Developper 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 “4.0” Current major API version is: 4

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
- Api changes from version 2.0 to 3.0
- Api changes from version 3.0 to 4.0

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	“FreeboxServer1,1” for the Freebox Server revision 1,1
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

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",
   "api_version": "4.0",
   "api_base_url": "/api/",
   "device_type": "FreeboxServer1,2"
   "api_domain": "example.fbxos.fr",
   "https_available": true,
   "https_port": 3615
}
```

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/v4/login/`

API conventions

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

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/
3twgV6Y26qlyHXXODUfV3arLCwFoNB+IIrde1E/JoOry9oKvF8DZTo/Qm6o2KsdZ
dxs/YcIUsCvKX8WCKtH6la/kFCUcXIb8f1u+Y4pjj3PBmKI/1+Rs9GqB0kt1otyx
Q6bqxqBSgsrkuhCfRxwjbfBgmXjIZ/a4muY5uMI0gbl9zbMFEJHDojhH6TUB5qd0
JJlI61gldaT5Ci1aLbvVcJtdeGhElf7pOE9JrXINpP3NOJJaUSueAvxyj/WWoo0v
4KO7njox8F6jCHALNDLdTsX0FTGmUZ/s/QfJry3VNwyjCyWDy1ra4KWoqt6U7SzM
d5jENIZChM8TnDXJzqc+mu00cI3icn9bV9flYCXLTIsprB21wVSMh0XeBGylKxeB
S27oDfFq04XSox7JM9HdTt2hLK96x1T7FpFrBTnALzb7vHv9MhXqAT90fPR/8A==
-----END CERTIFICATE-----
```

Authentication

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

- Login

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 specifiy 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 specfic 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** string *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)

API List

Downloads

- Download
- Download Stats
- Download Files
- Download Trackers [UNSTABLE]
- Download Peers [UNSTABLE]
- Download Pieces
- Download Blacklist [UNSTABLE]
- Download Feeds
- Download Configuration

File System Api

- File System
- File Sharing Link
- File Upload

Air Media

- AirMedia API

RRD

- RRD [UNSTABLE]

Calls / Contacts

- Call
- Contacts

Configuration

- Connection API
- Lan
- Lan Browser
- Freeplug
- DHCP
- Ftp

- NAT
- Port Forwarding
- Incoming port configuration
- UPnP IGD
- LCD
- Network Share
- UPnP AV
- Switch
- Wi-Fi
- System
- VPN Server [UNSTABLE]
- VPN Client [UNSTABLE]

Storage

- Storage API [UNSTABLE]

Parental filter

- Parental Control

PVR

- PVR [UNSTABLE]
- PVR Programmed records
- PVR Finished records
- Storage media