Building requests for FreeMile60

All API requests will be built of the following base URL format:

```
http(s)://<device_ip>:<http_port>/cgi.lua/apiv1/<route>
```

Where:

- *device_ip* is the IP address at which your device's web configuration UI is accessible.
- *http_port* is the port on which the local webserver is running. The default is 80 for HTTP and 443 for HTTPS, but this can be changed via configuration settings.
- *Route* is the route defined in the API routes section below.

The device uses self-signed SSL certificates, so if you wish to use HTTPS, you'll need to choose a client that does not reject self-signed SSL certificates.

Authentication & authorization

Both the API and the local web interface use a cookie-based authentication mechanism reflected in the/login route below. You will need to fetch an authentication token and include it with any subsequent requests.

See an example for the Windows command line below:

```
curl -c ./freemile.cookie -H "Content-type: application/json" -X POST -d
"{\"username\": \"your_user\", \"password\": \"your_password\"}"
http://192.168.1.1/cgi.lua/apiv1/login
```

Response for successful authentication:

```
{"level":0,"token":"BBNM6PDHAAAAAAE6JF42BCPSFTHN2KXHDTCEBBNFXXFQ====","auth"
:true}
```

The API token could be copied from the answer or found in the stored FreeMile.cookie file. The file contains the following information:

```
# Netscape HTTP Cookie File
# https://curl.haxx.se/docs/http-cookies.html
# This file was generated by libcurl! Edit at your own risk.

192.168.99.130    FALSE    /    FALSE    0    token
BBNM6PDHAAAAAAE6JF42BCPSFTHN2KXHDTCEBBNFXXFQ====
```

The value *BBNM6PDHAAAAAAE6JF42BCPSFTHN2KXHDTCEBBNFXXFQ====* is the api_token that you will use in future requests. This value should be included in the Cookie header. See below for an example fetching system stats:

```
curl  -X GET -H "Cookie:
api_token=BBMWEPDHAAAAAABGKXET4YWU2O4T6VL57N7TVXXBL3VNXIRAWM%3D%3D%3D%3D%3D%
3D" http://192.168.99.130/cgi.lua/apiv1/stats?type=system
```

Authentication tokens become invalid if inactive for more than 30 minutes, or until the unit is rebooted.

User roles

There are two levels of user roles:

- **Admin** (level 0): This role can access and change all settings via the web UI and access all APIv1 routes. For devices that have SSH enabled, the admin user can also access the device over SSH.

- **Read-only** (level 9): This role can fetch system stats, but cannot perform any operations that affect the device. It can also view the Dashboard page of the web UI. For devices that have SSH enabled, the guest user can not access the device over SSH.

All FreeMile60 devices ship with a default admin user root/admin, but you should change this default user's password before using your device in production.

Getting device's configuration

Send a GET request to the /config route to get the configuration, please find the example:

```
curl -X GET -H "Cookie: api_token=YOUR_API_TOKEN"
http://192.168.1.1/cgi.lua/apiv1/config
```

Response for successful request:


```
{"config":{"system":{"authentication":{"method":"local"},"device_name":"Free
mile60","users":[{"username":"root","type":"admin","password_hash":"$1$5HDT9
3ii$Z71a.vMeJR4CU65PiEkcX.","enabled":true}],"hostname":"freemileap","reset_
button":{"enabled":true},"timezone":"Europe/Riga","device_location":"Office"
,"use_advanced_timezones":true},"wireless":{"radios":{"wlan0":{"channel_widt
h":2160,"ssid":"freemile","boresight_lock":false,"max_mcs":12,"passphrase":"
passphrase","sta_profiles":{"enabled":false},"antenna_kit":"none","channel":
4,"mode":"ap","ptp":false,"security":"wpapsk"}}},"network":{"traffic_control
":{"limit_download":{"enabled":false},"limit_upload":{"enabled":false}},"poe
_out_enabled":false,"failover":{"enabled":false},"wan":{"ipv6":{"enabled":fa
lse},"data_vlan":{"enabled":false},"ipv4":{"enabled":true,"prefix":24,"gatew
ay":"192.168.1.254","dns_servers":["1.1.1.1","1.0.0.1"],"ipaddr":"192.168.1.
1"},"alternative_local_ip":{"enabled":true,"ipaddr":"169.254.1.1"},"mgmt_vla
n":{"enabled":false},"ip_mode":"static"},"dhcp_snooping":{"option_82_injecti
on":{"enabled":false},"block_rogue_servers":false},"general":{"max_fdb_entri
es":0,"ageing_time":300,"mtu":1500},"eth1_data_disabled":false},"services":{
"snmp":{"v2":{"enabled":false},"v3":{"enabled":false}},"ping_watchdog":{"ena
bled":false},"http":{"https_port":443,"port":80},"remote_syslog":{"enabled":
false},"snmp_traps":{"enabled":false},"discovery":{"enabled":true,"broadcast
_protocols":{"lldp":true,"cdp":true,"mndp":true},"lldp_server":{"enabled":tr
ue}},"ntp":{"enabled":false}}}}
```

Changing the device's configuration

You must POST the full modified configuration back to the device, for example, change the IP address and device name, and channel, add “read” user, and POST the config back to the device (the example below formatted for Windows command line, please check the next chapter for password forming):

```
curl -X POST -H "Cookie: api_token=YOUR_API_TOKEN" -H "Content-Type:
application/json" -d
"{\"config\":{\"system\":{\"authentication\":{\"method\":\"local\"},\"device
_name\":\"Freemile60\",\"users":[{"username":"root","type":"admin","password
_hash":"$1$5HDT93ii$Z71a.vMeJR4CU65PiEkcX.","enabled":true},{"username":"rea
d","type":"read-
only","password_hash":"$1$/j82BsDc$eJWfg8tSgfSLpy8NzwyQt1","enabled":true}],
\"hostname\":\"freemileap\",\"reset_button\":{\"enabled\":true},\"timezone\"
:\"Europe/Riga\",\"device_location\":\"Office\",\"use_advanced_timezones\":t
rue},\"wireless\":{\"radios\":{\"wlan0\":{\"channel_width\":2160,\"ssid\":\"
freemile\",\"boresight_lock\":false,\"max_mcs\":12,\"passphrase\":\"passphra
se\",\"sta_profiles\":{\"enabled\":false},\"antenna_kit\":\"none\",\"channel
\":5,\"mode\":\"ap\",\"ptp\":false,\"security\":\"wpapsk\"}}},\"network\":{\
"traffic_control\":{\"limit_download\":{\"enabled\":false},\"limit_upload\":
{\"enabled\":false}},\"poe_out_enabled\":false,\"failover\":{\"enabled\":fal
se},\"wan\":{\"ipv6\":{\"enabled\":false},\"data_vlan\":{\"enabled\":false},
\"ipv4\":{\"enabled\":true,\"prefix\":24,\"gateway\":\"192.168.1.254\",\"dns
_servers\":[\"1.1.1.1\",\"1.0.0.1\"],\"ipaddr\":\"192.168.1.2\"},\"alternati
ve_local_ip\":{\"enabled\":true,\"ipaddr\":\"169.254.1.1\"},\"mgmt_vlan\":{\
"enabled\":false},\"ip_mode\":\"static\"},\"dhcp_snooping\":{\"option_82_inj
ection\":{\"enabled\":false},\"block_rogue_servers\":false},\"general\":{\"m
ax_fdb_entries\":0,\"ageing_time\":300,\"mtu\":1500},\"eth1_data_disabled\":
false},\"services\":{\"snmp\":{\"v2\":{\"enabled\":false},\"v3\":{\"enabled\
":false}},\"ping_watchdog\":{\"enabled\":false},\"http\":{\"https_port\":443
,\"port\":80},\"remote_syslog\":{\"enabled\":false},\"snmp_traps\":{\"enable
d\":false},\"discovery\":{\"enabled\":true,\"broadcast_protocols\":{\"lldp\"
:true,\"cdp\":true,\"mndp\":true},\"lldp_server\":{\"enabled\":true}},\"ntp\
":{\"enabled\":false}}}}" "http://192.168.1.1/cgi.lua/apiv1/config"
```

The successful execution will be represented with something like this:

```
{"status_msg":"Configuration saved, applying changes
now.","response":{"reboot_required":false,"keys_removed":{},"warnings":{},"k
eys_changed":{"wireless.radios.wlan0.channel":{"new":5,"prev":4},"system.dev
ice_name":{"new":"Freemile60","prev":"Freemile60AA"}},"keys_added":{}}}
```

Passwords for local user accounts

User passwords are formatted in the same manner as a password in a standard Linux /etc/passwd file:

```
$HASH_TYPE_ID$SALT$HASH
```

To change a plain-text password into a valid value that can be set in the config, you must use a command or library such as the one shown below to generate the password.

Here's an example using the openssl passwd command. It uses an MD5-based algorithm (ID of 1) as the HASH_TYPE_ID, a SALT made of a random string of up to 16 characters long, and generates a resulting password, including the hash value:

```
openssl passwd -1 -salt ydj/osBB admin $1$ydj/osBB$nNVJYoK1VJITbHKFUw1Rf.
```

- *-1*: This is the algorithm ID: MD5 has ID = 1
- *ydj/osBB*: This is the salt.
- *admin*: This is the password you want to hash.
- 1$ydj/osBB$nNVJYoK1VJITbHKFUw1Rf.: This will be the resulting value that you will set in your configuration JSON.

Statistics requests

The following types of stats objects can be fetched from the device via the API:

System information: including version, resources, temperatures, etc.

Command example:

```
curl  -X GET -H "Cookie: api_token=YOUR_API_TOKEN"
http://192.168.1.1/cgi.lua/apiv1/stats?type=system
```

Successful answer:


```
{"system":{"epoch_time":1732035422,"description":"","fw_version":"1.12.0 rev

54586","serial_no":"504320111717","cpu_usage_percent":1,"bootbanks":{"backup
":{"bootbank":1,"version":"1.12.0 rev

54430"},"active":{"bootbank":2,"version":"1.12.0 rev

54586"}},"memory":{"used":142405632,"usage":34,"total":419270656,"free":2768
65024},"name":"Freemile60","model":"FreeMile60","location":"Office","hostnam
e":"freemileap","clock_datetime":"2024/11/19

18:57:02","cpu_temp":35}}
```


epoch_time: seconds since the Unix epoch

Wireless information: Wireless 60 GHz AP and client/peer stats.

Command example:

```
curl  -X GET -H "Cookie: api_token=YOUR_API_TOKEN"
http://192.168.1.1/cgi.lua/apiv1/stats?type=wireless
```

Successful answer:


```
{"wireless":{"peers":{},"histogram_updated_at":1732028219,"radios":{"wlan0":
{"mac":"00:04:A6:81:AB:86","op_mode":"master","ssid":"freemile","radio_temp"
:37,"freq":64800,"boresight_lock":false,"antenna_kit":"none","channel_width"
:2160,"modem_temp":41,"name":"wlan0","channel":4,"tx_power":40,"ptp":false,"
channel_label":"4","security":"WPA2-PSK

(GCMP)"}}}}
```


Network information: Device IPs, management VLAN and data VLAN information, DNS servers, and similar.

Command example:

```
curl  -X GET -H "Cookie: api_token=YOUR_API_TOKEN"
http://192.168.1.1/cgi.lua/apiv1/stats?type=network
```

Successful answer:


```
{"network":{"zones":{"wan":{"mgmt_vlan":{"enabled":false},"data_bridge":{"ma
c":"00:04:A6:81:AB:84","ipv6":{"ip_address":["fe80::204:a6ff:fe81:ab84/64"]}
,"data_vlan":{"enabled":false},"ipv4":{"ip_address":["169.254.1.1/16","192.1
68.33.3/24"],"gateway":"192.168.33.254"},"interface":"br-
wan"}}},"dns":{"wan":["1.1.1.1","1.0.0.1"]},"hostname":"freemileap","dhcp_st
atus":{},"dns6":{"wan":{}}}}
```


Ethernet information: Ethernet interface statistics. The radio's ethernet ports are part of an internal switch.

Command example:

```
curl  -X GET -H "Cookie: api_token=YOUR_API_TOKEN"
http://192.168.1.1/cgi.lua/apiv1/stats?type=ethernet
```

Successful answer:


```
{"ethernet":{"ports":{"eth0":{"rx_bytes":86600,"tx_bytes":52753,"link":true,
"autoneg":true,"port":4,"tx_error":0,"rx_packets":794,"speed":100,"rx_error"
:0,"tx_packets":370,"duplex":"full"},"eth1":{"rx_bytes":0,"tx_bytes":128545,
"link":false,"autoneg":true,"tx_packets":1122,"data_disabled":false,"rx_pack
ets":0,"tx_error":0,"poe_out":false,"port":3,"rx_error":0}},"switch_mac":"00
:04:A6:81:AB:84"}}
```


FW upgrade

Upgrading your device's firmware has several basic steps:

1) The device asynchronously downloads the firmware from the user-specified specified URL
2) The device asynchronously flashes the firmware to the boot bank not currently in use
3) The device reboots, causing the new firmware to become active

Tracking of the status is available during the upgrade by GET requests to the /update route.

Please find the upgrade example below for the FW file located on the local HTTP server with local access and IP address *192.168.1.2*, firmware file *fm60.bin*.

1) POST to /login, please refer to chapter Authentication & authorization.
2) PUT the device firmware URL to /update, the device should start downloading the firmware:

```
curl --location --request PUT
"http://192.168.1.1/cgi.lua/apiv1/update" --header "accept:
application/json" --header "Cookie: api_token=YOUR_API_TOKEN;
token=YOUR_API_TOKEN" --header "Content-Type: application/json" --
data "{ \"firmware_url\": \"http://192.168.1.2/firmware/fm60.bin\"
}"
```

Successful answer:

```
{"status":"ok"}
```

3) Checking the firmware status：

```
curl --location "http://192.168.1.1/cgi.lua/apiv1/update" --header
"accept: application/json" -header "Cookie:
api_token=YOUR_API_TOKEN"
```

Successful answer：

```
{"status":"SUCCESS","url":"http://192.168.1.2/firmware/fm60.bin","
state":"FIRMWARE_DOWNLOAD","last_changed":1732030479}
```

During upload “last_changed” will increase.

4) By POST to /update the device will start asynchronously flashing the downloaded firmware file. Firmware images are flashed to the alternate boot bank, so this will not affect the normal operation of the device:

```
curl --location "http://192.168.1.1/cgi.lua/apiv1/update" --header
"accept: application/json" --header "Cookie:
api_token=YOUR_API_TOKEN" --header "Content-Type:
application/json" --data "{ \"reset\": false, \"force\": false }"
```

Successful answer:

```
{"status":"ok"}
```

5) For the firmware flashing status poll the device, wait for the firmware flashing process to complete

```
curl --location --request GET
"http://192.168.1.1/cgi.lua/apiv1/update" --header "accept:
application/json" --header "Cookie: api_token=YOUT_API_TOKEN" --
header "Content-Type: application/json" --data "{ \"reset\":
false, \"force\": false }"
```

Successful answer for finished flushing:

```
{"status":"COMPLETE","last_changed":1732030950,"comment":"Reboot
device for the new firmware to become
active.","state":"FIRMWARE_FLASHING"}
```

Do not reboot the FreeMile60 until the flashing is finished.

6) Rebooting the FreeMile60:

```
curl -X POST "http://192.168.1.1/cgi.lua/apiv1/reboot" -H "accept:
application/json" -H "Cookie: api_token=YOUR_API_TOKEN" -d ""
```

Successful answer:

```
{"status":"ok"}
```

7) POST to /login, please refer to chapter Authentication & authorization.
8) To verify if the new version is active please request system status, please refer to chapter Statistics requests.

Please don't hesitate to contact Techsupport@saftehnika.com if you have any questions.