handshake failed during crypto
hs_crypto_disabled	handshake failed because encryption is disabled
torrent_not_found	torrent not found
read_failed	failed to read from peer
write_failed	failed to send data to peer
crap_received	received invalid data from peer
conn_closed	connection closed by remote peer
timeout	timeout
blocklist	peer is in a blocked ip range
user	manually blacklisted

`expire` `int` ***Read-only***

time left before blacklist removal

`global` `bool` ***Read-only***

does this entry applies to all torrents

Get the list of blacklist entries for a given download

Attempting to call this method on a download other than bittorent will fail.

`GET /api/v8/downloads/{task_id}/blacklist`

Example request:

```
GET /api/v8/downloads/5/blacklist HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:


```
{
    "success": true,
    "result": [
        {
            "host": "89.215.188.6",
            "expire": 90,
            "global": true,
            "reason": "torrent_not_found"
        },
        {
            "host": "94.23.0.89",
            "expire": 120,
            "global": true,
            "reason": "conn_closed"
        },
        {
            "host": "188.254.151.215",
            "expire": 150,
            "global": true,
            "reason": "timeout"
        },
        {
            "host": "201.25.54.26",
            "expire": 180,
            "global": true,
            "reason": "timeout"
        }
    ]
}
```


Empty the blacklist for a given download

This call allow to remove all global entries, and entries related to the given download

`DELETE /api/v8/downloads/{task_id}/blacklist/empty`

Example request:

```
DELETE /api/v8/downloads/5/blacklist/empty HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:


```
{
   "success": true
}
```


Delete a particular blacklist entry

`DELETE /api/v8/downloads/blacklist/{host}`

Example request:

```
DELETE /api/v8/downloads/blacklist/201.25.54.26 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:


```
{
   "success": true
}
```


Add a blacklist entry

`POST /api/v8/downloads/blacklist`

Example request:

```
POST /api/v8/downloads/blacklist HTTP/1.1
Host: mafreebox.freebox.fr

{
  "host": "8.8.8.8",
  "expire": 3600
}
```

Example response:

```
{
    "success": true,
    "result":
       {
          "host": "197.200.139.87",
          "expire": 300,
          "global": true,
          "reason": "user"
       }
}
```

Download Feeds

The Freebox downloader supports subscribing to RSS feeds, for automatic content download.

Download Feed object

Download Feeds have the following attributes:

`DownloadFeed`

`id` `int` ***Read-only***
id

`status` `enum` ***Read-only***
The feed can have the following status

Status	Description
ready	feed is up to date
fetching	feed is updating
error	there was an error trying to refresh this feed, see error

`url` `string` ***Read-only***
Feed URL

`title` `string` ***Read-only***
Feed title (extracted from the RSS)

`desc` `string` ***Read-only***
Feed description (extracted from the RSS)

`image_url` `string` ***Read-only***
Feed image URL (extracted from the RSS)

`nb_read` `int` ***Read-only***
Number of read items in the feed

`nb_unread` `int` ***Read-only***
Number of unread items in the feed

`auto_download` `bool`
If set to true, the downloader will automatically download new items

`fetch_ts` `timestamp` ***Read-only***
Last time the feed was fetched

`pub_ts` `timestamp` ***Read-only***
Last time the feed was published on remote server

`error` `enum` ***Read-only***
Error code (same as used in `Download` or `DownloadFile`).

Download Feed Errors

When attempting to access the download feed API, you may encounter the following errors:

error_code	Description
feed_not_found	No feed was found with the given id
item_not_found	No feed item was found with the given id
feed_is_recent	You are trying to update a feed that is already up to date
internal_error	Internal error

Download Feed API

Get the list of all download Feeds

`GET /api/v8/downloads/feeds/`

Returns the collection of all `DownloadFeed` feeds

Example request:

```
GET /api/v8/downloads/feeds/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": [
      {
          "auto_download": false,
          "id": 1,
          "desc": "Custom RSS feed based off search filters.",
          "error": "none",
          "nb_read": 0,
          "title": "ezRSS - Search Results",
          "image_url": "http://ezrss.it/images/ezrssit.png",
          "status": "ready",
          "url": "http://www.ezrss.it/search/index.php?show_name=Ubuntu&mode=rss",
          "nb_unread": 29,
          "fetch_ts": 1349885023,
          "pub_ts": 1350583600
```

```
        },
        {
            "auto_download": false,
            "id": 2,
            "desc": "Latest nzb for Debian",
            "error": "none",
            "nb_read": 0,
            "title": "Debian NZB RSS",
            "image_url": "",
            "status": "ready",
            "url": "http://www.nzb-rss.com/rss/Debian.rss",
            "nb_unread": 13,
            "fetch_ts": 1350469391,
            "pub_ts": 1350583600
        }
    ]
}
```


Get a download Feed

GET /api/v8/downloads/feeds/{id}

Gets the DownloadFeed with the given id

Example request:

```
GET /api/v8/downloads/feeds/2 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true,
  "result":
      {
          "auto_download": false,
          "id": 2,
          "desc": "Latest nzb for Debian",
          "error": "none",
          "nb_read": 0,
          "title": "Debian NZB RSS",
          "image_url": "",
          "status": "ready",
          "url": "http://www.nzb-rss.com/rss/Debian.rss",
          "nb_unread": 13,
          "fetch_ts": 1350469391,
          "pub_ts": 1350583600
      }
}
```


Add a Download Feed

POST /api/v8/downloads/feeds/

Creates a new DownloadFeed.

Example request:

```
POST /api/v8/downloads/feeds/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "url": "http://www.nzb-rss.com/rss/Debian-unstable.rss"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "auto_download": false,
        "error": "none",
        "desc": "",
        "status": "ready",
        "nb_read": 0,
        "title": "",
        "image_url": "",
        "feed_id": 6,
        "url": "http://www.nzb-rss.com/rss/Debian-unstable.rss",
        "nb_unread": 0,
        "fetch_ts": 0,
        "pub_ts": 1350583600
    }

}
```


Delete Download Feed

DELETE /api/v8/downloads/feeds/{id}

Deletes the DownloadFeed and all the associated items.

This will not alter the Download tasks.

Example request:

```
DELETE /api/v8/downloads/feeds/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Update a Download Feed

PUT /api/v8/downloads/feeds/{id}

Updates the DownloadFeed task with the given id

Example request:

```
PUT /api/v8/downloads/feeds/2 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
    "auto_download": true
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "auto_download": true,
        "error": "none",
        "desc": "Latest nzb for Debian",
        "title": "Debian NZB RSS",
        "status": "ready",
        "nb_read": 0,
        "image_url": "",
        "feed_id": 2,
        "url": "http://www.nzb-rss.com/rss/Debian.rss",
        "nb_unread": 13,
        "fetch_ts": 1350583674,
        "pub_ts": 1350583600
    }
}
```

Refresh a Download Feed

`POST /api/v8/downloads/feeds/{id}/fetch`

Remotely fetches the RSS feed and updates it.

Note that if the remote feed specifies a TTL, trying to update before the ttl will result in feed_is_recent error

Example request:

```
POST /api/v8/downloads/feeds/2/fetch HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Refresh all Download Feeds

`POST /api/v8/downloads/feeds/fetch`

Remotely fetches all the RSS feeds.

Example request:

```
POST /api/v8/downloads/feeds/fetch HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Download Feed Item object

Each RSS DownloadFeed contains feed items object

`DownloadFeedItem`

- `id` int ***Read-only***
  id
- `feed_id` int ***Read-only***
  id of the DownloadFeed
- `title` string[ro]
  item title
- `desc` string[ro]
  item description
- `author` string ***Read-only***
  item author
- `link` string ***Read-only***
  URL of the RSS feed attachment
- `is_read` bool
  you can mark the item as read manually, or it is marked as read automatically when the item is downloaded
- `is_downloaded` bool ***Read-only***
  mark downloaded items, automatically set to true when RSS item is downloaded
- `fetch_ts` timestamp ***Read-only***
  timestamp of the item creation
- `pub_ts` timestamp ***Read-only***
  item publish timestamp
- `enclosure_url` string ***Read-only***
  enclosure URL (if specified in RSS feed)
- `enclosure_type` string ***Read-only***
  enclosure mime type (if specified in RSS feed)
- `enclosure_length` int ***Read-only***
  enclosure size in bytes (if specified in RSS feed)

Get the items of a given RSS feed

`GET /api/v8/downloads/feeds/{feed_id}/items/`

Returns the collection of all `DownloadFeedItems` for a given DownloadFeed

Example request:

```
GET /api/v8/downloads/feeds/2/items/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true,
  "result": [
    {
      "pub_ts": 1350657300,
      "fetch_ts": 1350657317,
      "is_read": true,
      "title": "debian-6.0.4-amd64-CD-1.iso",
      "link": "http://bttracker.debian.org:6969/file/debian-6.0.4-amd64-CD-1.iso.torrent?info_hash=95ce23e889cc26901740f87ac25270da725bfd36",
      "id": 2845,
      "author": "debian",
      "feed_id": 2,
      "desc": ""
    },
    {
      "pub_ts": 1350657300,
      "fetch_ts": 1350657318,
      "is_read": false,
      "title": "debian-6.0.4-amd64-CD-2.iso",
      "link": "http://bttracker.debian.org:6969/file/debian-6.0.4-amd64-CD-2.iso.torrent?info_hash=34583a8e25ef1528a8bfce99d24f401acb24d982",
      "id": 2846,
      "author": "debian",
      "feed_id": 2,
      "desc": ""
    }
  ]
}
```


Update a feed item

`PUT /api/v8/downloads/feeds/{feed_id}/items/{item_id}`

Returns the collection of all `DownloadFeedItems` for a given DownloadFeed

Example request:

```
PUT /api/v8/downloads/feeds/2/items/2846 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
    "is_read": true
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true
}
```

Download a feed item

`POST /api/v8/downloads/feeds/{feed_id}/items/{item_id}/download`

This method will enqueue the RSS item to the download list

Example request:

```
POST /api/v8/downloads/feeds/2/items/2846/download HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true
}
```

Mark all items as read

`POST /api/v8/downloads/feeds/{feed_id}/items/mark_all_as_read`

This method will mark each items as read

Example request:

```
POST /api/v8/downloads/feeds/2/items/mark_all_as_read HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true
}
```

Download Configuration

Download configuration object

The download configuration is a singleton used to store the downloader preferences.

Global config

`DownloadConfiguration`

- `max_downloading_tasks` **int**
  max concurrent download tasks
- `download_dir` **string**
  the default path where downloads will be stored (base64 encoded)
- `watch_dir` **string**
  special folder that will be monitored. When a new supported file (.nzb, .torrent) is copied in that folder, the task is automatically added to the download queue.
  (base64 encoded)
- `use_watch_dir` **bool**
  if set to false, the watch_dir will not be monitored
- `throttling` **DlThrottlingConfig**
  throttling configuration
- `news` **DlNewsConfig**
  newsgroups configuration

`bt` **DlBtConfig**
bittorrent configuration

`feed` **DlFeedConfig**
RSS feed configuration

`blocklist` **DlBlockListConfig**
block list configuration

`dns1` **string**
dns server ip to use for downloader (leave blank for default dns server)

`dns2` **string**
dns server ip to use for downloader

Throttling config

`DlThrottlingConfig`

`normal` **DlRate**
download rate for normal time slot (in B/s)

`slow` **DlRate**
download rate for normal slow slot (in B/s)

`schedule` **enum[168]**
The schedule array represent the list of week hours timeslot, starting on monday a midnight. Therefore the complete week is represented in a array of 168 elements (24 * 7)

Each slot can have the following value:

Type	Description
normal	downloads will use normal DlRate config for this timeslot
slow	downloads will use slow DlRate config for this timeslot
hibernate	downloads will be paused for this timeslot

`mode` **enum**
Throttling mode can have to following values

Type	Description
normal	force use of normal rate limits (not using the scheduler)
slow	force use of slow rate limits (not using the scheduler)
hibernate	force hibernate (not using the scheduler)
schedule	use scheduded rate limit

`DlRate`

`tx_rate` **int**
maximum transmit rate (in byte/s) 0 means no limit

`rx_rate` **int**
maximum receive rate (in byte/s) 0 means no limit

Newsgroups config

`DlNewsConfig`

`server` **string**
NNTP server hostname

`port` **int**
NNTP server port

`ssl` **bool**
Use SSL to connect to server if set to true

`user` **string**
NNTP auth username (can be empty if no auth is required)

`password` **string** ***Write-only***
NNTP auth password (can be empty if no auth is required)

`nthreads` **int**
maximum concurrent connections to the NNTP server

`auto_repair` **bool**
automatically check and repair downloaded files using the provided par2 files

`lazy_par2` **bool**
if set to true the downloader will download the par2 files only if the download is corrupted

`auto_extract` **bool**
automatically attempt to extract downloaded files

`erase_tmp` **bool**
if auto_extract is enabled, delete archive files once successfully extracted

Bittorrent config

`DlBtConfig`

`max_peers` **int**
maximum number of peers at a given time

`stop_ratio` **int**
default stop_ratio for bt `Download` tasks

This value is scaled by a factor 100, for instance a stop_ratio of 200 means that the task will stop once tx_bytes = 2 * size

A value of 0 means that the task will continue seeding until it is manually stopped

`crypto_support` **enum**
The crypto_support can have the following values

Type	Description
unsupported	will never use bittorrent crypto
allowed	will select plain during handshake
preferred	will select crypto during handshake
required	will allow plain bittorrent

`enable_dht` **bool**
enable the dht protocol

`enable_pex` **bool**
enable the peer exchange protocol

`announce_timeout` **int**

timeout in seconds for announcing to tracker

`main_port` **int**
main bittorrent port

`dht_port` **int**
bittorrent dht port

Rss Feeds config

`DlFeedConfig`

`fetch_interval` **int**
interval between automatic RSS refresh (in minutes)

`max_items` **int**
maximum feed item to keep

BlockList config

`DlBlockListConfig`

`sources[]` **string**
list of block list URL source

The block list should be in cidr format

e.g.: http://list.iblocklist.com/?list=bt_level1&fileformat=cidr&archiveformat=

Get the current Download configuration

`GET /api/v8/downloads/config/`

Returns the current `DownloadConfiguration`

Example request:

```
GET /api/v8/downloads/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
       "feed": {
           "max_items": 0,
           "fetch_interval": 60
       },
       "use_watch_dir": true,
       "watch_dir": "L0Rpc3F1ZSBkdXIvLnF1ZXVl", /* /Disque dur/.queue */
       "news": {
           "user": "",
           "erase_tmp": true,
           "port": 119,
           "nthreads": 1,
           "auto_repair": true,
           "ssl": false,
           "auto_extract": true,
           "lazy_par2": true,
           "server": "news.free.fr"
       },
       "bt": {
           "max_peers": 50,
           "stop_ratio": 150,
           "crypto_support": "allowed"
       },
       "max_downloading_tasks": 5,
       "download_dir": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMv", /* /Disque dur/Téléchargements/ */
       "throttling": {
           "normal": {
               "rx_rate": 0,
               "tx_rate": 0
           },
           "slow": {
               "rx_rate": 512,
               "tx_rate": 42
           },
           "schedule": [
               "slow",
               "normal",
               "normal",

                [ ... ]

               "normal",
               "normal",
               "normal",
               "slow"
           ],
           "mode": "normal"
       }
   }
}
```


Update the Download configuration

`PUT /api/v8/downloads/config/`

Updates the `DownloadConfiguration`

Example request:

```
PUT /api/v8/downloads/config/ HTTP/1.1
Host: mafreebox.freebox.fr

{
    "throttling": {
        "normal": {
            "rx_rate": 512,
            "tx_rate": 40
        },
        "slow": {
            "rx_rate": 128,
            "tx_rate": 10
        },
        "mode": "normal",
        "schedule": [
            "slow",
```

```
            "normal",
            "normal",

            [ ... ]

            "normal",
            "normal",
            "normal",
            "normal",
            "slow"
        ]
    },
    "max_downloading_tasks": 5,
    "download_dir": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMv", /* /Disque dur/Téléchargements/ */
    "use_watch_dir": true,
    "watch_dir": "L0Rpc3F1ZSBkdXIvLnF1ZXVl", /* /Disque dur/.queue */
    "news": {
        "server": "news.free.fr",
        "port": "119",
        "ssl": false,
        "nthreads": 1,
        "user": "",
        "lazy_par2": true,
        "auto_repair": true,
        "auto_extract": true,
        "erase_tmp": true
    },
    "bt": {
        "max_peers": 50,
        "stop_ratio": 150,
        "crypto_support": "allowed"
    },
    "feed": {
        "fetch_interval": 60
    }
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "feed": {
            "max_items": 0,
            "fetch_interval": 60
        },
        "use_watch_dir": true,
        "watch_dir": "L0Rpc3F1ZSBkdXIvLnF1ZXVl", /* /Disque dur/.queue */
        "news": {
            "user": "",
            "erase_tmp": true,
            "port": 119,
            "nthreads": 1,
            "auto_repair": true,
            "ssl": false,
            "auto_extract": true,
            "lazy_par2": true,
            "server": "news.free.fr"
        },
        "bt": {
            "max_peers": 50,
            "stop_ratio": 150,
            "crypto_support": "allowed"
        },
        "max_downloading_tasks": 5,
        "download_dir": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHMv", /* /Disque dur/Téléchargements/ */
        "throttling": {
            "normal": {
                "rx_rate": 512,
                "tx_rate": 40
            },
            "slow": {
                "rx_rate": 128,
                "tx_rate": 10
            },
            "schedule": [
                "slow",
                "normal",
                "normal",
                "normal",

                [ ... ]

                "normal",
                "normal",
                "normal",
                "slow"
            ],
            "mode": "normal"
        }
    }

}
```

Updating the current Throttling mode

PUT /api/v8/downloads/throttling

You can force the throttling mode using this method. You can use any of the throttling modes defined in DlThrottlingConfig. Setting to schedule will automatically set correct throttling mode. Other values will force the throttling mode until you set it back to schedule.

Example request:

```
PUT /api/v8/downloads/throttling HTTP/1.1
Host: mafreebox.freebox.fr

{
    throttling: "slow"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "is_scheduled": false,
        "throttling": "slow"
    }
}
```


File System Api

File System

With the file system API you can access files on Freebox internal disk and disks connected to the Freebox.

Path encoding

NOTE:

For maximum compatibility issues path are encoded in base64, you *should* use the path as it is returned by the ls API call.

For instance this will solve problems with unicode equivalence .

Although “Spécial” (0x53 0x70 **0xc3 0xa9** 0x63 0x69 0x61 0x6c) and “Spécial” (0x53 0x70 **0x65 0xcc 0x81** 0x63 0x69 0x61 0x6c) are utf8 equivalent, it represents two different paths.

Some software/libraries will replace the original string with its normalized form, causing issues. The use of base64 encoded path will ensure the original path will be preserved.

File System Errors

When attempting to access the file system API, you may encounter the following errors:

error_code	Description
invalid_id	Invalid object id
path_not_found	File or folder not found
internal_error	Internal error
disk_unavailable	The disk is not mounted
invalid_request	Invalid request
invalid_conflict_mode	The conflict mode specified is invalid (see below)
exec_failed	Internal error
out_of_memory	Out of memory
task_not_found	Invalid task id
invalid_state	You tried to set an invalid state
invalid_task_type	This operation cannot be performed on this task
destination_conflict	The destination file/folder already exists
access_denied	Access to this file is denied
disk_full	The destination disk is full

Task

File system tasks have the following attributes:

FsTask

id int ***Read-only***

id

type enum ***Read-only***

The valid task types are:

Type	Description
cat	Concatenate multiple files
cp	Copy files
mv	Move files
rm	Remove files
archive	Creates an archive
extract	Extract an archive
repair	Check and repair files

state enum

State	Description
queued	Queued (only one task is active at a given time)
running	Running
paused	Paused (user suspended)
done	Done
failed	Failed (see error)

error enum ***Read-only***

Error	Description
none	No error
archive_read_failed	Error reading archive
archive_open_failed	Error opening archive
archive_write_failed	Error writing archive
chdir_failed	Error changing directory
dest_is_not_dir	The destination is not a directory
file_exists	File already exists
file_not_found	File not found
mkdir_failed	Unable to create directory
open_input_failed	Error opening input file
open_output_failed	Error opening output file
opendir_failed	Error opening directory
overwrite_failed	Error overwriting file

Error	Description
path_too_big	Path is too long
repair_failed	Failed to repair corrupted files
rmdir_failed	Error removing directory
same_file	Source and Destination are the same file
unlink_failed	Error removing file
unsupported_file_type	This file type is not supported
write_failed	Error writing file
disk_full	Disk is full
internal	Internal error
invalid_format	Invalid file format (corrupted ?)
incorrect_password	Invalid or missing password for extraction
permission_denied	Permission denied
readlink_failed	Failed to read the target of a symbolic link
symlink_failed	Failed to create a symbolic link
copy_into_itself	Attempted to copy a directory to a subdirectory of itself
truncate_failed	Failed to truncate file

`created_ts` `timestamp` ***Read-only***
task creation timestamp

`started_ts` `timestamp` ***Read-only***
task start timestamp

`done_ts` `timestamp` ***Read-only***
task end timestamp

`duration` `int` ***Read-only***
task duration in seconds

`progress` `int` ***Read-only***
task progress in percent (scaled by 100)

`eta` `int` ***Read-only***
estimated time remaining before the task completion (in seconds)

`from` `string` ***Read-only***
current source file (if available)

`to` `string` ***Read-only***
current destination file (if available)

`nfiles` `int` ***Read-only***
number of files to process

`nfiles_done` `int` ***Read-only***
number of files processed

`total_bytes` `int` ***Read-only***
total bytes to process

`total_bytes_done` `int` ***Read-only***
number of bytes processed

`curr_bytes` `int` ***Read-only***
size of the file currently processed

`curr_bytes_done` `int` ***Read-only***
number of bytes processed for the current file

`rate` `int` ***Read-only***
processing rate in byte/s

`src[]` `array of string` ***Read-only***
task source files

`dst` `string` ***Read-only***
task destination path

List every tasks

`GET /api/v15/fs/tasks/`

Returns the collection of all FsTask tasks

Example request:

```
GET /api/v15/fs/tasks/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
   success: true,
   result: [
      {
         curr_bytes_done: 0,
         total_bytes: 0,
         nfiles_done: 0,
         started_ts: 1355834253,
         duration: 3,
         done_ts: 0,
         curr_bytes: 0,
         type: "extract",
         to: "oxygennosvg/128x128/mimetypes/application_x_nzb.png",
         id: 12,
         nfiles: 0,
         created_ts: 1355834253,
         state: "paused",
         total_bytes_done: 0,
         from: "/Disque dur/tests/oxygennosvg.tar.gz",
         rate: 0,
         eta: 0,
         error: "none",
         progress: 0
         src: [
```

```
          "/Disque dur/tests/oxygennosvg.tar.gz"
        ],
        dst: "/Disque dur/tests/oxygennosvg"
      },
      {
        id: 11,
        curr_bytes_done: 0,
        total_bytes: 0,
        nfiles_done: 0,
        started_ts: 1355834187,
        duration: 0,
        done_ts: 1355834187,
        curr_bytes: 0,
        type: "rm",
        to: "",
        nfiles: 0,
        created_ts: 1355834187,
        state: "done",
        total_bytes_done: 0,
        from: "/Disque dur/test/testiso.1.iso",
        rate: 0,
        eta: 0,
        error: "none",
        progress: 100,
        src: [
          "/Disque dur/test/testiso.1.iso"
        ]
      }
    ]
}
```


List a task

`GET /api/v15/fs/tasks/{id}`

Returns the FsTask task with the given id

Example request:

```
GET /api/v15/fs/tasks/12 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
   success: true,
   result: {
      curr_bytes_done: 0,
      total_bytes: 0,
      nfiles_done: 0,
      started_ts: 1355834253,
      duration: 268,
      done_ts: 0,
      curr_bytes: 0,
      type: "extract",
      to: "oxygennosvg/16x16/actions/format_stroke_color.png",
      id: 12,
      nfiles: 0,
      created_ts: 1355834253,
      state: "running",
      total_bytes_done: 0,
      from: "/Disque dur/tests/oxygennosvg.tar.gz",
      rate: 0,
      eta: 0,
      error: "none",
      progress: 0,
      src: [
        "/Disque dur/tests/oxygennosvg.tar.gz"
      ],
      dst: "/Disque dur/tests/oxygennosvg"
   }
}
```


Delete a task

`DELETE /api/v15/fs/tasks/{id}`

Deletes the FsTask task with the given id, if the task was running, stop it.

No rollback is done, if a file as already been processed it will be left as is.

Example request:

```
DELETE /api/v15/fs/tasks/12 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Update a task

`PUT /api/v15/fs/tasks/{id}`

Updates the FsTask task with the given id

Example request:

```
PUT /api/v15/fs/tasks/15 HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
   "state": "paused"
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "curr_bytes_done": 0,
        "total_bytes": 2410125312,
```

```
        "nfiles_done": 0,
        "started_ts": 1355835094,
        "duration": 27,
        "done_ts": 0,
        "curr_bytes": 0,
        "type": "cp",
        "to": "/Disque dur/old_hdd/testiso.1.iso",
        "id": 15,
        "nfiles": 1,
        "created_ts": 1355835094,
        "state": "paused",
        "total_bytes_done": 595591168,
        "from": "/Disque dur/old_hdd/testiso.iso",
        "rate": 0,
        "eta": 85,
        "error": "none",
        "progress": 24,
        "src": [
          "/Disque dur/old_hdd/testiso.iso"
        ],
        "dst": "/Disque dur/old_hdd"
    }
}
```


Listing

File info

`FileInfo`

`path` `string` ***Read-only***
file path (encoded in base64 as explained in Path Encoding)

`name` `string` ***Read-only***
file name (in clear text)

`mimetype` `string` ***Read-only***
file mimetype

`type` `enum`

Type	Description
dir	Directory
file	Regular file

`size` `int` ***Read-only***
file size in bytes

`modification` `int` ***Read-only***
file modification timestamp

`index` `int` ***Read-only***
display order for natural sort

`link` `boolean` ***Read-only***
is this file a link

`target` `string` ***Read-only***
symlink target path (encoded in base64 as explained in Path Encoding) (only present when link is set to true)

`hidden` `boolean` ***Read-only***
should the file be hidden to user

`foldercount` `int` ***Read-only***
number of subfolders

only relevant for dir, only provided if “countSubFolder” parameter is set

`filecount` `int` ***Read-only***
number of files inside directory

only relevant for dir, only provided if “countSubFolder” parameter is set

`exif` `object` ***Read-only***
EXIF metadada if available.

only relevant for supported image files (JPEG, HEIC), when the “exifMode” parameter is set

List files

`GET /api/v15/fs/ls/{path}`

Returns the list of `FileInfos` for the given path

Parameters

- **onlyFolder** (*bool*) – Only list folders
- **countSubFolder** (*bool*) – Return files and subfolder count for folders
- **removeHidden** (*bool*) – Don’t return hidden files in directory listing
- **exifMode** (*string*) – Return EXIF metadata for supported image files (JPEG, HEIC). Value can be “light” (basic metadata), “full” (all metadata) or “base64” (all metadata encoded in base64)
- **limit** (*integer*) – Maximum number of entries in response [optional]
- **cursor** (*string*) – Opaque value to include in next request to continue path listing [optional]

Example request:

```
GET /api/v15/fs/ls/L0Rpc3F1ZSBkdXI=&limit=100 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{

    "success": true,
    "result": {
      "entries": [
        {
            "path": "L0Rpc3F1ZSBkdXIvRW5yZWdpc3RyZW1lbnRz",
            "filecount": 0,
            "link": false,
            "modification": 1362005535,
            "foldercount": 0,
            "name": "Enregistrements",
            "index": 1,
            "mimetype": "inode/directory",
            "hidden": false,
            "type": "dir",
            "size": 4096
```

```
        },

        /* Note: for the two following folders path are different, but name is utf8 equivalent */

        {
            "path": "L0Rpc3F1ZSBkdXIvTGUgU3DDqWNpYWwgMg==",
            "filecount": 0,
            "link": false,
            "modification": 1362492511,
            "foldercount": 0,
            "name": "Le Spécial 2",
            "index": 3,
            "mimetype": "inode/directory",
            "hidden": false,
            "type": "dir",
            "size": 4096
        },
        {
            "path": "L0Rpc3F1ZSBkdXIvTGUgU3BlzIFjaWFsIDI=",
            "filecount": 4,
            "link": false,
            "modification": 1361995307,
            "foldercount": 1,
            "name": "Le Spécial 2",
            "index": 4,
            "mimetype": "inode/directory",
            "hidden": false,
            "type": "dir",
            "size": 4096
        },

         [ ... ]

        {
            "path": "L0Rpc3F1ZSBkdXIvVmlkw6lvcw==",
            "filecount": 8,
            "link": false,
            "modification": 1361887598,
            "foldercount": 2,
            "name": "Vidéos",
            "index": 16,
            "mimetype": "inode/directory",
            "hidden": false,
            "type": "dir",
            "size": 4096
        }
    ],
    "cursor": "eyJvZmZzZXQiOjIwMTMwMzk5MTQ2NzU5MzM4OTR9"
  }

}
```

Get file information

GET /api/v15/fs/info/{path}

Returns the `FileInfos` for the given path

Example request:

```
GET /api/v15/fs/info/L0Rpc3F1ZSBkdXIvdG90bw== HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "type": "dir",
        "link": true,
        "parent": "L0Rpc3F1ZSBkdXI=",
        "modification": 1370354349,
        "hidden": false,
        "mimetype": "inode/directory",
        "name": "toto",
        "target": "L0Rpc3F1ZSBkdXIvUGhvdG9z",
        "path": "L0Rpc3F1ZSBkdXIvdG90bw==",
        "size": 4096
    }
}
```

Batch file information

POST /api/v15/fs/info

Returns a `FileInfos` list for a given path list. Invalid paths are ignored.

Example request:

```
POST /api/v15/fs/info HTTP/1.1
Host: mafreebox.freebox.fr
```

```
[ "L0Rpc3F1ZSBkdXIvRW5yZWdpc3RyZW1lbnRz", "L0Rpc3F1ZSBkdXIvTGUgU3DDqWNpYWwgMg==", "L0Rpc3F1ZSBkdXIvVmlkw6lvcw==" ]
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
          {
              "path": "L0Rpc3F1ZSBkdXIvRW5yZWdpc3RyZW1lbnRz",
              "filecount": 0,
              "link": false,
              "modification": 1362005535,
              "foldercount": 0,
              "name": "Enregistrements",
              "index": 1,
              "mimetype": "inode/directory",
              "hidden": false,
              "type": "dir",
              "size": 4096
          },
          {
              "path": "L0Rpc3F1ZSBkdXIvTGUgU3DDqWNpYWwgMg==",
```

```
            "filecount": 0,
            "link": false,
            "modification": 1362492511,
            "foldercount": 0,
            "name": "Le Spécial 2",
            "index": 3,
            "mimetype": "inode/directory",
            "hidden": false,
            "type": "dir",
            "size": 4096
        },
        {
            "path": "L0Rpc3F1ZSBkdXIvVmlkw6lvcw==",
            "filecount": 8,
            "link": false,
            "modification": 1361887598,
            "foldercount": 2,
            "name": "Vidéos",
            "index": 16,
            "mimetype": "inode/directory",
            "hidden": false,
            "type": "dir",
            "size": 4096
        }
    ]
}
```


Operations

Each time you want to perform a modification on the file system you will have to create a new `FsTask` that you will be able to monitor.

NOTE: The requested operation may be en-queued to avoid performance drop because of excessive disk io

Conflict resolution

For certain file operations where a file name conflict can happen, you must specify a conflict resolution mode.

Valid resolution modes are:

Conflict mode	Description
overwrite	Overwrite the destination file
both	Keep both files (rename the file adding a suffix)
recent	Only overwrite if newer than destination file
skip	Keep the destination file

Move files

`POST /api/v15/fs/mv/`

Parameters

- **files** (*string[]*) – The list of files to move
- **dst** (*string*) – The destination
- **mode** (*enum*) – The conflict resolution mode

Example request for moving files:

```
POST /api/v15/fs/mv/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "files":
      [
         "L0Rpc3F1ZSBkdXIvUGhvdG9zL0RTQ18zNDkxLmpwZw==", /* /Disque dur/Photos/DSC_3491.jpg */
         "L0Rpc3F1ZSBkdXIvUGhvdG9zL0RTQ18zNTAwLmpwZw==" /* /Disque dur/Photos/DSC_3500.jpg */
      ],
    "dst": "L0Rpc3F1ZSBkdXIvUGhvdG9zL0xhdW5jaHBhZA==", /* /Disque dur/Photos/Launchpad */
    "mode": "overwrite"
}
```

Example response:


```
{
    "success": true,
    "result": {
        "curr_bytes_done": 0,
        "total_bytes": 0,
        "nfiles_done": 0,
        "started_ts": 1355840585,
        "duration": 0,
        "done_ts": 0,
        "curr_bytes": 0,
        "type": "mv",
        "to": "",
        "id": 39,
        "nfiles": 0,
        "created_ts": 1355840585,
        "state": "running",
        "total_bytes_done": 0,
        "from": "",
        "rate": 0,
        "eta": 0,
        "error": "none",
        "progress": 0,
        "src": [
          "/Disque dur/Photos/DSC_3491.jpg",
          "/Disque dur/Photos/DSC_3500.jpg"
        ],
        "dst": "/Disque dur/Photos/Launchpad"
    }
}
```


Copy files

`POST /api/v15/fs/cp/`

Parameters

- **files** (*string[]*) – The list of files to copy
- **dst** (*string*) – The destination
- **mode** (*enum*) – The conflict resolution mode

Example request:

```
POST /api/v15/fs/cp/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "files":
      [
         "L0Rpc3F1ZSBkdXIvUGhvdG9zL0xhdW5jaHBhZC9EU0NfMzQ5MS5qcGcK", /* /Disque dur/Photos/Launchpad/DSC_3491.jpg */
         "L0Rpc3F1ZSBkdXIvUGhvdG9zL0xhdW5jaHBhZC9EU0NfMzUwMC5qcGcK", /* /Disque dur/Photos/Launchpad/DSC_3500.jpg */
      ],
    "dst": "L0Rpc3F1ZSBkdXIvUGhvdG9zL1JvY2tldHMK", /* /Disque dur/Photos/Rockets */
    "mode": "both"
 }
```

Example response:

```
{
    "success": true,
    "result": {
        "curr_bytes_done": 0,
        "total_bytes": 0,
        "nfiles_done": 0,
        "started_ts": 1355840943,
        "duration": 0,
        "done_ts": 0,
        "curr_bytes": 0,
        "type": "cp",
        "to": "",
        "id": 43,
        "nfiles": 0,
        "created_ts": 1355840943,
        "state": "running",
        "total_bytes_done": 0,
        "from": "",
        "rate": 0,
        "eta": 0,
        "error": "none",
        "progress": 0,
        "src": [
          "/Disque dur/Photos/Launchpad/DSC_3491.jpg",
          "/Disque dur/Photos/Launchpad/DSC_3500.jpg"
        ],
        "dst": "/Disque dur/Photos/Rockets"
    }
}
```

Remove files

`POST /api/v15/fs/rm/`

Parameters

- **files** (*string[]*) – The list of files to remove

Example request:

```
POST /api/v15/fs/rm/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "files":
      [
         "L0Rpc3F1ZSBkdXIvUGhvdG9zL1JvY2tldHMvRFNDXzM0OTEuanBnCg==", /* /Disque dur/Photos/Rockets/DSC_3491.jpg */
         "L0Rpc3F1ZSBkdXIvUGhvdG9zL1JvY2tldHMvRFNDXzM1MDAuanBnCg==" /* /Disque dur/Photos/Rockets/DSC_3500.jpg */
      ]
 }
```

Example response:

```
{
    "success": true,
    "result": {
        "curr_bytes_done": 0,
        "total_bytes": 0,
        "nfiles_done": 0,
        "started_ts": 1355841064,
        "duration": 0,
        "done_ts": 0,
        "curr_bytes": 0,
        "type": "rm",
        "to": "",
        "id": 45,
        "nfiles": 0,
        "created_ts": 1355841064,
        "state": "running",
        "total_bytes_done": 0,
        "from": "/Disque dur/Photos/Rockets/DSC_3491.jpg",
        "rate": 0,
        "eta": 0,
        "error": "none",
        "progress": 0,
        "src": [
          "/Disque dur/Photos/Rockets/DSC_3491.jpg",
          "/Disque dur/Photos/Rockets/DSC_3500.jpg"
        ]
    }
}
```

Cat files

`POST /api/v15/fs/cat/`

Parameters

- **files** (*string[]*) – The list of files to concatenate
- **dst** (*string*) – The destination
- **multi_volumes** (*bool*) – Enable multi-volumes mode, it will start at XXX001 and concatenate XXX002, XXX003, …
- **delete_files** (*bool*) – Deletes source files
- **overwrite** (*bool*) – Overwrites the destination
- **append** (*bool*) – Append to the destination

Example request:

```
POST /api/v15/fs/cat/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "files":
      [
         "L0Rpc3F1ZSBkdXIvZmlsZTE=", /* /Disque dur/file1 */
         "L0Rpc3F1ZSBkdXIvZmlsZTI="  /* /Disque dur/file2 */
      ],
```

```
    "dst": "L0Rpc3F1ZSBkdXIvZmlsZTEy", /* /Disque dur/file12 */
    "multi_volumes": false,
    "delete_files": false,
    "append": true,
    "overwrite": false
}
```

Or if you want to do a multi-volumes concatenation:

```
{
   "files":
     [
        // You don't need to specify file002, file003, ...
        // They'll be found by cat.
        "L0Rpc3F1ZSBkdXIvZmlsZTAwMQ==", /* /Disque dur/file001 */
     ],
    "dst": "L0Rpc3F1ZSBkdXIvZmlsZQ==", /* /Disque dur/file */
    "multi_volumes": true,
    "delete_files": true,
    "append": false,
    "overwrite": true
}
```

Example response:

```
{
    "success": true,
    "result": {
        "curr_bytes_done": 0,
        "total_bytes": 0,
        "nfiles_done": 0,
        "started_ts": 1355840943,
        "duration": 0,
        "done_ts": 0,
        "curr_bytes": 0,
        "type": "cat",
        "to": "",
        "id": 43,
        "nfiles": 0,
        "created_ts": 1355840943,
        "state": "running",
        "total_bytes_done": 0,
        "from": "",
        "rate": 0,
        "eta": 0,
        "error": "none",
        "progress": 0
    }
}
```

Create an archive

`POST /api/v15/fs/archive/`

Parameters

- **files** (*string[]*) – The list of files to archive
- **dst** (*string*) – The destination

Example request:

```
POST /api/v15/fs/archive/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "files":
     [
        "L0Rpc3F1ZSBkdXIvUGhvdG9zL0xhdW5jaHBhZC9EU0NfMzQ5MS5qcGc=", /* /Disque dur/Photos/Launchpad/DSC_3491.jpg */
        "L0Rpc3F1ZSBkdXIvUGhvdG9zL0xhdW5jaHBhZC9EU0NfMzUwMC5qcGc="  /* /Disque dur/Photos/Launchpad/DSC_3500.jpg */
     ],
    "dst": "L0Rpc3F1ZSBkdXIvUGhvdG9zL3JvY2tldHMuemlw" /* /Disque dur/Photos/rockets.zip */
 }
```

Example response:

```
{
    "success": true,
    "result": {
        "curr_bytes_done": 0,
        "total_bytes": 0,
        "nfiles_done": 0,
        "started_ts": 1355840943,
        "duration": 0,
        "done_ts": 0,
        "curr_bytes": 0,
        "type": "archive",
        "to": "",
        "id": 42,
        "nfiles": 0,
        "created_ts": 1355840943,
        "state": "running",
        "total_bytes_done": 0,
        "from": "",
        "rate": 0,
        "eta": 0,
        "error": "none",
        "progress": 0,
        "src": [
          "/Disque dur/Photos/Launchpad/DSC_3491.jpg",
          "/Disque dur/Photos/Launchpad/DSC_3500.jpg"
        ],
        "dst": "/Disque dur/Photos/rockets.zip"
    }
}
```

Extract a file

`POST /api/v15/fs/extract/`

Parameters

- **src** (*string*) – The archive file
- **dst** (*string*) – The destination folder
- **password** (*string*) – The archive password
- **delete_archive** (*boolean*) – Delete archive after extraction
- **overwrite** (*boolean*) – Overwrite files on conflict

Example request:

```
POST /api/v15/fs/extract/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "src": "L0Rpc3F1ZSBkdXIvb2xkX2hkZC90ZXN0aXNvLjEuaXNv", /* /Disque dur/old_hdd/testiso.1.iso */
   "dst": "L0Rpc3F1ZSBkdXIvb2xkX2hkZA==" /* /Disque dur/old_hdd */
   "password": "",
   "delete_archive": false,
   "overwrite": true
}
```

Example response:

```
{
    "success": true,
    "result": {
        "curr_bytes_done": 0,
        "total_bytes": 0,
        "nfiles_done": 0,
        "started_ts": 1355842252,
        "duration": 0,
        "done_ts": 0,
        "curr_bytes": 0,
        "type": "extract",
        "to": "/Disque dur/old_hdd",
        "id": 48,
        "nfiles": 0,
        "created_ts": 1355842252,
        "state": "running",
        "total_bytes_done": 0,
        "from": "/Disque dur/old_hdd/testiso.1.iso",
        "rate": 0,
        "eta": 0,
        "error": "none",
        "progress": 0,
        "src": [
          "/Disque dur/old_hdd/testiso.1.iso"
        ],
        "dst": "/Disque dur/old_hdd"
    }
}
```

Repair a file

`POST /api/v15/fs/repair/`

Parameters

- **src** (*string*) – The .par2 file
- **delete_archive** (*boolean*) – Delete par2 files after repair

Example request:

```
POST /api/v15/fs/repair/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "src": "L0Rpc3F1ZSBkdXIvdGVzdHMvcGFyMi9saWNlbnNlLnR4dC5wYXIy", /* /Disque dur/tests/par2/license.txt.par2 */
   "delete_archive": false
}
```

Example response:

```
{
    "success": true,
    "result": {
        "curr_bytes_done": 0,
        "total_bytes": 0,
        "nfiles_done": 0,
        "started_ts": 1355842559,
        "duration": 0,
        "done_ts": 0,
        "curr_bytes": 0,
        "type": "repair",
        "to": "",
        "id": 50,
        "nfiles": 0,
        "created_ts": 1355842559,
        "state": "running",
        "total_bytes_done": 0,
        "from": "",
        "rate": 0,
        "eta": 0,
        "error": "none",
        "progress": 0
    }
}
```

Hash a file

`POST /api/v15/fs/hash/`

Parameters

- **src** (*string*) – The file to hash
- **hash_type** (*string*) – The type of hash (md5, sha1, …)

Example request:

```
POST /api/v15/fs/hash/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "src": "L0Rpc3F1ZSBkdXIvbXlfZmlsZQ==", /* /Disque dur/my_file */
   "hash_type": "md5"
}
```

Example response:

```
{
    "success": true,
    "result": {
        "curr_bytes_done": 0,
        "total_bytes": 4242,
        "nfiles_done": 0,
        "started_ts": 1355842559,
        "duration": 0,
        "done_ts": 0,
```

```
        "curr_bytes": 4242,
        "type": "hash",
        "to": "",
        "id": 50,
        "nfiles": 1,
        "created_ts": 1355842559,
        "state": "running",
        "total_bytes_done": 0,
        "from": "/Disque dur/my_file",
        "rate": 0,
        "eta": 0,
        "error": "none",
        "progress": 0
    }
}
```

Get the hash value
To get the hash, the task must have succeed and be in the state “done”.

`GET /api/v15/fs/tasks/{id}/hash`

Example request:

```
GET /api/v15/fs/tasks/50/hash HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "hash": "94baaad4d1347ec6e15ae35c88ee8bc8"
    }
}
```

Create a directory

Contrary to other file system tasks, this operation is done synchronously.

Instead of a returning a FsTask a call to this API will only return success status

`POST /api/v15/fs/mkdir/`

Parameters

- **parent** (*string*) – The parent directory path (base64 encoded)
- **dirname** (*string*) – The name of the directory to create

Example request:

```
POST /api/v15/fs/mkdir/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "parent": "L0Rpc3F1ZSBkdXI=", /* /Disque dur */
   "dirname": "Test"
}
```

Example response:

```
{
    "success": true
}
```

Rename a file/folder

Contrary to other file system tasks, this operation is done synchronously.

Instead of a returning a FsTask a call to this API will only return success status and the new path as a result

`POST /api/v15/fs/rename/`

Parameters

- **src** (*string*) – The source file path (base64 encoded)
- **dst** (*string*) – The new name of the file (clear text, without path)

Example request:

```
POST /api/v15/fs/rename/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "src": "L0Rpc3F1ZSBkdXIvdGVzdC50eHQ=", /* /Disque dur/test.txt */
   "dst": "plop.txt"
}
```

Example response:

```
{
    "success": true,
    "result": "L0Rpc3F1ZSBkdXIvcGxvcC50eHQ=" /* /Disque dur/plop.txt */
}
```

Download a file

`GET /api/v15/dl/{path}`

Example request:

```
GET /api/v15/dl/L0Rpc3F1ZSBkdXIvUGhvdG9zL1BsYW5zIHNlY3JldHMuanBn HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 600864
Content-Disposition: attachment; filename="Plans secrets.jpg"

[ ... ]
```

File Sharing Link

This API allows you to create a unique link to share content hosted on you Freebox.

NOTE: this feature is available only if you enable HTTP remote access to your Freebox.

File Sharing Errors

When attempting to access the file sharing API, you may encounter the following errors:

error_code	Description
invalid_id	Invalid object id
path_not_found	File or folder not found
internal_error	Internal error

File Sharing Link object

Share link have the following attributes:

ShareLink

- token string ***Read-only***
  The link unique sharing token
- path string ***Read-only***
  The root path of the share, if the path is a regular file, only this file will be shared
- name string ***Read-only***
  The readable name of the shared file/folder
- expire timestamp ***Read-only***
  Link expiration timestamp, 0 means no expiration.
- fullurl string ***Read-only***
  Full URL to use for remote access. If remote access is disabled, the field will be empty.

File Sharing Link API

Retrieve a File Sharing link

GET /api/v8/share_link/

Returns the collection of all ShareLink

Example request:

```
GET /api/v8/share_link/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   success: true,
   result: [
      {
          "path": "L0Rpc3F1ZSBkdXIvUGhvdG9zL01lcyB2YWNhbmNlcyBlbiByb3Vsb3R0ZQ==" /* /Disque dur/Photos/Mes vacances en roulotte */
          "name": "Mes vacances en roulotte",
          "token": "gAnweF2Xg5OwcJWn",
          "expire": 1355852344,
          "fullurl": "http://13.37.42.69/api/v8/share/gAnweF2Xg5OwcJWn/"
      },
      {
          "path": "L0Rpc3F1ZSBkdXIvc2hhcmVk", /* /Disque dur/shared */
          "name": "shared",
          "token": "s8a+4VtOQNkkQ55f",
          "expire": 1355866268,
          "fullurl": "http://13.37.42.69/api/v8/share/s8a+4VtOQNkkQ55f/"
      }
   ]
}
```

GET /api/v8/share_link/{token}

Returns the ShareLink task with the given id

Example request:

```
GET /api/v8/share_link/gAnweF2Xg5OwcJWn HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "path": "L0Rpc3F1ZSBkdXIvUGhvdG9zL01lcyB2YWNhbmNlcyBlbiByb3Vsb3R0ZQ==" /* /Disque dur/Photos/Mes vacances en roulotte */
        "name": "Mes vacances en roulotte",
        "token": "gAnweF2Xg5OwcJWn",
        "expire": 1355852344,
        "fullurl": "http://13.37.42.69/api/v8/share/gAnweF2Xg5OwcJWn/"
    }
}
```

Delete a File Sharing link

DELETE /api/v8/share_link/{token}

Deletes the ShareLink task with the given token, if the task was running, stop it.

No rollback is done, if a file as already been processed it will be left as is.

Example request:

```
DELETE /api/v8/share_link/gAnweF2Xg5OwcJWn HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Create a File Sharing link

POST /api/v8/share_link/

Create a new ShareLink

Example request:

```
POST /api/v8/share_link/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "path": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHM=", /* /Disque dur/Téléchargements */
   "expire": 1355932880,
   "fullurl": ""
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "path": "L0Rpc3F1ZSBkdXIvVMOpbMOpY2hhcmdlbWVudHM=", /* /Disque dur/Téléchargements */
        "name": "Téléchargements",
        "token": "6Hj57zgTfoQqb_vH",
        "expire": 1355932880,
        "fullurl": "http://13.37.42.69/api/v8/share/6Hj57zgTfoQqb_vH/"
    }
}
```

File Upload

This API allows you to upload files to the Freebox Server.

NOTE: for large transfer files, you should prefer FTP over HTTP transfer

WARNING the previous http upload method is now deprecated since api v4, you must now use the new WebSocket upload Api. If you can't support WebSocket, you must use ftp for file transfer

File Upload Errors

When attempting to access the file upload API, you may encounter the following errors:

error_code	Description
invalid_request	Invalid request
path_not_found	File or folder not found
access_denied	Write permission denied in the destination folder
destination_conflict	A file with same name already exists
invalid_id	Invalid file upload id
cancelled	Someone on a side channel as cancelled the upload
noent	No upload with this id

File Upload object

File uploads have the following attributes:

`FileUpload`

`id` `int` ***Read-only***
upload id

`size` `int` ***Read-only***
Upload file size in bytes

`uploaded` `int` ***Read-only***
Uploaded bytes

`status` `enum` ***Read-only***
upload status can have the following values

status	Description
authorized	Upload authorization is valid, upload has not started yet
in_progress	Upload in progress
done	Upload done
failed	Upload failed
conflict	Destination file conflict
timeout	Upload authorization is no longer valid
cancelled	Upload cancelled by user

`start_date` `timestamp` ***Read-only***
upload start date

`last_update` `timestamp` ***Read-only***
last update of file upload object

`upload_name` `string` ***Read-only***
name of the file uploaded

`dirname` `string` ***Read-only***
upload destination directory

WebSocket File Upload API

The file upload WebSocket path is /api/v8/ws/upload

With this new API, the need for creating a 'file upload authorization' has now been removed.

To be able to upload a file to the Freebox, you must open a WebSocket connection to the upload api, then for each file you want to upload you must :

- send a `FileUploadStartAction` with the action 'upload_start'
- wait for the associated `WebSocketResponse` that indicates success, then start transferring the file content by chunks, each chunk being a binary WebSocket frame.

  For each chunk you send, you'll get a WsUploadProgress response indicating that the associated chunk has been received and processed. Note that you should not wait for this response before sending the next data chunk in order to get good bandwidth performance.
- once all chunks have been transferred, you should send a `FileUploadFinalizeAction` with the action 'upload_finalize' and wait for the associated `WebSocketResponse` indicating success

Note that if you have multiple files to send, you should reuse the same WebSocket connection, and repeat the upload steps again.

If for any reason the WebSocket is closed during upload, the partially sent file will be left as-is on the Freebox to allow resuming upload at a later point.

If you want to cancel an ongoing upload ou can send a `FileUploadCancelAction`. The partially uploaded file will then be deleted

File Upload Start Action

`FileUploadStartAction`

request_id int
optional request_id

action string
must be 'upload_start'

size int
optional file size

dirname string
the destination directory (encoded value)

filename string
the destination filename

force enum
select the way conflicts are handled

Force mode	Description
missing	The response to the FileUploadStartAction will be an error with 'destination_conflict' if the destination file already exists. The response will also contain a file_size attribute containing the existing file length (useful for resuming upload)
overwrite	If the target file already exists it will be overridden
resume	The upload will resume, all sent chunks will then be appended to the existing file.

File Upload Finalize action

FileUploadFinalizeAction

request_id int
optional request_id

action string
must be 'upload_finalize'

File Upload Cancel action

FileUploadCancelAction

request_id int
optional request_id

action string
must be 'upload_cancel'

File Upload Chunk

File upload chunk are just Binary WebSocket frames containing raw file content.

File Upload Chunk Response

For each received chunk, the Freebox will send a chunk response containing upload progress information the request_id used in response will be the one from the FileUploadStartAction, and 'action' value will be 'upload_data'

FileUploadChunkResponse

total_len int
target file current length

complete bool
will be true in a reply to FileUploadFinalizeAction or FileUploadCancelAction

cancelled bool
will be true in a reply FileUploadCancelAction

File Upload example

GET /api/v8/ws/upload

Start the WebSocket handshake:

Client ==> Freebox

```
GET ws://mafreebox.freebox.fr/api/v8/ws/upload HTTP/1.1
Host: mafreebox.freebox.fr
Connection: Upgrade
Upgrade: websocket
Sec-WebSocket-Version: 13
Sec-WebSocket-Key: LhYCx4FBJE6pqrIL3tDC3g==
X-Fbx-App-Auth: 35JYdQSvkcBYK84IFMU7H86clfhS75OzwlQrKlQN1gBch\/Dd62RGzDpgC7YB9jB2
```

Handshake response:

Client <== Freebox

```
HTTP/1.1 101 Switching Protocols
Connection: upgrade
Upgrade: websocket
Sec-WebSocket-Accept: IqwCz8z8sON/eWQqkYKLu6iLkzo=
```

Start upload:

Client ==> Freebox

```
{
  "action": "upload_start",
  "request_id": 3615,
  "size": 8526224,
  "dirname": "L0Rpc3F1ZSBkdXIvMF91cGxvYWRfdGVzdA==",
  "filename": "test_file.bin"
}
```

Start upload response:

Client <== Freebox

```
{
  "success": false,
  "action": "upload_start",
  "request_id": 3615,
  "msg": "Le fichier existe déjà",
  "file_size": 8526224,
  "error_code": "conflict"
}
```

Start upload with overwrite force mode:

Client ==> Freebox

```
{
  "action": "upload_start",
  "request_id": 6969,
  "size": 8526224,
  "dirname": "L0Rpc3F1ZSBkdXIvMF91cGxvYWRfdGVzdA==",
  "filename": "test_file.bin",
```

```
  "force": "overwrite"
}
```

Start upload response:

Client <== Freebox

```
{
  "action": "upload_start",
  "success": true,
  "request_id": 6969
}
```

Send data chunk:

Client ==> Freebox

[BINARY WEBSOCKET FRAME MESSAGE containing file offset: 0, length: 512k]

[BINARY WEBSOCKET FRAME MESSAGE containing file offset: 512k, length: 512k]

[BINARY WEBSOCKET FRAME MESSAGE containing file offset: 1024k, length: 512k]

[...]

Receive upload response:

Client <== Freebox

```
{
  "request_id": 6969
  "action": "upload_data",
  "success": true,
  "result": {
          "total_len": 524288,
          "complete": false
  },
}

{
  "request_id": 6969,
  "action": "upload_data",
  "success": true,
  "result": {
          "total_len": 1048576,
          "complete": false
  }
}

[ ... ]
```

This will be received for each sent data chunk

Send upload finalize:

Client ==> Freebox

```
{
  "action": "upload_finalize",
  "request_id":3615
}
```

Receive upload finalize confirmation:

Client <== Freebox

```
{
  "request_id": 3615,
  "action": "upload_finalize",
  "success": true,
  "result": {
    "total_len": 8526224,
    "complete": true
  }
}
```

At this point you can start uploading a new file by repeating the previous steps starting from *Start upload* step

Upload Progress tracking API

Get the list of uploads

`GET /api/v8/upload/`

Example request:

```
GET /api/v8/upload/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
        {
            "id": 1678139709,
            "size": 54960,
            "uploaded": 54960,
            "status": "done",
            "last_update": 1361465608,
            "start_date": 1361465608,
            "upload_name": "playlist.m3u",
            "dirname": "/Disque 1"
        }
    ]
}
```

Track an upload status

`GET /api/v8/upload/{id}`

With this API you can track the progress of your `FileUpload` task

Example request:

```
GET /api/v8/upload/1678139709 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "id": 1678139709,
        "size": 54960,
        "uploaded": 54960,
        "status": "done",
        "last_update": 1361465608,
        "start_date": 1361465608,
        "upload_name": "playlist.m3u",
        "dirname": "/Disque 1"
    }
}
```


Cancel an upload

`DELETE /api/v8/upload/{id}/cancel`

Cancel the given FileUpload closing the connection The upload status must be in_progress

Example request:

```
DELETE /api/v8/upload/136419941/cancel HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Delete an upload

`DELETE /api/v8/upload/{id}`

Delete the given FileUpload closing the connection if needed

Example request:

```
DELETE /api/v8/upload/136419941 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Home

Home API

The Home API allows you to access features related to home automation

List Home Adapters

Home Adapter Object

`HomeAdapter`

HomeAdapter has the following attributes:

`id` `int` ***Read-only***
: this object id

`icon_url` `String` ***Read-only***
: Url of the adapter icon

`label` `String` ***Read-only***
: The displayable name of this adapter

`status` `enum`
: Adapter status

status	Description
unplugged	The adapter is not available
disabled	The adapter has been disabled
active	the adapter is active

`type` `AdapterType` ***Read-only***
: The technical type of this adapter.

`props` `Map`
: Technical data related to this adapter, useful fo developers

Get Home Adapters List

`GET /api/v8/home/adapters`

Retrieve the list of registered HomeAdapter. A new adapters appear when the user plugs a new home automation dongle.

Example request:

```
GET /api/v8/home/adapters HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "result": [
    {
      "icon_url": "http://images.com/adapter_dm.png",
      "id": 1,
      "label": "Gestionnaire de caméra",
      "status": "active",
      "type": {
        "name": "adapter::cam"
      }
    },
    {
      "icon_url": "http://images.com/adapter_dm.png",
      "id": 2,
      "label": "Réseau Rts",
      "status": "active",
```

```
      "type": {
        "name": "adapter::rts"
      }
    },
    {
      "icon_url": "http://images.com/adapter_dm.png",
      "id": 3,
      "label": "Réseau IOHome",
      "props": {
        "Addr": 160,
        "SomfyId": "00:00:00:00"
      },
      "status": "active",
      "type": {
        "name": "adapter::ios"
      }
    },
    {
      "icon_url": "http://images.com/adapter_dm.png",
      "id": 4,
      "label": "Réseau Domus",
      "props": {
        "Network ID": 50791
      },
      "status": "active",
      "type": {
        "name": "adapter::domus"
      }
    }
  ],
  "success": true
}
```


Get a Home Adapter

`GET /api/v8/home/adapters/{id}`

Fetch information about a single HomeAdapter.

Example request:

```
GET /api/v8/home/adapters/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "result": {
      "icon_url": "http://images.com/adapter_dm.png",
      "id": 1,
      "label": "Gestionnaire de caméra",
      "status": "active",
      "type": {
        "name": "adapter::cam"
      }
  }
  "success": true
}
```


Change a Home Adapter status

`PUT /api/v8/home/adapters/{id}`

Change the status of a HomeAdapter.

Example request:

```
PUT /api/v8/home/adapters/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "status": "disabled"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true
}
```


Pair a new object

Pairing Step

`HomePairingStep`

This represents a pairing process step.

`fields[] array of HomePairingStepField` ***Read-only***

A collection of ui elements to display.

`icon_url String` ***Read-only***

The url of an image which represents this step.

`pageid int` ***Read-only***

The identifier of this step.

`refresh int`

The delay in millisecond after which to request a new step update.

`session int` ***Read-only***

The id of this session process.

`HomePairingStepField`

`widget enum` ***Read-only***

The type of ui element to display.

widget	Description
label	A simple text field
select	A selectable list item
button	A clickable button
display_qrcode	A qrcode

widget	Description
input	An input text field
checkbox	A checkable button
progress	A progress bar
bar_button_left	A button displayed at the left of the bottom nav bar
bar_button_right	A button displayed at the right of the bottom nav bar

`text` `string` ***Read-only***

The data to use with the displayed widget.

widget	text usage
label	The label text
select	The item caption
button	The button caption
display_qrcode	The data to encode in the qrcode
input	The default text
checkbox	The button caption
progress	The progress value, in percent, as int
bar_button_left	The button caption
bar_button_right	The button caption

Start Pairing

`POST /api/v8/home/pairing/{adapter_id}`

Start the pairing process on a specific HomeAdapter.

op: start

type: the type of object to pair. This parameter is only relevant for the domus adapter.

type	Description
node::domus::freebox::secmod	Pair the security module
node::domus::sercomm::pir	Pair a movement detector
node::domus::sercomm::keyfob	Pair an alarm remote control
node::domus::sercomm::doorswitch	Pair an opening detector

Example request:

```
POST /api/v8/home/pairing/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "op": "start",
  "type": "node::domus::freebox::secmod"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true
}
```

Current Pairing Step

`GET /api/v8/home/pairing/{adapter_id}`

Get the current HomePairingStep on a specific HomeAdapter

Example request:

```
GET /api/v8/home/pairing/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "result" : {
      "fields" : [
          {
            "text" : "Veuillez vérifier que votre wifi est bien activé.",
            "widget" : "label"
          }
        ],
      "icon_url" : "/resources/images/home/pairing/wifi.png",
      "pageid" : 2,
      "refresh" : 1000,
      "session" : 62328
    },
  "success" : true
}
```

Next Step

`POST /api/v8/home/pairing/{adapter_id}`

Send current step result and get the next step in the process. Call this when the user clicks on a button, bar_button_left, bar_button_right or a select item.

field is a list of value corresponding to the current page widgets.

widget	value in fields
label	null
select	The index of the selected item, null if none selected
button	true if the button has been clicked, false otherwise
display_qrcode	null
input	The text entered
checkbox	true if checked, false otherwise
progress	The progress value, in percent, as int
bar_button_left	The button caption

widget	value in fields
bar_button_right	The button caption

Example request:

```
POST /api/v8/home/pairing/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "op": "next",
  "session": "659887",
  "pageid": "1".
  "fields": [null,null,"mon texte", false, true]
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "result" : {
      "fields" : [
          {
            "text" : "Veuillez vérifier que votre wifi est bien activé.",
            "widget" : "label"
          }
        ],
      "icon_url" : "/resources/images/home/pairing/wifi.png",
      "pageid" : 2,
      "refresh" : 1000,
      "session" : 62328
    },
  "success" : true
}
```

Stop Pairing

`POST /api/v8/home/pairing/{adapter_id}`

Stop the pairing process on a specific HomeAdapter.

op: stop session: the id of the pairing session to stop

Example request:

```
POST /api/v8/home/pairing/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "op": "stop",
  "session": 15645
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true
}
```

Home Nodes

Acces objects connected to the automation network.

Home Node Object

`HomeNode`

`adapter` int ***Read-only***
Id of the HomeAdapter this node is connected to.

`category` String ***Read-only***
???

`id` int ***Read-only***
Id of this node.

`label` String ***Read-only***
Displayable name of this node

`name` String ***Read-only***
Technical name of this node

`show_endpoints[]` array of HomeNodeEndpoint ***Read-only***
Endpoints exposed by this node

`signal_links[]` array of HomeNodeLink ***Read-only***
Links from other objects to this node signals

`slot_links[]` array of HomeNodeLink ***Read-only***
Links from other objects to this node slots

`status` enum ***Read-only***
Status of this node

status	Description
unreachable	The adapter is not reachable
disabled	The node has been disabled
active	The node is connected
unpaired	The node has not been paired to any network

`type` HomeNodeType ***Read-only***
Node type info

`HomeNodeEndpoint`

`category` String ***Read-only***
???

`ep_type` enum ***Read-only***
The endpoint type

ep_type	Description
signal	The endpoint outputs an information
slot	A endpoint that controls the object

`id` `int` ***Read-only***

The endpoint id

`visibility` `enum` ***Read-only***

Visibility level of this endpoint

visibility	Description
internal	For internal use only, never exposed
normal	The endpoint is available for scenarii but does not display info to the user
dashboard	The endpoint expose data that can be displayed on UI

`access` `enum` ***Read-only***

Access mode of this endpoint

access	Description
r	Read only
w	Write only
rw	Read and write

`HomeNodeType`

`icon` `String` ***Read-only***

The node icon name or url

`label` `String` ***Read-only***

The node displayable type

`label name` ***Read-only***

The node type technical name

`physical` `boolean` ***Read-only***

True when the node is an actual connected object, false when it's a virtual node

`HomeNodeEndpointUi`

`display` `enum` ***Read-only***

Display mode of this data

display	Description
text	This displays the endpoint value as text. Read access is always allowed when "text" is used. When write access is allowed, the text may be editable on user request. When the "unit" entry is present and not null, it specifies the physical unit associated to the endpoint value.
icon	This displays the icon fetched from "icon_url" with % being replaced by the string representation of the endpoint value. For *string* value type, the % is replaced by the endpoint value. For *int* and *float* value types, this requires an "icon_ranges" array of threshold values. The % is replaced by the index in the "range" array which is just below the endpoint value. For *boolean* value type, the % is replaced by "on" or "off". When the "value" is null, the % is replaced by the empty string. Read access is always allowed when "icon" is used. Write access is not used.
button	This displays a push button. Write access is always allowed when "button" is used. A null value must be send to the endpoint when pushed.
slider	This displays a slider with the cursor located according to the endpoint value in the range specified by "range". Read access is always allowed when "slider" is used. When write access is allowed, the cursor may be moved by the user. When write access is not allowed it may be displayed as a progress bar.
toggle	This displays an on/off switch. Read access is always allowed when "switch" is used. When write access is allowed, switch may be toggled by the user. A *boolean* value must be send to the endpoint when toggled.
color	This displays a color value. The value type is an *int* representing the RGB color. Read access is always allowed when "color" is used.
warning	This display the icon fetched from "icon_url" when the value condition is true. For *boolean* value type, the value is the condition. For *int* and *float* value types, this requires a "range" of size 2. If the value is within the range, the condition is true.

`icon_url` `String` ***Read-only***

Url or name of the icon to display. The icon may be displayed for any value of "display".

`unit` `String` ***Read-only***

The unit of the value to display.

`icon_color` `String` ***Read-only***

The hexadecimal presentation of the tint to apply to the icon fetched from "icon_url".

`text_color` `String` ***Read-only***

The hexadecimal presentation of the color of this endpoint label.

`value_color` `String` ***Read-only***

The hexadecimal presentation of the color of this endpoint value.

`range[]` `array of double` ***Read-only***

Range of array of threshold values for this endpoint value.

`icon_color_range[]` `array of String` ***Read-only***

A range of colors to choose from instead of "icon_color". The index in the range is the index in the "range" array which is just below the endpoint value.

`text_color_range[]` `array of String` ***Read-only***

A range of colors to choose from instead of "text_color". The index in the range is the index in the "range" array which is just below the endpoint value.

`value_color_range[]` `array of String` ***Read-only***

A range of colors to choose from instead of "value_color". The index in the range is the index in the "range" array which is just below the endpoint value.

`status_text_range[]` `array of String` ***Read-only***

Text values to display instead of the value itself. The index in the range is the index in the "range" array which is just below the endpoint value.

Get Home Nodes

`GET /api/v8/home/nodes`

Get the list of HomeNode A node is either a physical home automation device or a virtual black box used to interact with other nodes. Physical nodes are associated to an adapter. Nodes may have slot and signal endpoints. They can be used to interact with the node from the user interface. They can also be connected together using links.

Example request:

```
GET /api/v8/home/nodes HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": [
      [...]
  ]
}
```

Get a Home Node

`GET /api/v8/home/nodes/{id}`

Get a specific HomeNode

Example request:

```
GET /api/v8/home/nodes HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
      [...]
    }
  }
}
```

Rename a Home Node

`PUT /api/v8/home/nodes/{id}`

Rename a HomeNode

Example request:

```
PUT /api/v8/home/nodes HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "label": "Mon objet"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true
}
```

Delete a Home Node

`DELETE /api/v8/home/nodes/{id}`

Remove a HomeNode from the automation network. The object will need to be paired again if the node is physical.

Example request:

```
DELETE /api/v8/home/nodes HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true
}
```

Home Nodes Values

Endpoint value object

`HomeNodeEndpointValue`

`value` String ***Read-only***

The current value of the endpoint

`unit` String ***Read-only***

The displayable unit of the value

`refresh` int ***Read-only***

The period this value need to be refreshed

`value_type` enum ***Read-only***

The type of value this endpoint expose

value_type
bool
int
float
void

Fetch Endpoint Value

`GET /api/v8/home/endpoints/{node_id}/{endpoint_id}`

Retrieve the current value of the specified node endpoint. The last pushed value is returned for slot endpoints. For signal endpoint, the value is retrieved directly from the node specific back-end.

Example request:

```
GET /api/v8/home/endpoints/14/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "result": {
    "value": false,
    "value_type": "bool"
  },
  "success": true
}
```

Change Endpoint Value

`PUT /api/v8/home/endpoints/{node_id}/{endpoint_id}`

Push a value to the specified node slot endpoint. Only slot endpoint accept this operation.

Example request:

```
PUT /api/v8/home/endpoints/14/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "value": true
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true
}
```

Home Tileset

The tileset is a user-friendly representation of connected objects which expose features instead of the actual objects

HomeTileObject

`HomeTile`

`node_id` `int` ***Read-only***
Id of the `HomeNode` providing this tile data

`label` `String` ***Read-only***
Displayable label of this tile

`action` `enum` ***Read-only***
Action provided by this tile

action	Description
tileset	Open the related node sub-tileset
graph	Open a graph detail page
store	Display a store simple command
store_slider	Display a store slider command
color_picker	Display a color selection widget
heat_picker	Display a white tone selection widget
intensity_picker	Display an intensity selection widget
none	No action

`type` `enum` ***Read-only***
The type of tile to display

type	Description
action	A button tile that present no data
info	A generic tile that displays datas according to their UI field
light	A light control tile with color, intensity and head pickers
alarm_sensor	A tile representing a sensor that belongs to an alarm system
alarm_control	A tile representing an alarm system control
camera	A tile representing a camera

`group` `HomeNodeGroup` ***Read-only***
Displayable label of this tile

`data[]` `array of` `HomeTileData` ***Read-only***
Displayable label of this tile

`HomeNodeGroup`

`label` `String` ***Read-only***
The displayable name of this group

`icon_url` `String` ***Read-only***
The icon url or name

`HomeTileData`

`refresh` `int` ***Read-only***
The period this data needs to be refreshed

`label` `String` ***Read-only***
The displayable name of this data

`ep_id` `int` ***Read-only***
Id of the `HomeNodeEndpoint` related to this data

`value_type` `enum` ***Read-only***
The data value type

value_type
bool
int
float
string

`value` `String`
The data value

`value` `String` ***Read-only***
The data value history as string in the format: “timestamp:value” separated by semicolons

`ui` `HomeNodeEndpointUi` ***Read-only***
Ui descriptor for this data to know how to display it

List all Tiles

`GET /api/v8/home/tileset/all`

Get the list of all tiles.

Example request:

```
GET /api/v8/home/tileset/all HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "result" : [
      {
        "data" : [
            {
              "ep_id" : 0,
              "label" : "Trigger",
              "ui" : {
                  "access" : "rw",
                  "display" : "text"
                },
              "value" : null,
              "value_type" : "void"
            },
            {
              "category" : "alarm",
              "ep_id" : 1,
              "label" : "Alarme",
              "ui" : {
                  "access" : "rw",
                  "display" : "toggle",
                  "icon_url" : "http://lagabardine.ovh/~jeremie/img/Alarm.png"
                },
              "value" : false,
              "value_type" : "bool"
            },
            {
              "ep_id" : 2,
              "label" : "Pin Code",
              "ui" : {
                  "access" : "rw",
                  "display" : "text"
                },
              "value" : 0,
              "value_type" : "int"
            },
            {
              "ep_id" : 3,
              "label" : "Sirène",
              "refresh" : 2000,
              "ui" : {
                  "access" : "r",
                  "display" : "toggle",
                  "icon_url" : "http://lagabardine.ovh/~jeremie/img/Alarm.png"
                },
              "value" : false,
              "value_type" : "bool"
            }
          ],
        "ep_type" : "slot",
        "group" : {
            "icon_url" : "http://lagabardine.ovh/~jeremie/img/favori.png",
            "label" : ""
          },
        "node_id" : 17,
        "type" : "alarm_control"
      },
      {
        "data" : [
            {
              "ep_id" : 0,
              "history" : "1539868875260:1;1539876788228:0;1539876788530:1;1539876788796:0;1539876788850:1;1539876798829:0;1539876799143:1;1540282834199:1;1540305925
              "label" : "Fenêtre",
              "ui" : {
                  "access" : "r",
                  "display" : "icon",
                  "icon_color_range" : [
                      "#ff0000",
                      "#00ff00"
                    ],
                  "icon_url" : "home_picto_dws",
                  "status_text_range" : [
                      "Ouvert",
                      "Fermé"
                    ],
                  "value_color" : "#00ff00"
                },
              "value" : null,
              "value_type" : "bool"
            },
            {
              "ep_id" : 1,
              "history" : "",
              "label" : "Couvercle",
              "ui" : {
                  "access" : "r",
                  "display" : "warning",
                  "icon_color" : "#00ff00",
                  "icon_url" : "home_picto_cover_alert"
                },
              "value" : null,
              "value_type" : "bool"
            },
            {
              "ep_id" : 2,
              "label" : "Niveau de Batterie",
              "ui" : {
                  "access" : "r",
                  "display" : "warning",
                  "icon_color" : "#00ff00",
                  "icon_url" : "home_picto_battery_alert",
                  "range" : [
                      0,
                      10
                    ],
                  "unit" : "%"
                },
              "value" : null,
              "value_type" : "int"
            }
          ],
        "ep_type" : "signal",
        "group" : {
            "icon_url" : "http://lagabardine.ovh/~jeremie/img/favori.png",
            "label" : "alarm"
```

```
        },
        "label" : "Détecteur d'ouvertures",
        "node_id" : 24,
        "type" : "alarm_sensor"
      },
      {
        "data" : [
            {
              "ep_id" : 0,
              "history" : "1539597596899:1;1539867684806:1;1539868117300:0;1539868164089:1;1540282931546:1;1540296461125:0;1540296468385:1;",
              "label" : "Détection",
              "ui" : {
                  "access" : "r",
                  "display" : "icon",
                  "icon_color_range" : [
                      "#ff0000",
                      "#00ff00"
                    ],
                  "icon_url" : "home_picto_pir",
                  "status_text_range" : [
                      "Mouvement détecté",
                      "Aucun movement"
                    ],
                  "unit" : ""
                },
              "value" : null,
              "value_type" : "bool"
            },
            {
              "ep_id" : 1,
              "history" : "",
              "label" : "Couvercle",
              "ui" : {
                  "access" : "r",
                  "display" : "warning",
                  "icon_url" : "home_picto_cover_alert",
                  "unit" : ""
                },
              "value" : null,
              "value_type" : "bool"
            },
            {
              "ep_id" : 2,
              "label" : "Niveau de Batterie",
              "ui" : {
                  "access" : "r",
                  "display" : "warning",
                  "icon_url" : "home_picto_battery_alert",
                  "range" : [
                      0,
                      10
                    ],
                  "unit" : "%"
                },
              "value" : null,
              "value_type" : "int"
            }
          ],
        "ep_type" : "signal",
        "group" : {
            "icon_url" : "http://lagabardine.ovh/~jeremie/img/favori.png",
            "label" : "alarm"
          },
        "label" : "move",
        "node_id" : 26,
        "type" : "alarm_sensor"
      }
    ],
  "success" : true
}
```


List a Node sub-tileset

`GET /api/v8/home/tileset/{node_id}`

Get the list of all tiles corresponding to a node with “action”=”tileset”.

Example request:

```
GET /api/v8/home/tileset/42 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success" : true,
   "result" : [...]
}
```

Special Tiles specification

Alarm Tiles

Alarm control

This tile gives the current state of the alarm and allow to turn it on an off

type

alarm_control

data

Index	Value type	Access	Description
0	enum	r	The current alarm state
1	void	w	Activate the main alarm
2	void	w	Activate the night alarm
3	void	w	Deactivate the alarm
4	void	w	Skip the alarm activation timer
5	int	r/w	Alarm PIN code that should be asked before changing the alarm state
6	string	r	Alarm error code

state values

State	Description
idle	The alarm is off
alarm1_arming	The main alarm is being activated, it's a countdown when only the sensors not in the timed zone can trigger the alert
alarm2_arming	The night alarm is being activated, it's a countdown when only the sensors not in the timed zone can trigger the alert
alarm1_armed	The main alarm is on
alarm2_armed	The night alarm is on
alarm1_alert_timer	The main alarm has been triggered by a sensor in the timed zone and the siren will ring after a countdown
alarm2_alert_timer	The night alarm has been triggered by a sensor in the timed zone and the siren will ring after a countdown
alert	The siren is ringing

Alarm sensor
This tile represents a connected sensor used to trigger the alarm

type
alarm_sensor

data

Index	Value type	Access	Description
0	boolean	r	The state of this sensor: false=opening detected
1..n	*any*	r	Any data with *warning* display type

Alarm sensor
This tile represents a connected sensor used to trigger the alarm

type
alarm_sensor

data

Index	Value type	Access	Description
0	boolean	r	The state of this sensor: false=opening detected
1..n	*any*	r	Any data with *warning* display type

Camera
This tile represents a camera

type
camera

data

Index	Value type	Access	Description
0	string	r	The url of this camera on the local network

Automation tiles

Simple store
This tile represents a store with simple commands

type
info

action
store

data

Index	Value type	Access	Description
0	boolean	r	The state of the store: true=open, false=closed, null=undetermined
1	void	w	Command to open the store
2	void	w	Command to stop the store at its current position
3	void	w	Command to close the store

Commanded store
This tile represents a store with precise position command

type
info

action
store_slider

data

Index	Value type	Access	Description
0	int	rw	The position of store in percent: 0=fully opened, 100=fully closed
1	void	w	Command to stop the store at its current position

Color light bulb
This tile represents a connected light bulb with full color and intensity control

type
light

action
color_picker

data

Index	Value type	Access	Description
0	void	rw	The state of the light: true=on
1	int	rw	The H and S components of the color HSV value (H: 16 bits, S: 8 bit)
2	int	rw	The V value of the color HSV value (V: 8 bits)

White light bulb
This tile represents a connected light bulb with intensity and white tone control only

type
light

action
heat_picker

data

Index	Value type	Access	Description
0	void	rw	The state of the light: true=on

Index	Value type	Access	Description
1	int	rw	The H and S components of the color HSV value (H: 16 bits, S: 8 bit)
2	int	rw	The V value of the color HSV value (V: 8 bits)

Luminosity light bulb

This tile represents a connected light bulb with intensity control only

type

light

action

intensity_picker

data

Index	Value type	Access	Description
0	void	rw	The state of the light: true=on
1	int	rw	The luminosity value in percent

Cameras

The Camera API allows you to access features related to cameras.

Camera Errors

When attempting to access the Camera API, you may encounter the following errors:

error_code	Description
noent	no camera with this id
inval	invalid parameters

Camera object

Camera object have the following properties

`Camera`

`id` **string**
camera id

`node_id` **int**
camera node id

`name` **string**
camera name

`stream_url` **string**
camera stream url

`lan_gid` **string**
camera lan id

Camera API

Get list of cameras

`GET /api/v8/camera/`

Returns the collection of all `Camera`

Example request:

```
GET /api/v8/camera/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
      {
          "id": "012345678901",
          "node_id": 0,
          "name": "Caméra du salon",
          "stream_url": "/camera/stream/012345678901/stream.m3u8",
          "lan_gid": "ether-3c:98:72:fa:36:15"
      },
      {
          "id": "012345678902",
          "node_id": 1,
          "name": "Caméra du bureau",
          "stream_url": "/camera/stream/012345678902/stream.m3u8",
          "lan_gid": "ether-3c:98:72:fa:42:58"
      }
    ]
}
```


Access a given camera

`GET /api/v8/camera/{id}`

Returns the `Camera` with the given id

Example request:

```
GET /api/v8/camera/012345678901 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
       "id": "012345678901",
       "node_id": 0,
       "name": "Caméra du salon",
       "stream_url": "/camera/stream/012345678901/stream.m3u8",
       "lan_gid": "ether-3c:98:72:fa:36:15"
    }
}
```


Delete a camera

Use Home Node Api to delete camera (like a node) with its node id

Language

Language support

With this API you can fetch the list of supported languages on the Freebox, and change the current language.

Language support Object

`LanguageSupport`

`lang` **enum**

Currently configured language.

`avalaible[]` **array of string** ***Read-only***

List of supported languages, in iso 639-3 (alpha-3) format, used for changing the language.

Get language status

`GET /api/v8/lang/`

Get the current language in iso 639-3 (alpha-3) format, as well as the list of supported languages.

Example request:

```
GET /api/v8/lang HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
      "lang": "fra",
      "avalaible": [
         "fra",
         "eng"
      ]
    }
}
```

Set language

`POST /api/v8/lang/`

Set the current language.

Example request:

```
POST /api/v8/lang HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "lang": "eng"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
}
```

Notification

Notif

The Notification API allows you to access features related with notification,

Notification Errors

When attempting to access the Notification API, you may encounter the following errors:

error_code	Description
noent	no device with this id
inval	invalid parameters

Notification Target object

Target Notification Target object have the following properties

`NotificationTarget`

`id` **string**
device unique id

`last_use` **int**

`type` **string**
ios | android | firebase

`name` **string**
device name

`api_url` **string**
url of the notification server used to handle communication with the devices

`message_type` **string**
notification message type

Type	Description
data	only send the notification payload to the device
notification	send the notification payload along a notification title and body to the device

`subscriptions` **array**
permission list array

Type	Description
phone	notification when missing call
download	notification when download is finished
security	notification when alarm is on
box_state	notification when box state changed

Type	Description
lan_host	notification related to lan events
password_change	notification when admin password is changed

Notification API

Get list of notification target

`GET /api/v11/notif/targets`

Returns the collection of all `Notification Target`

Example request:

```
GET /api/v11/notif/targets HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success":true,
   "result":[
      {
         "last_use":0,
         "type":"ios",
         "name":"iPhone de Xavier",
         "id":"11111111-2222-3333-4444-555555555555",
         "subscriptions":[
            "security",
                              "downloader",
                              "phone",
         ],
         "api_url": "https://monserver.example.com/mon_app",
         "message_type": "notification"
      },
      {
         "last_use":0,
         "type":"android",
         "name":"mamy",
         "id":"22222222-1111-3333-4444-555555555555",
         "subscriptions":[
                              "phone"
         ],
         "api_url": "https://monserver.example.com/mon_app",
         "message_type": "notification"
   ]
}
```

Get a given notification target by this id

`GET /api/v11/notif/targets/{id}`

Returns the `Notification Target` with the given id

Example request:

```
GET /api/v11/notif/targets/11111111-2222-3333-4444-555555555555 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success":true,
   "result":[
      {
         "last_use":0,
         "type":"ios",
         "name":"iPhone de Xavier",
         "id":"11111111-2222-3333-4444-555555555555",
         "subscriptions":[
            "security",
                              "downloader",
                              "phone",
         ],
         "api_url": "https://monserver.example.com/mon_app",
         "message_type": "notification"
      }
   ]
}
```

Delete a notification target

`DELETE /api/v11/notif/targets/{id}`

Deletes the `Notification Target` with the given id.

Example request:

```
DELETE /api/v11/notif/targets/22222222-1111-3333-4444-555555555555 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Update a notification target

`PUT /api/v11/notif/targets/{id}`

Update the `Notification Target` with the given id.

Example request:

```
PUT /api/v11/notif/targets/22222222-1111-3333-4444-555555555555 HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
              "name": "iPhone de Xavier",
              "type": "ios",
              "token": "token_token_token_token_token_token_token",
              "subscriptions": ["download", "phone"],
```

```
                "api_url": "https://monserver.example.com/mon_app",
   "message_type": "notification"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Add a notification target

`POST /api/v11/notif/targets/`

Create an new `Notification Target`.

Example request:

```
POST /api/v11/notif/targets/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
                "name": "iPhone de Xavier",
                "type": "ios",
                "token": "token_token_token_token_token_token_token",
                "subscriptions": ["download", "phone"],
                "api_url": "https://monserver.example.com/mon_app",
   "message_type": "notification"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Notification server specification

When a notification should be sent, the Freebox will use this API on the address specified in the notification target. Your server must implement this API contract :

`POST /register`

A new target has been registered

```
{
  "box_id":"", //uuid of the box that is sending the request
  "device_type":"ios|android|firebase", //notification service type of the target
  "token":"", //the notification service token
  "device_name":"",
  "device_id":"" //the target id
}
```

`DELETE /register/{box_id}/{device_id}`

A target has been deleted

`POST /send`

Send a notification

```
{
  "devices":["", "", "", ...], //an array of target id
  "title":"", //notification title (optional - only sent if target message type is "notification")
  "body":"", //notification body (optional - only sent if target message type is "notification")
  "payload": {}, //json payload to send as notification data
  "box_id":"" //uuid of the box that is sending the request
}
```

Response : This API send back the device ids in two lists : failure and success

```
{
  "failureIds": ["device_id_1", "device_id_2", ...],
  "successIds": ["device_id_3", ...]
}
```

Notifications specification

Notifications sent to registered devices has a payload depending on notification type :

`downloader`

`box_id` **string**
ID of the box that sent the notification

`type` **string**
Notification type : downloader

`data` **int**
ID of the download task that triggered the notification

`event` **enum**
Downloader event that triggered the notification

event	Description
task_done	The download task is complete
task_error	The download task has failed
task_seeding_done	The download task seeding is complete

`phone`

`box_id` **string**
ID of the box that sent the notification

`type` **string**
Notification type : phone

`data` **CallEntry**
Call object that triggered the notification

`event` **enum**
Phone event that triggered the notification

event	Description
missed_call	A call has been missed

`box_state`

box_id string
ID of the box that sent the notification

type string
Notification type : box_state

event enum
Box state event that triggered the notification

event	Description
pub_up	Wan public connection went up
enter_sleep	Box will enter sleep mode
shut_down	Box will shut down
reboot	Box will reboot

lan_host

box_id string
ID of the box that sent the notification

type string
Notification type : lan

host_id string
ID of the host that triggered the notification

interface string
The LAN interface the host is connected to

event enum
LAN host event that triggered the notification

event	Description
first_connection	The device is connected for the first time

password_change

box_id string
ID of the box that sent the notification

type string
Notification type : password_change

ip string
IP of the lan host that requested password change

Parental filter

Profile management

Profile Object

Profile

id int ***Read-only***

unique id of this profile

name string

name of this profile

icon string

URL of the icon relative to root of the API domain.

Profiles API

Get the list of profiles

GET /api/v8/profile

Example request:

```
GET /api/v8/profile HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
      {
          "id": 2,
          "name": "r0ro",
          "url": "/resources/images/profile/profile_04.png"
      },

        [ ... ]

      {
          "id": 7,
          "name": "Xav",
          "url": "/resources/images/profile/profile_02.png"
      }
    ]
}
```

Get a profile

GET /api/v8/profile/{id}

Get the Profile with the given id

Example request:

```
GET /api/v8/profile/2 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
```

```
        "id": 2,
        "name": "r0ro",
        "url": "/resources/images/profile/profile_04.png"
    }
}
```

Add a profile

`POST /api/v8/profile/`

Example request:

```
POST /api/v8/profile HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "name": "Pierrot",
   "url": "/resources/images/profile/profile_04.png"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result":
       {
          "id": 3,
       }
}
```

Delete a profile

`DELETE /api/v8/profile/{id}`

Example request:

```
DELETE /api/v8/profile/2 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
}
```

Update a profile

`PUT /api/v8/profile/3`

Example request:

```
PUT /api/v8/profile HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "name": "Pierrot",
   "url": "/resources/images/profile/profile_02.png"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result":
       {
          "id": 3,
          "name": "Pierrot",
          "url": "/resources/images/profile/profile_02.png"
       }
}
```

Network Control Object

The different modes supported are :

mode	Description
allowed	access is allowed
denied	access is denied
webonly	access is granted only for HTTP and HTTPS traffic; legacy mode, use not recommended.

`NetworkControl`

`profile_id` **int** ***Read-only***

Id of the profile this network control is associated with. This is read-only, unless you use the POST api to add a network control.

`next_change` **int** ***Read-only***

UNIX timestamp of next rule change in seconds. 0 if no next change.

`override_mode` **enum**

mode of current override.

`current_mode` **enum** ***Read-only***

mode in use. If override is true, it will be override_mode, otherwise it's the mode from the rules attached to this NetworkControl.

`rule_mode` **enum** ***Read-only***

mode that would be in use if there was no override. Depends only on rules, and is useful to determine what will happen when override is lifted.

`override_until` **int**

Unix timestamp in seconds when override ends. Relevant when override is true. Set at 0 for unlimited.

`override` **bool**

Whether there's an override at the moment.

`macs[]` **array of string**

List of mac adresses associated with this profile's network control.

`hosts[]` **array of** **LanHost** ***Read-only***

List of Lan Host objects associated with this profile's network control. Derived from the macs array.

`resolution` **int** ***Read-only***

Control resolution per day of this network control. Currently at 288.

cdayranges[] array of string

list of custom day range, each custom day range represents a group of days for which you want to use a different planning than other week days.

For instance a custom day range can contain the list of your children holidays.

cdayranges	Description
:fr_bank_holidays	French bank holidays
:fr_school_holidays_a	French school holidays - Zone A
:fr_school_holidays_b	French school holidays - Zone B
:fr_school_holidays_c	French school holidays - Zone C
:fr_school_holidays_corse	French school holidays - Corse

each cdayranges can be a coma separated list of cdayranges, for instance ":fr_bank_holidays,:fr_school_holidays_b"

Network Control API

Get Network Control for all profiles

`GET /api/v8/network_control`

Get Network Control for a profile

`GET /api/v8/network_control/{profile_id}`

Example request:

```
GET /api/v8/network_control/5 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result":
     {
         "profile_id": 5,
         "next_change": 0,
         "override": false,
         "override_mode": "denied",
         "current_mode": "allowed",
         "macs": [
             "D8:A2:CA:FE:BA:DF",
             "D0:23:BE:DE:AD:EF"
         ],
         "hosts": [
            "PC-de-mamie",
            "Cantal-chromebook"
         ],
         "resolution": 288,
         "cdayranges": []
     }
}
```


Update Network Control for a profile

`PUT /api/v8/network_control/{profile_id}`

Example request:

```
PUT /api/v8/network_control/3 HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
    "profile_id": 3,
    "next_change": 0,
    "override": false,
    "override_mode": "allowed",
    "current_mode": "denied",
    "macs": [
        "98:E8:FA:FE:BA:42",
        "2C:CC:44:D1:AD:4F"
    ],
    "hosts": [
       "3DS-Thibault",
       "Vita-Rodolphe"
    ],
    "resolution": 288,
    "cdayranges": []
}
```


Example response:


```
{
    "success": true,
    "result":
     {
         "profile_id": 3,
         "next_change": 0,
         "override": false,
         "override_mode": "allowed",
         "current_mode": "denied",
         "macs": [
             "98:E8:FA:FE:BA:42",
             "2C:CC:44:D1:AD:4F"
         ],
         "hosts": [
            "3DS-Thibault",
            "Vita-Rodolphe"
         ],
         "resolution": 288,
         "cdayranges": []
     }
}
```


Get migration to new default mode status

Verify if migration to new default mode has been done ("allowed" only) if default mode was modified.

`GET /api/v8/network_control/migrate`

Example request:

```
GET /api/v8/network_control/migrate HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result":
     {
         "default_mode_migrated": false
     }
}
```

Migrate to new default mode

Do migration to new default mode (“allowed”) if it was modified previously.

`POST /api/v8/network_control/migrate`

Example request:

```
POST /api/v8/network_control/migrate HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result":
     {
         "default_mode_migrated": true
     }
}
```

Rule Object

`NetworkControlRule`

`id` int ***Read-only***

Unique rule identifier.

`profile_id` int ***Read-only***

Id of profile this rule applies to.

`name` string

Rule name

`mode` enum

Mode described in Network Control Object

`start_time`

Seconds since start of day (00:00) when rule starts. Must be in increments of the resolution. When resolution is 288, it means 5 minutes slots, so the value must be a multiple of 300.

`end_time`

Time of day in seconds since start of day (00:00) when rule ends. end_time modulo 300 must always be zero when resolution is 288.

`weekdays[]` array of bool

Array of days of weeks when this rule apply. 8th one is for cdayranges.

`enabled` bool

Whether rule is enabled.

Rule API

Get Network Control Rules for a profile

`GET /api/v8/network_control/{profile_id}/rules`

Returns the list of rules for this profile

Get a Network Control Rule

`GET /api/v8/network_control/{profile_id}/rules/{rule_id}`

Returns one rule.

Create a Network Control Rule

`POST /api/v8/network_controlr/{profile_id}/rules/`

Create a rule given in parameter.

Update a Network Control Rule

`PUT /api/v8/network_control/{id}/rules/{rule_id}`

Update rule.

Delete a Network Control Rule

`DELETE /api/v8/network_control/{id}/rules/{rule_id}`

Delete rule.

Player devices

Player [UNSTABLE]

* INTERNAL USE ONLY *

With the player API you access and control a Freebox Player connected on the same local network as the Freebox Server. Available players can be enumerated, and the listed player identifier can be used to dispatch commands.

Player Errors

When attempting to access the player API, you may encounter the following errors:

error_code	Description
internal_error	Internal error
inval	Invalid parameters
noent	no player with this id

Player Objects

Player

`Player`

`id` int

`device_name` string

uid string

reachable bool

api_version string

api_available bool

Player Status Foreground App

PlayerStatusForegroundApp

package_id id

cur_url string

context object

package string

Player Status Capabilities

Capabilities of a media player.

PlayerStatusCapabilities

play bool

pause bool

stop bool

next bool

prev bool

record bool

record_stop bool

seek_forward bool

seek_backward bool

seek_to bool

shuffle bool

repeat_all bool

repeat_one bool

select_stream bool

select_audio_track bool

select_srt_track bool

Player Status Informations

PlayerStatusInformations

name string

last_activity long

capabilities PlayerStatusCapabilities

Player Status

PlayerStatus

power_state string

state	Description
standby	freebox player is currently in standby mode
running	freebox player is on

player PlayerStatusInformations

State of the active media player on the device.

foreground_app PlayerStatusForegroundApp

The context of the currently running application. The fields exposed in this object are left to the discretion of the application author, and thus subject to change at any time.

Player API

List every player devices

GET /api/v8/player

Returns the list of all player devices registered on the local network ([Player]).

Example request:

```
GET /api/v8/player HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true,
   "result": [
      {
         "device_name": "Freebox Player",
         "stb_type": "stb_v7",
         "uid": "12345678901234567891123456789212 3",
         "reachable": true,
         "api_version": "6.0",
         "id": 11,
         "api_available": true
      }
   ]
}
```

Get player device status

GET /api/v8/player/{id_player}/api/v6/status/

Returns the current state of a player device (Player).

Example request:

```
GET /api/v8/player/11/api/v6/status/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true,
   "result": {
      "power_state": "standby"
   }
}
```

Control the active media player of a device

`POST /api/v8/player/{id_player}/api/v6/control/mediactrl/`

Parameters

- **cmd** (*string*) – Command to execute

Send a command to the active media player of a device. Not all commands are always available, the capabilities of the active media player can be retrieved in the device status to determine which commands ca be used.

command	Description
play_pause	toggle play pause
stop	stop
prev	previous
next	next
select_stream	select quality of the stream
select_audio_track	select audio track
select_srt_track	select subtitle track

Example request:

```
POST /api/v8/player/11/api/v6/control/mediactrl/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "cmd": "play_pause"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true
}
```

Control the playback volume of the device

`GET /api/v8/player/{id_player}/api/v6/control/volume/`

Example request:

```
GET /api/v8/player/11/api/v6/control/volume/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true,
   "result": {
      "mute": false
      "volume": 25
   }
}
```

`PUT /api/v8/player/{id_player}/api/v6/control/volume/`

Parameters

- **volume** (*integer*) – Master volume from 0 to 100
- **mute** (*boolean*) – Mute

Example request:

```
PUT /api/v8/player/{id_player}/api/v6/control/volume/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "volume": 50
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
   "success": true,
   "result": {
      "mute": false
      "volume": 50
   }
}
```

Open a url on a player device

`POST /api/v8/player/{id_player}/api/v6/control/open`

Parameters

- **url** (*string*) – Url to open on the Freebox Player
- **type** (*string*) – Mime type of the content to open on the Freebox Player (optional: default is empty)

Here are some useful examples calls:

Open the video player:

```
{ "url": "http://jell.yfish.us/media/jellyfish-3-mbps-hd-h264.mkv",
  "type": "video/x-matroska" }
```

Open the web browser:

```
{ "url": "https://www.google.com",
  "type": "text/html" }
```

Open TV on channel 2:

```
{ "url": "tv:?channel=2" }
```

Open a YouTube video:

```
{ "url": "https://www.youtube.com/watch?v=pltY5vS-aOY" }
```

Example request:

```
POST /api/v8/player/11/api/v6/control/open HTTP/1.1
Host: mafreebox.freebox.fr

{
   "url": "tv:?channel=123"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
   "success": true
}
```

PVR

PVR [UNSTABLE]

* INTERNAL USE ONLY *

PVR Errors

error_code	Description
noent	wrong id
inval	invalid params
inval_date_fmt	invalid date format
inval_end_before_start	start time must be before end time
system_time_incorrect	system time not available
record_duration_too_long	record duration is too long
record_date_in_past	record date is already passed
unknown_channel	unknown channel
no_channel_svc	no service for this channel
only_auto_disable	can't disable manual precord
cannot_change_en_state	can't change enabled state
cannot_disable_has_data	can't disable started record
internal_error	internal error

PVR Config

PVR config has the following attributes:

`PvrConfig`

- `margin_before` `int`
  default margin before recording start time
- `margin_after` `int`
  default margin after recording end time

PVR Config API

Get the current PVR configuration

`GET /api/v8/pvr/config/`

Returns the current PvrConfig

Example request:

```
GET /api/v8/pvr/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
    "success": true,
    "result": {
        "margin_before": 10,
        "margin_after": 5
    }
}
```

Update the current PVR configuration

`PUT /api/v8/pvr/config/`

Update the current PvrConfig

PVR Quota

PVR Quota has the following attributes:

`PvrQuota`

- `quota_exceeded` `bool`
  is quota exceeded
- `needed_tresh` `int`
  needed quota threshold
- `cur_tresh` `int`
  current quota threshold

PVR Quota API

Getting the current quota info

`GET /api/v8/pvr/quota/`

Example request:

```
GET /api/v8/pvr/quota/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "quota_exceeded": true,
        "needed_tresh": 80,
        "cur_tresh": 40
    }
}
```

Request next quota threshold

`PUT /api/v8/pvr/quota/`

Request next quota threshold. You don't have to provide any arguments, the quota will be adjusted automatically if needed.

Example request:

```
PUT /api/v8/pvr/quota/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{ }
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "quota_exceeded": false,
        "needed_tresh": 80,
        "cur_tresh": 80
    }
}
```

PVR Programmed records

Precords (Programmed records) are records that are planned. Precords can be manual, or generated using a PVR Generator (see below). Only manual Precords can be edited directly.

Precord

Precord has the following attributes:

`Precord`

`id` **string** ***Read-only***
: precord id

`media` **string**
: media name on which the record will be written to. See the Media API for more info. This property and can be empty when the file backing the record is not available, for example when secure is set.

`path` **string**
: destination directory on the media storage where the record will be written to

`has_record_gen` **bool** ***Read-only***
: if true, this precord has been generated using a Generator

`record_gen_id` **int** ***Read-only***
: if has_record_gen, this is the id of the generator

`conflict` **bool** ***Read-only***
: if true this record may conflict with another record

`overlap_list[]` **array of int** ***Read-only***
: in case of conflict, this will contain the list of records id that may conflict with this record

`enabled` **bool**
: it only applies to generated records. If false the generated precord will be skipped.

`altered` **bool** ***Read-only***
: a precord is altered when some part of the recording may be missing. This can be the case if a conflict occurred during the recording (or connection was down)

`state` **enum** ***Read-only***

State	Description
disabled	disabled
start_error	failed to start
waiting_start_time	scheduled
starting	starting
running	running
running_error	running with error
failed	failed
finished	finished

`error` **enum** ***Read-only***

Error	
none	
file_access_error	
disk_full	
private_but_no_private_dir	
network_problem	
resource_problem	
no_stream_available	
no_data_received	
missed	
stopped	

Error	
internal_error	
unknown_error	

`channel_uuid` **string**
channel uuid

`channel_name` **string**
optional channel name

`channel_quality` **enum**

channel_quality	
auto	
hd	
sd	
ld	
3d	

`channel_type` **enum**

channel_type	Description
'' (empty string)	auto
iptv	use only iptv streams
dvb	use only dvb streams

`name` **string**
record name

`subname` **string**
record subname

`broadcast_type` **enum**

broadcast_type	
tv	
radio	

`start` **int**
record start timestamp

`end` **int**
record end timestamp

`legacy_uri` **string**
only used for legacy apps. Use channel_uuid instead when available NOTE: only visible when called from player

`force_channel_name` **string**
only used for legacy apps. Use channel_uuid instead when available NOTE: only visible when called from player

Precord API

Getting the list of precords

`GET /api/v8/pvr/programmed/`

Example request:

```
GET /api/v8/pvr/programmed/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "has_record_gen": true,
            "channel_name": "France 2",
            "overlap_list": [
                195
            ],
            "end": 1403755697,
            "media": "Disque dur",
            "path": "Enregistrements",
            "record_gen_id": 10,
            "enabled": true,
            "id": 190,
            "start": 1403755628,
            "broadcast_type": "tv",
            "subname": "",
            "state": "waiting_start_time",
            "channel_type": "",
            "name": "Test Repeat",
            "channel_quality": "auto",
            "conflict": true,
            "channel_uuid": "uuid-webtv-201",
            "error": "none",
            "altered": false
        }

        [ ... ]

        {
            "has_record_gen": false,
            "channel_name": "France 2",
            "overlap_list": [ ],
            "end": 1403541511,
            "media": "NO NAME",
            "path": "Enregistrements",
            "record_gen_id": 0,
            "enabled": true,
            "id": 236,
            "start": 1403541361,
            "broadcast_type": "tv",
            "subname": "Sub Test",
            "state": "finished",
            "channel_type": "iptv",
            "name": "Test",
```

```
            "channel_quality": "auto",
            "conflict": false,
            "channel_uuid": "uuid-webtv-201",
            "error": "none",
            "altered": true
        }
    ]
}
```

Getting a specific precord

`GET /api/v8/pvr/programmed/{id}`

Returns the requested Precord

Example request:

```
GET /api/v8/pvr/programmed/236 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "has_record_gen": false,
        "channel_name": "France 2",
        "overlap_list": [ ],
        "end": 1403541511,
        "media": "NO NAME",
        "path": "Enregistrements",
        "record_gen_id": 0,
        "enabled": true,
        "id": 236,
        "start": 1403541361,
        "broadcast_type": "tv",
        "subname": "Sub Test",
        "state": "finished",
        "channel_type": "iptv",
        "name": "Test",
        "channel_quality": "auto",
        "conflict": false,
        "channel_uuid": "uuid-webtv-201",
        "error": "none",
        "altered": true
    }
}
```

Updating a precord

`PUT /api/v8/pvr/programmed/{id}`

Update a Precord properties

Example request:

```
PUT /api/v8/pvr/programmed/236 HTTP/1.1
Host: mafreebox.freebox.fr

{
  "name": "test 2"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "has_record_gen": false,
        "channel_name": "France 2",
        "overlap_list": [ ],
        "end": 1403541511,
        "media": "NO NAME",
        "path": "Enregistrements",
        "record_gen_id": 0,
        "enabled": true,
        "id": 236,
        "start": 1403541361,
        "broadcast_type": "tv",
        "subname": "Sub Test",
        "state": "finished",
        "channel_type": "iptv",
        "name": "test 2",
        "channel_quality": "auto",
        "conflict": false,
        "channel_uuid": "uuid-webtv-201",
        "error": "none",
        "altered": true
    }
}
```

Delete a precord

`DELETE /api/v8/pvr/programmed/{id}`

Delete a Precord

Example request:

```
DELETE /api/v8/pvr/programmed/236 HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
}
```

Create a precord

`POST /api/v8/pvr/programmed/`

Create a new Precord

** Example request**:

```
POST /api/v8/pvr/programmed/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
    "start": 1444240500,
    "end": 1444244100,
    "channel_uuid": "uuid-webtv-374",
    "name": "Secret Story",
    "subname: "La soirée des habitants"
}
```

** Example response**:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "id": 63,
        "media": "Disque dur",
        "path": "Enregistrements",
        "channel_uuid": "uuid-webtv-374",
        "channel_name": "NT1",
        "channel_type": "",
        "channel_quality": "auto",
        "broadcast_type": "tv",
        "start": 1444240500,
        "end": 1444244100,
        "name": "Secret Story",
        "subname": "La soirée des habitants",
        "state": "starting",
        "error": "none",
        "enabled": true,
        "altered": false,
        "conflict": false,
        "overlap_list": [],
        "margin_before": 0,
        "margin_after": 0,
        "has_record_gen": false,
        "record_gen_id": 0
    }
}
```


PVR Finished records

Frecords (Finished records) are records that are finished or in progress. An *Frecord* object is created automatically when a *Precord* start time is reached.

Frecord

Frecord has the following attributes:

`Frecord`

`id` `string` ***Read-only***
: frecord id

`media` `string` ***Read-only***
: media name on which the record is written. See the Media API for more info. This property and can be empty when the file backing the record is not available, for example when secure is set.

`path` `string` ***Read-only***
: destination directory on the media storage

`filename` `string` ***Read-only***
: filename of the record

`byte_size` `int` ***Read-only***
: size of the record file in bytes

`has_record_gen` `bool` ***Read-only***
: if true, this frecord has been generated using a Generator

`record_gen_id` `int` ***Read-only***
: if has_record_gen, this is the id of the generator

`altered` `bool` ***Read-only***
: an frecord is altered when some part of the recording may be missing. This can be the case if a conflict occurred during the recording (or connection was down)

`state` `enum` ***Read-only***

State	Description
disabled	disabled
start_error	failed to start
waiting_start_time	scheduled
starting	starting
running	running
running_error	running with error
failed	failed
finished	finished

`error` `enum` ***Read-only***

Error	
none	
file_access_error	
disk_full	
private_but_no_private_dir	
network_problem	
resource_problem	
no_stream_available	
no_data_received	
missed	
stopped	
internal_error	
unknown_error	

`channel_uuid` `string` ***Read-only***

channel uuid

`channel_name` `string` ***Read-only***

optional channel name

`channel_quality` `enum` ***Read-only***

channel_quality	
auto	
hd	
sd	
ld	
3d	

`channel_type` `enum` ***Read-only***

channel_type	Description
'' (empty string)	auto
iptv	use only iptv streams
dvb	use only dvb streams

`name` `string`

record name

`subname` `string`

record subname

`broadcast_type` `enum` ***Read-only***

broadcast_type	
tv	
radio	

`start` `int` ***Read-only***

record start timestamp

`end` `int` ***Read-only***

record end timestamp

`secure` `bool` ***Read-only***

flag set when the record is protected by DRM

Frecord API

Getting the list of frecords

`GET /api/v8/pvr/finished/`

Example request:

```
GET /api/v8/pvr/finished/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "id": 5,
            "media": "Disque dur",
            "path": "Enregistrements",
            "filename": "M6 - Fier de ma maison - 27-06-2013 16h35 01h15 (5).m2ts",
            "byte_size": 4433869440,
            "has_record_gen": false,
            "record_gen_id": 0,
            "broadcast_type": "tv",
            "channel_uuid": "uuid-webtv-613",
            "channel_name": "M6",
            "channel_type": "dvb",
            "channel_quality": "hd",
            "name": "Fier de ma maison",
            "subname": "",
            "start": 1372343700,
            "end": 1372348200,
            "state": "finished",
            "error": "none",
            "enabled": true,
            "altered": true,
            "secure": false
        },

        [ ... ]

        {
            "id": 22,
            "media": "",
            "path": "",
            "filename": "TF1 - Nos chers voisins - 17-09-2014 15h23 01h (22).m2ts",
            "byte_size": 2421095040,
            "has_record_gen": false,
            "record_gen_id": 0,
            "broadcast_type": "tv",
            "channel_uuid": "uuid-webtv-612",
            "channel_name": "TF1",
            "channel_type": "",
            "channel_quality": "auto",
            "name": "Nos chers voisins",
            "subname": "",
            "start": 1410960180,
            "end": 1410963780,
            "state": "finished",
            "error": "none",
            "enabled": true,
            "altered": true,
            "secure": true
        }
    ]
}
```

Getting a specific frecord

`GET /api/v8/pvr/finished/{id}`

Returns the requested Frecord

Example request:

```
GET /api/v8/pvr/finished/236 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "id": 236,
        "media": "NO NAME",
        "path": "",
        "filename": "France 3 - Tout le sport - 10-04-2015 20h00 10m (24).m2ts",
        "byte_size": 341752320,
        "has_record_gen": false,
        "record_gen_id": 0,
        "broadcast_type": "tv",
        "channel_uuid": "uuid-webtv-202",
        "channel_name": "France 3",
        "channel_type": "",
        "channel_quality": "auto",
        "name": "Tout le sport",
        "subname": "",
        "start": 1428688800,
        "end": 1428689400,
        "state": "finished",
        "error": "none",
        "enabled": true,
        "altered": true,
        "secure": false
    }
}
```


Updating an frecord

`PUT /api/v8/pvr/finished/{id}`

Update a Frecord properties

Example request:

```
PUT /api/v8/pvr/finished/236 HTTP/1.1
Host: mafreebox.freebox.fr

{
  "name": "Tout le sport",
  "subname": "On est les champions"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "id": 236,
        "media": "NO NAME",
        "path": "",
        "filename": "France 3 - Tout le sport - 10-04-2015 20h00 10m (24).m2ts",
        "byte_size": 341752320,
        "has_record_gen": false,
        "record_gen_id": 0,
        "broadcast_type": "tv",
        "channel_uuid": "uuid-webtv-202",
        "channel_name": "France 3",
        "channel_type": "",
        "channel_quality": "auto",
        "name": "Tout le sport",
        "subname": "On est les champions",
        "start": 1428688800,
        "end": 1428689400,
        "state": "finished",
        "error": "none",
        "enabled": true,
        "altered": true,
        "secure": false
    }
}
```


Delete an frecord

`DELETE /api/v8/pvr/finished/{id}`

Delete a Frecord and associated files

Example request:

```
DELETE /api/v8/pvr/finished/236 HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
}
```


Storage media

Media objects represent a storage on which records can be written to, typically a disk.

Media

Media has the following attributes:

`Media`

`media` `string` ***Read-only***
name of the storage medium

`free_bytes` int ***Read-only***

number of free bytes on the medium

`total bytes int [ro]`

total number of bytes on the medium

`record_time` int ***Read-only***

estimated record time in seconds for multiple channel types and qualities

Media API

Getting the list of media

`GET /api/v8/pvr/media/`

Example request:

```
GET /api/v8/pvr/media/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
        {
            "media": "Disque dur",
            "free_bytes": 39700000000,
            "total_bytes": 244950000000,
            "record_time": {
                "dvb":  { "sd": 48461, "hd": 35245, "3d": 35245 },
                "iptv":  { "ld": 155078, "sd": 110770, "hd": 51012, "3d": 51012 }
            }
        },

        [ ... ]

        {
            "media":  "NO NAME",
            "free_bytes": 873930000,
            "total_bytes":  7790000000,
            "record_time":  {
                "dvb":  { "sd": 1066, "hd": 775, "3d": 775 },
                "iptv":  { "ld": 3413, "sd": 2438, "hd": 1122, "3d": 1122 }
            }
        }
    ]
}
```

RRD

RRD [UNSTABLE]

With the rrd API you can retrieve stats collected on the Freebox. Right now the stats available are: network stats, switch stats, dsl stats, and temperature stats.

RRD Fetch Object

This is the object used to get stats

`RRDFetch`

`db enum`

Name of the rrd database to read. It can take one of the following values

Db	Description
net	network stats
temp	temperature stats
dsl	xDSL stats
switch	switch stats

`date_start` int ***Optionnal***

The requested start timestamp of the stats to get

NOTE: this can be adjusted to fit the best available resolution

`date_end` int ***Optionnal***

The requested end timestamp of the stats to get

NOTE: this can be adjusted to fit the best available resolution

`precision` int ***Optionnal***

By default all values are cast to int, if you need floating point precision you can provide a precision factor that will be applied to all values before being returned.

For instance if you want 2 digit precision you should use a precision of 100, and divide the obtained results by 100.

`fields[] array of string` ***Optionnal***

If you are only interested in getting some fields you can provide the list of fields you want to get.

For the net database the fields are:

Field	Description
bw_up	upload available bandwidth (in byte/s)
bw_down	download available bandwidth (in byte/s)
rate_up	upload rate (in byte/s)
rate_down	download rate (in byte/s)
vpn_rate_up	vpn client upload rate (in byte/s)
vpn_rate_down	vpn client download rate (in byte/s)

For the temp database the fields are:

Field	Description
cpum	temperature cpum (in °C)
cpub	temperature cpub (in °C)
sw	temperature sw (in °C)
hdd	temperature hdd (in °C)
fan_speed	fan rpm
temp1	temperature sensor 1 (in °C) [DEPRECATED, use cpum]

Field	Description
temp2	temperature sensor 2 (in °C) [DEPRECATED, use cpub]
temp3	temperature sensor 3 (in °C) [DEPRECATED, use sw]

For the dsl database the fields are:

Field	Description
rate_up	dsl available upload bandwidth (in byte/s)
rate_down	dsl available download bandwidth (in byte/s)
snr_up	dsl upload signal/noise ratio (in 1/10 dB)
snr_down	dsl download signal/noise ratio (in 1/10 dB)

For the switch database the fields are:

Field	Description
rx_1	receive rate on port 1 (in byte/s)
tx_1	transmit on port 1 (in byte/s)
rx_2	receive rate on port 2 (in byte/s)
tx_2	transmit on port 2 (in byte/s)
rx_3	receive rate on port 3 (in byte/s)
tx_3	transmit on port 3 (in byte/s)
rx_4	receive rate on port 4 (in byte/s)
tx_4	transmit on port 4 (in byte/s)

Get RRD stats [UNSTABLE]

`POST /api/v8/rrd/`

Example request:

```
POST /api/v8/rrd/ HTTP/1.1
Host: mafreebox.freebox.fr

{
   "db": "temp",
   "fields": [ "temp1" ],
   "precision": 10
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "date_start": 1353048060,
        "data": [
            {
                "temp1": 540,
                "time": 1353060840
            },
            {
                "temp1": 545,
                "time": 1353060900
            },

            [ ... ],

            {
                "temp1": 540,
                "time": 1353069600
            }
        ],
        "date_end": 1353069660
    }
}
```

`GET /api/v8/rrd/`

Same as post request, but allowed without 'settings' permission

Standby

Standby

The Standby API allows you to configure Wi-Fi schedule. On boxes that have has_standby set to true in their SystemConfig information, it is possible to configure box standby and wake-up.

Standby Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
inval	invalid parameters

Standby config object

Standby config object have the following properties:

`StandbyConfig`

`use_planning` **bool**

is the planning enabled

`planning_mode` **enum**

current planning mode

Type	Description
wifi_off	Wi-Fi disabled
standby	Freebox standby

`resolution` **int** ***Read-only***

planning resolution (number of slots per day)

`mapping[]` **array of bool**

mapping for planning : true or false

mapping[0] is monday at 0:0

mapping[7 * resolution - 1] is sunday last slot

(each slot has a duration of 60 * 24 / resolution minutes)

The boolean value indicates whether the planning is in effect (i.e: Wi-Fi disabled, or box standing by)

Standby status object

Standby status object have the following properties:

`StandbyStatus`

`use_planning` `bool` ***Read-only***

is the planning enabled

`planning_mode` `enum` ***Read-only***

Type of planning that is configured, just like in `StandbyConfig`

`next_change` `timestamp` ***Read-only***

timestamp of the scheduled next change, according to planning

`available_planning_modes` `array` ***Read-only***

array of available planning modes. Individual array elements are enum values just like *planning_mode* in `StandbyConfig`

Standby API

Get standby status

`GET /api/v11/standby/status`

Returns the `Standby status object`

Example request:

```
GET /api/v11/standby/status HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
    "use_planning": true,
    "planning_mode": "standby",
    "next_change": 1651135474996,
    "available_planning_modes": [ "wifi_off", "standby" ]
  }
}
```

Get standby config

Get the `StandbyConfig`

Example request:

```
GET /api/v11/standby/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
    "use_planning": false,
    "planning_mode": "suspend",
    "mapping": [
      false,
      false,
      false,
      false,

      [ ... ]

      false,
      false,
      false,
      false
    ],
    "resolution": 48
  }
}
```

Update standby config

`PUT /api/v11/standby/config`

Example request:

```
PUT /api/v11/standby/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "use_planning": true,
  "planning_mode": "suspend",
  "mapping": [
    false,
    false,
    false,
    false,

    [ ... ],

    false,
    false,
    false,
    false
  ],
  "resolution": 48
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
```

```
  "result": {
    "use_planning": false,
    "planning_mode": "suspend",
    "mapping": [
      false,
      false,
      false,
      false,
      false,

      [ ... ]

      false,
      false,
      false,
      false
    ],
    "resolution": 48
  }
}
```


Storage

Storage API [UNSTABLE]

This API allows you to manage the Freebox internal disk and disks connected to the Freebox

This API is unstable, it can be modified without notice in next releases.

Storage API Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
not_found	No disk/partition with this id
invalid_disk	No such disk
is_a_partition	This is not a disk but a partition
is_internal	This action is not permitted on internal disk
op_not_supported	Operation not supported
op_failed	Operation failed
disk_busy	Disk is busy
partition_not_found	Partition not found
partition_needed	Partition needed

Disk Partition object

Operation progress has the following attributes:

`OperationProgress`

`done_steps` `int` ***Read-only***
number of steps done

`max_steps` `int` ***Read-only***
total number of steps

`percent` `int` ***Read-only***
current step progress

Disk partitions have the following attributes:

`DiskPartition`

`id` `int` ***Read-only***
unique partition id

`disk_id` `int` ***Read-only***
related disk id

`state` `enum`

state	Description
error	Partition has error
checking	Partition check in progress
formatting	Partition format in progress
mounting	Partition mount in progress
maintenance	Partition is in maintenance mode
mounted	Partition is ready
umounting	Partition umount in progress
umounted	Partition is umounted
ejecting	Partition ejection in progress

`fstype` `enum` ***Read-only***

fstype	
empty	
unknown	
xfs	
ext4	
vfat	
ntfs	
hf	
hfsplus	
swap	
exfat	

`label` `string`
partition name

`path` `string` ***Read-only***

partition mount point (encoded in base64 as explained in fs API)

`total_bytes` int ***Read-only***
partition size (in bytes)

`used_bytes` int ***Read-only***
partition used space (in bytes)

`free_bytes` int ***Read-only***
partition free space (in bytes)

`fsck_result` enum ***Read-only***
fsck result

state	Description
no_run_yet	Partition has not been checked yet
running	Check is in progress
fs_clean	File system is ok
fs_corrected	File system was corrected
fs_needs_correction	File system need correction
failed	File system has unrecoverable error

`operation_pct` OperationProgress ***Read-only***
partition operation progress

Storage Disk object

Storage disks have the following attributes:

`StorageDisk`

`id` int ***Read-only***
the disk id

`type` enum ***Read-only***

type	Description
internal	Freebox internal disk
usb	usb disk
sata	sata disk
nvme	nvme disk

`state` enum

state	Description
error	Disk has error
disabled	Disk is disabled
enabled	Disk is enabled
formatting	Disk is formatting

`connector` int ***Read-only***
Disk physical connector id

`total_bytes` int ***Read-only***
Disk size (in bytes)

`table_type` int ***Read-only***

table_type	
msdos	
gpt	
superfloppy	
empty	

`model` string ***Read-only***
Disk model

`serial` string ***Read-only***
Disk serial number

`firmware` string ***Read-only***
Disk firmware version

`temp` int ***Read-only***
Disk temperature (when supported) in °C

`operation_pct` OperationProgress ***Read-only***
partition operation progress

`partitions[]` array of DiskPartition ***Read-only***
list of disk partitions

`idle` bool ***Read-only***
is disk idle (when available)

`idle_duration` int ***Read-only***
disk idle duration (in seconds) (when available)

`spinning` bool ***Read-only***
is disk spinning (when available)

`active_duration` int ***Read-only***
disk activity duration (in seconds) (when available)

`time_before_spindown` int ***Read-only***
seconds left before disk spin down (in seconds) (when available)

`read_requests` int ***Read-only***
Number of read requests sent since to disk since boot (when available)

`read_error_requests` int ***Read-only***
Number of read requests in error since boot. Might indicate disk failure (when available)

`write_requests` int ***Read-only***
Number of write requests sent since to disk since boot (when available)

`write_error_requests` int ***Read-only***
Number of write requests in error since boot. Might indicate disk failure (when available)

Storage Disk API

Get the list of disks

`GET /api/v8/storage/disk/`

Returns the collection of all StorageDisk

Example request:

```
GET /api/v8/storage/disk/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "idle_duration": 368,
            "spinning": true,
            "table_type": "msdos",
            "firmware": "PB2ICC0E",
            "type": "internal",
            "idle": true,
            "connector": 0,
            "id": 1,
            "state": "enabled",
            "time_before_spindown": 232,
            "total_bytes": 250059350016,
            "model": "Hitachi HCC545025B9A300",
            "active_duration": 0,
            "temp": 51,
            "serial": "GSCH35VC",
            "partitions": [
                {
                    "fstype": "ext4",
                    "total_bytes": 245091500032,
                    "label": "Disque dur",
                    "id": 3,
                    "fsck_result": "no_run_yet",
                    "state": "mounted",
                    "disk_id": 1,
                    "free_bytes": 68120969216,
                    "used_bytes": 164520534016,
                    "path": "L0Rpc3F1ZSBkdXI="
                }
            ]
        },
        {
            "type": "usb",
            "total_bytes": 125435904,
            "connector": 1,
            "id": 1001,
            "active_duration": 0,
            "partitions": [
                {
                    "fstype": "ext4",
                    "total_bytes": 121418752,
                    "label": "Disque 1",
                    "id": 1002,
                    "fsck_result": "no_run_yet",
                    "state": "mounted",
                    "disk_id": 1001,
                    "free_bytes": 108904448,
                    "used_bytes": 6245376,
                    "path": "L0Rpc3F1ZSAx"
                }
            ],
            "idle_duration": 0,
            "state": "enabled",
            "idle": false,
            "spinning": false,
            "model": "",
            "table_type": "gpt",
            "temp": 0,
            "serial": "",
            "firmware": ""
        }
    ]
}
```


Get a given disk info

`GET /api/v8/storage/disk/{id}`

Returns the StorageDisk with the given id

Example request:

```
GET /api/v8/storage/disk/1 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "idle_duration": 464,
        "spinning": true,
        "table_type": "msdos",
        "firmware": "PB2ICC0E",
        "type": "internal",
        "idle": true,
        "connector": 0,
        "id": 1,
        "state": "enabled",
        "time_before_spindown": 136,
        "total_bytes": 250059350016,
        "model": "Hitachi HCC545025B9A300",
        "active_duration": 0,
        "temp": 51,
        "serial": "GSCH35VC",
        "partitions": [
            {
                "fstype": "ext4",
```

```
                "total_bytes": 245091500032,
                "label": "Disque dur",
                "id": 3,
                "fsck_result": "no_run_yet",
                "state": "mounted",
                "disk_id": 1,
                "free_bytes": 68120969216,
                "used_bytes": 164520534016,
                "path": "L0Rpc3F1ZSBkdXI="
            }
        ]
    }
}
```


Update a disk state

`PUT /api/v8/storage/disk/{id}`

Enable/Disable a disk

Example request:

```
PUT /api/v8/storage/disk/1 HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
   "state": "disabled"
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "type": "usb",
        "total_bytes": 125435904,
        "connector": 1,
        "id": 1001,
        "active_duration": 0,
        "partitions": [
            {
                "fstype": "ext4",
                "total_bytes": 121418752,
                "label": "Disque 1",
                "id": 1002,
                "fsck_result": "no_run_yet",
                "state": "umounted",
                "disk_id": 1001,
                "free_bytes": 108904448,
                "used_bytes": 6245376,
                "path": "L0Rpc3F1ZSAx"
            }
        ],
        "idle_duration": 0,
        "state": "disabled",
        "idle": false,
        "spinning": false,
        "model": "",
        "table_type": "gpt",
        "temp": 0,
        "serial": "",
        "firmware": ""
    }
}
```


Get FS advices

`GET /api/v8/storage/disk/{disk_id}/fsadvice?partition_id={partition_id}&dedicated_disk={bool}`

Check disk FS and get formatting advices.

To be able to get FS advice for a disk you need to provide the disk_id. Specify dedicated_disk for a disk that will only be used with the Freebox server (no need to specify it for a SATA internal disk). If the disk is empty do not specify partition_id in order to get advice for creating a new one. If the disk contains a partition specify the partition_id that needs to be checked.

Example request:

```
GET /api/v8/storage/disk/1000/fsadvice?partition_id=1003&dedicated_disk=false HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
    "result":
    {
        "fstype": "exfat",
        "table_type": "gpt",
        "reason": "max_file_size",
        "partitions_to_delete": [
        {
            "fstype": "exfat",
            "total_bytes": 1000000000000,
            "label": "EFI",
            "id": 1001,
            "internal": false,
            "fsck_result": "no_run_yet",
            "state": "mounted",
            "disk_id": 1000,
            "free_bytes": 1000000000000,
            "used_bytes": 1310000,
            "path": "L0Rpc3F1ZSAxIDE="
        },
        {
            "fstype": "exfat",
            "total_bytes": 1000000000000,
            "label": "DATA",
            "id": 1002,
            "internal": false,
            "fsck_result": "no_run_yet",
            "state": "mounted",
            "disk_id": 1000,
            "free_bytes": 1000000000000,
            "used_bytes": 1310000,
            "path": "L1ZvbHVtZSAxMDAwR28="
        },
```

```
        ]
    },
}
```

Reasons can be one of the following:

Reason	Description
max_file_size	Performance and bigger that 4GB files support
perf_and_compat	Performance and device compatibility
sata_performance	Performance for SATA disk
nvme_performance	Performance for NVMe disk
no_partition	Missing partition id on already formatted disk
partition_error	Partition is in error state

Format a disk

`PUT /api/v8/storage/disk/{id}/format/`

Format the disk with the given id

To be able to format a disk you need to provide the following parameters (JSON encoded). There will be one partition using all the available space on disk. All previous data will be lost.

This parameters will be ignored if you format the Freebox internal disk

Parameters

- **table_type** (*string*) – The partition table format
- **fs_type** (*string*) – The partition type
- **label** (*string*) – The partition label

NOTE: once started you can monitor the format process getting the disk information (see StorageDisk operation_pct field)

Example request:

```
PUT /api/v8/storage/disk/1001/format HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "label": "freebox",
   "fs_type": "vfat",
   "table_type": "msdos"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true
}
```

Storage Partition API

Get the list of partitions

`GET /api/v8/storage/partition/`

Returns the collection of all DiskPartition

Example request:

```
GET /api/v8/storage/partition/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": [
        {
            "fstype": "ext4",
            "total_bytes": 245091500032,
            "label": "Disque dur",
            "id": 3,
            "fsck_result": "no_run_yet",
            "state": "umounted",
            "disk_id": 1,
            "free_bytes": 68120969216,
            "used_bytes": 164520534016,
            "path": "L0Rpc3F1ZSBkdXI="
        },
        {
            "fstype": "vfat",
            "total_bytes": 123485184,
            "label": "freebox",
            "id": 1002,
            "fsck_result": "no_run_yet",
            "state": "mounted",
            "disk_id": 1001,
            "free_bytes": 123484672,
            "used_bytes": 512,
            "path": "L2ZyZWVib3g="
        }
    ]
}
```

Get a given partition info

`GET /api/v8/storage/partition/{id}`

Returns the DiskPartition with the given id

Example request:

```
GET /api/v8/storage/partition/1002 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "fstype": "vfat",
```

```
        "total_bytes": 123485184,
        "label": "freebox",
        "id": 1002,
        "fsck_result": "no_run_yet",
        "state": "mounted",
        "disk_id": 1001,
        "free_bytes": 123484672,
        "used_bytes": 512,
        "path": "L2ZyZWVib3g="
    }
}
```


Update a partition state

`PUT /api/v8/storage/partition/{id}`

Enable/Disable a partition

Example request:

```
PUT /api/v8/storage/partition/1 HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
   "state" : "umounted"
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "fstype": "vfat",
        "total_bytes": 123485184,
        "label": "freebox",
        "id": 1002,
        "fsck_result": "no_run_yet",
        "state": "umounted",
        "disk_id": 1001,
        "free_bytes": 123484672,
        "used_bytes": 512,
        "path": "L2ZyZWVib3g="
    }
}
```


Check a partition

`PUT /api/v8/storage/partition/{id}/check/`

Checks the partition with the given id

To be able to check a partition you need to provide the following parameters (JSON encoded):

Parameters

- **checkmode** (*enum*) – 'ro' for read only check, 'rw' to attempt to repair errors

NOTE: once started you can monitor the fsck process getting the partition information (see DiskPartition operation_pct field)

Example request:

```
PUT /api/v8/storage/partition/1002/check HTTP/1.1
Host: mafreebox.freebox.fr
```


```
{
   "checkmode": "ro"
}
```


Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true
}
```


Storage Config

StorageConfig has the following attributes:

`StorageConfig`

`external_pm_enabled` **bool**
: enable/disable external disk power management

`external_pm_idle_before_spindown` **int**
: idle time in minutes to wait before spinning down an external disk

Storage config API

Get the current storage configuration

`GET /api/v8/storage/config/`

Get the StorageConfig

Example request:

```
GET /api/v8/storage/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
  "success": true,
  "result": {
     "external_pm_idle_before_spindown": 10,
     "external_pm_enabled": true
  }
}
```


Update the External Storage configuration

`PUT /api/v8/storage/config/`

Update the StorageConfig

Example request:

```
PUT /api/v8/storage/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
   "external_pm_enabled": false
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
    "success": true,
    "result": {
        "external_pm_idle_before_spindown": 10,
        "external_pm_enabled": false
    }
}
```

RAID API [UNSTABLE]

This API allows you to manage the Freebox internal raid arrays for disks connected to the Freebox

This API is unstable, it can be modified without notice in next releases.

RAID API Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
inval	Invalid parameters(s)
no_sys	Function not available
member_not_found	No member found
members_too_many	Too many members
array_not_found	RAID array not found
array_stop_failed	Error when stopping the RAID array
array_start_failed	Error when starting the RAID array
array_destroy_failed	Error when destroying the RAID array
array_not_running	The RAID array is not active
array_not_stopped	The RAID array is not stopped
array_degraded	The RAID array is degraded
array_not_degraded	The RAID array is not degraded
array_complete	The RAID array is full
already_member	The specified disks are already members of a RAID array
disk_more_than_once	The same disk has been specified more than once
disks_missing	Insufficient number of disks
bad_disk_location	Only internal drives can be used in a RAID array
disk_internal	This disk cannot be used in a RAID array
disk_busy	Disk is busy
create_failed	RAID array creation failed
create_too_many_members	The number of disks is too high (basic)
create_not_enough_members	The number of disks is too small
create_bad_member_count	The number of disks is incorrect (raid10)
sync_action_bad_level	This type of RAID array does not support synchronization
sync_action_array_busy	This RAID array is being resynchronized/restored
sync_action_bad_action	It is not possible to force resynchronization manually
sync_action_failed	This action has been denied
check_interval_too_large	Check interval is too long
check_interval_not_supported	This check interval is not supported
remove_bad_level	This type of RAID array does not allow member removal
remove_not_enough_active	Not enough active members to allow removal of a member
remove_failed	Failure to remove a member
add_too_many	Too many new members
add_member_too_small	One of the members is too small to be added to this array
add_failed	Failed to add member
member_examine_data_failed	Unable to examine member data
sync_speed_min_greaterthan_max	Minimum sync speed is more important than maximum speed
sync_speed_min_toohigh	The minimum sync speed is too high
sync_speed_max_toohigh	The maximum sync speed it too high
sync_speed_min_toolow	The minimum sync speed is too low
sync_speed_max_toolow	The maximum sync speed is too low
sync_speed_set_failed	Error changing synchronization speed
grow_bad_level	RAID level migration not possible
grow_not_enough_disks	Not enough disks for expansion
grow_failed	Expansion failed
grow_array_busy	Cannot extend a busy RAID array
grow_member_too_small	One of the members is too small to expand the raid array
rescan_member_failed	One or more members could not be rescanned
add_spares_busy	Cannot add out-of-sync disks when the array is busy
add_spares_nospares	No out-of-sync member detected

error_code	Description
add_spares_complete	The RAID Array is full and cannot add an out of sync member
add_spares_failed	Failed to add out-of-sync disks

RAID API objects

RAID Array object

`RaidArray`

`id` `int` ***Read-only***

unique id of this array. Used as a reference for API calls.

`state` `enum`

state	Description
stopped	Array is stopped
running	Array is running
error	Array is in error

`name` `string`

The array name

`level` `enum`

level	Description
basic	Basic RAID level, like a single drive raid1 array
raid0	RAID 0
raid1	RAID 1
raid5	RAID 5
raid10	RAID 10

`disk_id` `int` ***Read-only***

The disk id of the array, for use with the disk format API.

`uuid` `string` ***Read-only***

The array unique id. Only this id is guaranteed to stay stable across reboots.

`sync_action` `enum` ***Read-only***

sync_action	Description
idle	Array is idle
resync	Sync operation in progress
recover	Recover operation in progress
check	Array is being checked
repair	Repair operation in progress
reshape	Array growth in progress
frozen	Array is frozen

`sysfs_state` `enum` ***Read-only***

Low-level Linux-specific md state value read in sysfs array_state property.

sysfs_state
clear
inactive
suspended
readonly
read_auto
clean
active
write_pending
active_idle

`array_size` `int` ***Read-only***

Size of array in bytes.

`raid_disks` `int` ***Read-only***

Number of members that should be in this array.

`sync_speed` `int` ***Read-only***

Sync speed in bytes per second

`sync_completed_pos` `int` ***Read-only***

Current position of sync process.

`sync_completed_end` `int` ***Read-only***

End position of sync process: total of bytes to sync.

`sync_completed_percent` `int` ***Read-only***

Percentage of sync completion.

`check_interval` `int` ***Read-only***

Check interval in seconds.

`last_check` `int` ***Read-only***

Unix timestamp of last check in seconds.

`next_check` `int` ***Read-only***

Unix timestamp of next check in seconds. Might be 0 if check_interval is 0.

`degraded` `bool` ***Read-only***

Whether the array is degraded or not.

`members[]` `array of` **RaidMember**

List of members of this array

RAID Member object

`RaidMember`

`id` `int` ***Read-only***

unique id of this member. This corresponds to the disk id, usable with the Storage Disk API.

`array_id` `int` ***Read-only***
id of the array this member is in

`role` `enum` ***Read-only***

role	Description
active	Active member of the array
faulty	Faulty member
spare	Member kept as spare
missing	Missing (removed or dead) member of the array

`set_name` `string` ***Read-only***
name of the array this member is into

`set_uuid` `string` ***Read-only***
uuid of the array this member is into

`dev_uuid` `string` ***Read-only***
uuid of this member

`device_location` `enum` ***Read-only***
internal location of this member. Possible slot values: sata-internal-p0, sata-internal-p1, sata-internal-p2, sata-internal-p4

`total_bytes` `int` ***Read-only***
size of this member in bytes

`active_device` `int` ***Read-only***
device number inside the array

`corrected_read_errors` `int` ***Read-only***
Device read errors count

`sct_erc_supported` `bool` ***Read-only***
Whether SCT_ERC is supported by the device according to its S.M.A.R.T. data.

`sct_erc_enabled` `bool` ***Read-only***
Whether SCT_ERC is enabled on the device according to its S.M.A.R.T. data.

`disk` `RaidDisk` ***Read-only***
A few properties of the disk.

RAID Disk object

`RaidDisk`

`model` `string` ***Read-only***
Disk model.

`serial` `string` ***Read-only***
Disk serial number.

`firmware` `string` ***Read-only***
Disk firmware revision

`temp` `int` ***Read-only***
Disk temperature in °C.

RAID API actions

Get the list of RAID arrays

`GET /api/v8/storage/raid/`

Returns the collection of all `RaidArray`

Example request:

```
GET /api/v8/storage/raid/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": [
        {
            "degraded": false,
            "raid_disks": 4,
            "next_check": 0,
            "sync_action": "idle",
            "level": "raid5",
            "uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895",
            "sysfs_state": "clear",
            "id": 0,
            "sync_completed_pos": 0,
            "members": [
                {
                    "total_bytes": 1000000000000,
                    "active_device": 0,
                    "id": 1000,
                    "corrected_read_errors": 0,
                    "array_id": 0,
                    "disk": {
                        "firmware": "02.01A02",
                        "temp": 43,
                        "serial": "WD-WX91A42F69NE",
                        "model": "WDC WD10JUCX-56WPNY0"
                    },
                    "role": "active",
                    "sct_erc_supported": false,
                    "sct_erc_enabled": false,
                    "dev_uuid": "666793c9-2d04-9d9e-5c8a-2f13eb7f2e9e",
                    "device_location": "sata-internal-p1",
                    "set_name": "Freebox",
                    "set_uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895"
                },
                {
                    "total_bytes": 1000000000000,
                    "active_device": 1,
                    "id": 2000,
                    "corrected_read_errors": 0,
                    "array_id": 0,
                    "disk": {
                        "firmware": "02.01A02",
                        "temp": 47,
```

```
                        "serial": "WD-WX91A42F1337",
                        "model": "WDC WD10JUCX-56WPNY0"
                    },
                    "role": "active",
                    "sct_erc_supported": false,
                    "sct_erc_enabled": false,
                    "dev_uuid": "231b35d0-c37f-9d3c-be7a-b7b8485341ce",
                    "device_location": "sata-internal-p0",
                    "set_name": "Freebox",
                    "set_uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895"
                },
                {
                    "total_bytes": 1000000000000,
                    "active_device": 2,
                    "id": 3000,
                    "corrected_read_errors": 0,
                    "array_id": 0,
                    "disk": {
                        "firmware": "02.01A02",
                        "temp": 46,
                        "serial": "WD-WX91A42FZ3I9",
                        "model": "WDC WD10JUCX-56WPNY0"
                    },
                    "role": "active",
                    "sct_erc_supported": false,
                    "sct_erc_enabled": false,
                    "dev_uuid": "d28e5fd8-5e2a-baf3-fd24-6fe5ff2593d6",
                    "device_location": "sata-internal-p2",
                    "set_name": "Freebox",
                    "set_uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895"
                },
                {
                    "total_bytes": 1000000000000,
                    "active_device": 3,
                    "id": 4000,
                    "corrected_read_errors": 0,
                    "array_id": 0,
                    "disk": {
                        "firmware": "02.01A02",
                        "temp": 46,
                        "serial": "WD-WX91A42F1333",
                        "model": "WDC WD10JUCX-56WPNY0"
                    },
                    "role": "active",
                    "sct_erc_supported": false,
                    "sct_erc_enabled": false,
                    "dev_uuid": "fdf5a84a-c427-e1ef-aa12-1732d2cf689f",
                    "device_location": "sata-internal-p3",
                    "set_name": "Freebox",
                    "set_uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895"
                }
            ],
            "array_size": 3000000000000,
            "state": "running",
            "sync_speed": 0,
            "name": "Freebox",
            "check_interval": 0,
            "disk_id": 6000,
            "last_check": 1576082428,
            "sync_completed_end": 0,
            "sync_completed_percent": 0
        }
    ]
}
```


Get a given RAID array info

`GET /api/v8/storage/raid/{id}`

Returns a single RaidArray

Example request:

```
GET /api/v8/storage/raid/0 HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```


```
{
    "success": true,
    "result": {
        "degraded": false,
        "raid_disks": 4,
        "next_check": 0,
        "sync_action": "idle",
        "level": "raid5",
        "uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895",
        "sysfs_state": "clear",
        "id": 0,
        "sync_completed_pos": 0,
        "members": [
            {
                "total_bytes": 1000000000000,
                "active_device": 0,
                "id": 1000,
                "corrected_read_errors": 0,
                "array_id": 0,
                "disk": {
                    "firmware": "02.01A02",
                    "temp": 43,
                    "serial": "WD-WX91A42F69NE",
                    "model": "WDC WD10JUCX-56WPNY0"
                },
                "role": "active",
                "sct_erc_supported": false,
                "sct_erc_enabled": false,
                "dev_uuid": "666793c9-2d04-9d9e-5c8a-2f13eb7f2e9e",
                "device_location": "sata-internal-p1",
                "set_name": "Freebox",
                "set_uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895"
            },
            {
                "total_bytes": 1000000000000,
                "active_device": 1,
```

```
                "id": 2000,
                "corrected_read_errors": 0,
                "array_id": 0,
                "disk": {
                    "firmware": "02.01A02",
                    "temp": 47,
                    "serial": "WD-WX91A42F1337",
                    "model": "WDC WD10JUCX-56WPNY0"
                },
                "role": "active",
                "sct_erc_supported": false,
                "sct_erc_enabled": false,
                "dev_uuid": "231b35d0-c37f-9d3c-be7a-b7b8485341ce",
                "device_location": "sata-internal-p0",
                "set_name": "Freebox",
                "set_uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895"
            },
            {
                "total_bytes": 1000000000000,
                "active_device": 2,
                "id": 3000,
                "corrected_read_errors": 0,
                "array_id": 0,
                "disk": {
                    "firmware": "02.01A02",
                    "temp": 46,
                    "serial": "WD-WX91A42FZ3I9",
                    "model": "WDC WD10JUCX-56WPNY0"
                },
                "role": "active",
                "sct_erc_supported": false,
                "sct_erc_enabled": false,
                "dev_uuid": "d28e5fd8-5e2a-baf3-fd24-6fe5ff2593d6",
                "device_location": "sata-internal-p2",
                "set_name": "Freebox",
                "set_uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895"
            },
            {
                "total_bytes": 1000000000000,
                "active_device": 3,
                "id": 4000,
                "corrected_read_errors": 0,
                "array_id": 0,
                "disk": {
                    "firmware": "02.01A02",
                    "temp": 46,
                    "serial": "WD-WX91A42F1333",
                    "model": "WDC WD10JUCX-56WPNY0"
                },
                "role": "active",
                "sct_erc_supported": false,
                "sct_erc_enabled": false,
                "dev_uuid": "fdf5a84a-c427-e1ef-aa12-1732d2cf689f",
                "device_location": "sata-internal-p3",
                "set_name": "Freebox",
                "set_uuid": "a4f1fbf3-f8e7-453f-19ec-842d6f4e2895"
            }
        ],
        "array_size": 3000000000000,
        "state": "running",
        "sync_speed": 0,
        "name": "Freebox",
        "check_interval": 0,
        "disk_id": 6000,
        "last_check": 1576082428,
        "sync_completed_end": 0,
        "sync_completed_percent": 0
    }
}
```


Create a RAID array

`POST /api/v8/storage/raid/`

Send a `RaidArray` with the following members:

- level
- name
- members

Each member should have the following property:

- id

Delete a RAID array

`DELETE /api/v8/storage/raid/{id}`

Start or stop a RAID array

Send a `RaidArray` with properties “id” and “state”.

This is used to start and stop an array by changing the state to “stopped” or “running”. These are the only two supported operations. Any change to other fields is ignored.

`PUT /api/v8/storage/raid/{id}`

Force start a RAID array

In case an array is incomplete, but has enough data to start in degraded mode, it won’t start automatically at boot, and the force start can be used. Can only be done if array state is “error”.

`POST /api/v8/storage/raid/{id}/forcestart`

Remove faulty members from RAID array

In case an array has faulty members, it might be desirable to delete them to add others members instead. Can only be done if array is not running.

`DELETE /api/v8/storage/raid/{id}/members/faulty`

Add members to an existing array that has missing members

In case an array is incomplete (has missing members), either because they were removed physically, or after becoming faulty, it’s possible to add new members to let the reconstruction happen. Can only be done if array is not running.

`PUT /api/v8/storage/raid/{id}/members`

Send a json object containing a “members” property, which is array of `RaidMember` objects. Only the “id” property of each member is required.

Re-add out-of-sync members that appear as spares

In case an array has been force-started without a member, and then said member is physically plugged, it won’t be added automatically and will appear with the “spare” role, this operation must be used. Can only be done if the array has a member with the “spare” role, and is not running.

`POST /api/v8/storage/raid/{id}/members/addspares`

SFP

SFP

On boxes that have has_lan_sfp set to true in their SystemConfig information, it is possible to configure the LAN SFP port.

SFP Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
inval	invalid parameters
noent	invalid id
internal	system internal error

SFP config object

SFP config object has the following properties:

SfpConfig

sfp_type_forced bool
Indicate whether the SFP type is forced

sfp_type_forced_value enum
What SFP type is forced (valid only when *sfp_type_forced* is true). Valid values are provided in *available_sfp_types*

available_sfp_types[] array of enum ***Read-only***
array containing what SFP types can be configured on the LAN SFP port. Possible values are listed in the following table:

Type	Description
p2p_1g	1000BASE-X
p2p_2d5g_no_aneg	2500BASE-X
p2p_10g	10GBASE-R
copper_1g	1000BASE-T
copper_sgmii_1g	SGMII
copper_sgmii_10g	USXGMII

SFP status object

SFP status object has the following properties:

SfpStatus

present bool ***Read-only***
Indicates whether an SFP module present in the port

eeprom_valid bool ***Read-only***
Indicates whether the SFP module has a valid EEPROM

supported bool ***Read-only***
Indicates whether the SFP module is supported

type enum ***Read-only***
SFP type read from EEPROM

power_good bool ***Read-only***
SFP port is powered

link bool ***Read-only***
link status

vendor_name string ***Read-only***
vendor name

part_number string ***Read-only***
part number

hardware_rev string ***Read-only***
hardware revision

serial_number string ***Read-only***
serial number

SFP API

Get SFP status

GET /api/v11/sfp/status
Returns the `SFP status object`

Example request:

```
GET /api/v11/sfp/status HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
    "type": "copper_1g",
    "present": true,
    "link": true,
    "supported": true,
    "vendor_name": "SFP Vendor",
    "serial_number": "1122334455",
    "part_number": "SFP-V-Part-01R",
    "power_good": true,
    "hardware_rev": "A",
    "eeprom_valid": true
  }
}
```

Get SFP config

Get the SfpConfig

Example request:

```
GET /api/v11/sfp/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
    "sfp_type_forced": false,
    "sfp_type_forced_value": "",
    "available_sfp_types": [
      "p2p_1g",
      "p2p_10g",
      "copper_1g",
      "copper_sgmii_1g",
      "copper_usxgmii_10g"
    ]
  }
}
```

Update SFP config

`PUT /api/v11/sfp/config`

Example request:

```
PUT /api/v11/sfp/config/ HTTP/1.1
Host: mafreebox.freebox.fr
```

```
{
  "sfp_type_forced": true,
  "sfp_type_forced_value": "copper_1g"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
    "sfp_type_forced_value": "copper_1g",
    "sfp_type_forced": true,
    "available_sfp_types": [
      "p2p_1g",
      "p2p_10g",
      "copper_1g",
      "copper_sgmii_1g",
      "copper_usxgmii_10g"
    ]
  }
}
```

Update

Update Status

The Update API allows you to access box firmware update status

Update status object

Update status object have the following properties

`UpdateStatus`

`state` **enum**

update current state

State	Description
initializing	update process is initializing
upgrading	firmware is upgrading
up_to_date	firmware is up to date
error	an error occurred during update

`upgrade_state` **UpgradeState**

Upgrade status object

Details of current box upgrade. Only relevant for “upgrading” and “upgrade_failed” states.

`UpgradeState`

`state` **enum**

upgrade state

State	Description
downloading	downloading update
download_failed	update downloading has failed
checking	checking the downloaded data
check_failed	downloaded data check has failed
prepare_write	preparing to write data
prepare_write_failed	preparing to write data ha failed
writing	writing the data
write_failed	data writing has failed
reread	checking written data
reread_failed	written data checking has failed
commit	applying the update
commit_failed	update applying has failed

`old_version` **string**

current firmware version

`new_version` **string**

new firmware version being downloaded

`percent` **int**

download progress if state is downloading

`error_string` **string**

update error if state is download_failed

Update API

Get the update status

`GET /api/v11/update/`

Returns the `Upgrade status object`

Example request:

```
GET /api/v11/update/ HTTP/1.1
Host: mafreebox.freebox.fr
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "success": true,
  "result": {
    "state": "auto_up_to_date"
  }
}
```

Virtual machines

VM API [UNSTABLE]

This API allows to control VMs on boxes that have has_vm to true in their SystemConfig information.

VM API Errors

When attempting to access this API, you may encounter the following errors:

error_code	Description
initfail	VM cannot be initialized
startfail	The VM cannot be launched
inval	Invalid parameter
nomem	Not enough memory available
already_running	The VM is already running
not_running	The VM is not running
too_big	Size too big
too_small	Size too small
exists	File exists
too_many_vms	The maximum number of configurable VMs has been reached
no_such_vm	VM does not exist
disk_in_use	The disk is already in use
nocpu	Not enough CPUs available
no_such_usb_port	USB port does not exist
usb_in_use	Another VM is already using USB
usb_init_fail	Unable to initialize USB
disk_not_qcow2	The disk is not in Qcow2 format
unsupported_disk_type	Unsupported disk format
file_not_found	Disk file not found
efi_file_in_use	EFI settings file is already in use
efi_file_fail	Cannot open EFI settings file
distro_http	Internal http error
distro_sig	Internal sig error
distro_json	Internal json error
create	Unable to create file
perm_own	Incorrect permission
open_info	Unable to open file for information
open_resize	Cannot open file for resizing
resize_trunc	Unable to resize raw disk
power_button	Unable to send shutdown to VM
restart	Cannot send restart to VM
open_launch_disk	Error opening disk file
open_launch_cd	Error opening cdrom file
start_nodisk	Cannot start without disk
init_vm_control	Unable to initialize VM control
set_nodisk	Cannot set up a VM without disk
set_badformat	Unsupported disk format
save_data	Cannot save VM settings
stop_control	Cannot stop VM control
info	Unable to retrieve disk info
info_parse	Unable to analyze disk information
info_novirtual	Unable to retrieve disk size
info_noactual	Unable to retrieve actual disk size
info_noformat	Unable to retrieve disk format
create_qcow	Unable to create qcow2 disk
resize_qcow	Unable to resize qcow2 disk
set_too_many_disks	The VM has too many disks
set_empty_disk_path	Empty disk path

error_code	Description
task_notfound	The task does not exist
not_stopped	The VM is not stopped

VM API objects

VM object

`VM`

`id` `int` ***Read-only***
unique id of this VM

`name` `string`
Name of this VM. Max 31 characters.

`disk_path` `string`
Base64-encoded path to the hard disk image of this VM.

`disk_type` `enum`
Type of disk image.

disk_type	Description
raw	Raw disk data
qcow2	Qcow2 image type. Usually qcow version 3. Note: not all features are supported. In particular, reference to other images is disabled.

`cd_path` `string` ***Optionnal***
Base64-encoded path to CDROM device ISO image. Optional.

`memory` `int`
Memory allocated to this VM in megabytes.

`vcpus` `int`
Number of virtual CPUs to allocate to this VM.

`status` `enum` ***Read-only***
VM status

status	Description
stopped	VM is stopped
running	VM is running
starting	VM is starting up. Transitional state
stopping	VM is being stopped. Transitional state

`enable_screen` `bool`
Whether or not this VM should have a virtual screen, to use with the VNC websocket protocol.

`bind_usb_ports[]` `array of enum`
List of ports that should be bound to this VM. Only one VM can use USB at given time, whether is uses only one or all USB ports. The list of system USB ports is available in `VmSystemInfo`. For example: “usb-external-type-a”, “usb-external-type-c”.

`enable_cloudinit` `bool`
Whether or not to enable passing data through cloudinit. This uses the NoCloud iso image method; it will add a virtual cdrom drive (distinct from the one passed by cd_path) with the data in cloudinit_userdata and cloudinit_hostname when enabled.

`cloudinit_hostname` `string`
When cloudinit is enabled, hostname desired for this VM. Max 59 characters.

`cloudinit_userdata` `string`
When cloudinit is enabled, raw yaml to be passed in the user-data file. Maximum 32767 characters.

`mac` `string` ***Read-only***
VM ethernet interface MAC address.

`os` `string`
Type of OS used for this VM. Only used to set an icon for now. Example values:

- unknown
- fedora
- debian
- ubuntu
- freebsd
- opensuse
- centos
- jeedom
- homebridge

VM System Info object

`VmSystemInfo`

`total_memory` `int` ***Read-only***
Total memory available to VMs.

`used_memory` `int` ***Read-only***
Currently used memory by all VMs.

`total_cpus` `int` ***Read-only***
Total number of vCPUs available to VMs.

`used_cpus` `int` ***Read-only***
Currently used vCPUs by all VMs.

`usb_ports[]` `array of string` ***Read-only***
List of USB ports available on this system

`usb_used` `bool` ***Read-only***
Whether a VM is currently using USB. (only one can use USB at a given time)

VM Distribution object

`VmDistribution`

`name` `string` ***Read-only***
Name of downloadable distribution image.

`url` `string` ***Read-only***
URL of distribution. Usually an arm64 qcow2 cloud image, supporting EFI boot and cloud-init.

`hash` `string` ***Read-only***
Hash in the format sha256:<hash> or sha512:<hash>; or a URL to a SHA256SUMS or SHA512SUMS file (used by Ubuntu, Debian), or to a -CHECKSUM file (used by Fedora). It is designed to be passed as-is to the download add API.

`os` `string` ***Read-only***
OS of this distribution image; to be passed as a os type in the `VM`.

VM Disk info object

`VmDiskInfo`

`type` `enum` ***Read-only***
Type of disk, just like in `VM.disk_type`

`actual_size` `int` ***Read-only***
Space used by virtual image on disk. This is how much filesystem space is consumed on the box.

`virtual_size` `int` ***Read-only***
Size of virtual disk. This is the size the disk will appear inside the VM.

VM Disk task object

`VmDiskTask`

`id` `int` ***Read-only***
Task id.

`type` `enum` ***Read-only***
Type of disk operation:

- create
- resize

`done` `bool` ***Read-only***
Is task done

`error` `bool` ***Read-only***
Is task in error

VM API actions

Get VM System Info

`GET /api/v8/vm/info/`
Returns a `VmSystemInfo`

Get Installable VM distributions

`GET /api/v8/vm/distros/`
Returns a collection of `VmDistribution`

Get the list of all VMs

`GET /api/v8/vm/`
Returns a collection of `VM`

Get a VM

`GET /api/v8/vm/{id}`
Returns a `VM` object

Add a VM

`POST /api/v8/vm/`
Needs to be passed a `VM` object

Delete a VM

`DELETE /api/v8/vm/{id}`

Only works if vm is stopped.

Update a VM

`PUT /api/v8/vm/{id}`

Only works if vm is stopped.

Start a VM

`POST /api/v8/vm/{id}/start`

Only works if vm is stopped.

Send a powerbutton signal to a VM

`POST /api/v8/vm/{id}/powerbutton`

This will send an ACPI shutdown button event to the VM, so that it can decide to shutdown itself.

Only works if vm is running.

Stop a VM

Immediately stops the VM without any safety.

`POST /api/v8/vm/{id}/stop`

Only works if vm is running.

Reset a VM

Immediately restarts the VM without any safety.

`POST /api/v8/vm/{id}/restart`

Only works if vm is running.

Watch for VM status changes

You should use the websocket `RegisterAction` API with the `vm_state_changed` event to watch for changes in VM status, instead of polling.

The event will contain this object:

`VmStateChange`

`id` `int` ***Read-only***
VM id.

`status` `enum` ***Read-only***
New `VM.status`.

You can also watch for `lan_host_l3addr_reachable` and compare it with `VM.mac` to get the VM IP when it starts.

VM virtual console

The serial port of the VM is available via a WebSocket.

`GET /api/v8/vm/{id}/console`

It uses the QEMU websocket chardev device. Call must be authentified like the rest of the API.

VM virtual screen

When `VM.enable_screen` is `true`, the VM will have a VNC over websocket device available.

`GET /api/v8/vm/{id}/vnc`

It uses the QEMU VNC websocket device. Call must be authentified like the rest of the API. This device should work with noVNC unmodified.

Get information on a virtual disk

`POST /api/v8/vm/disk/info`

Parameters

- **disk_path** (*string*) – base64-encoded disk path

Returns a `VmDiskInfo` object.

Create a virtual disk

`POST /api/v8/vm/disk/create`

Parameters

- **disk_path** (*string*) – base64-encoded disk path
- **size** (*int*) – Size in bytes of virtual disk.
- **disk_type** (*enum*) – Type of `VM.disk_type`

Returns a task id. Task should not be polled, use the `vm_disk_task_done` websocket event with `RegisterAction`.

Resize a virtual disk

`POST /api/v8/vm/disk/resize`

Parameters

- **disk_path** (*string*) – base64-encoded disk path
- **size** (*int*) – New size of virtual disk
- **shrink_allow** (*bool*) – Whether shrinking the disk is allowed. Setting to true means this operation can be destructive.

Returns a task id. Task should not be polled, use the `vm_disk_task_done` websocket event with `RegisterAction`.

Get a virtual disk task

`GET /api/v8/vm/disk/task/{id}`

Returns a `VmDiskTask`

Delete a virtual disk task

`DELETE /api/v8/vm/disk/task/{id}`

Delete your tasks once they are done.