# Summary

This document describes the REST Web Service interface exposed by the Zim printer through its HTTP interface.

# Requests destination

All HTTP requests will be sent to:

[http[s]://[zim]/rest/*function?parameters*](http://[zim]/rest/function?parameters)

Where [zim] will be returned by the “Directory server-1.3.1” API list “/listprinter.ashx” function.

Except for some mentioned exceptions, requests will be sent by GET HTTP method.

Requests sent through the local network must be performed using HTTP, requests sent through tromboning servers must be performed using HTTPS.

# Interface

Successful requests will return a 200 HTTP error with a possible answer in the body, depending of the function.

Unsuccessful requests will return a custom HTTP error code specified for each function

Text response bodies will be UTF-8 (RFC 3629) encoded and “text/plain; charset=UTF-8” MIME typed; other responses should use appropriate encoding and type (ex.: image/png…).

## Network

### Get

Get the network configuration.

Syntax: /getnetwork

Return value:

Json formatted string (RFC 4627):

* WiFi access point

{"topology":"p2p,

"medium":”wifi”,

"ssid":"*user\_defined\_ssid*"}

* WiFi network

{"topology":"network",

"medium":”wifi”,

"ssid":"*user\_network\_ssid*"}

* Ethernet network

{"topology":"network",

"medium":"eth",

["ipv4":{“user\_assigned\_address”:"*1st\_IPV4\_address*",

“user\_assigned\_gateway":”*1st\_gateway\_IPV4\_address*",

“user\_assigned\_mask":”*1st\_mask\_IPV4\_address*"}]}

Note:

In network Ethernet mode, only address, gateway and mask provided by the user will be returned (as opposed to those fetched from a DHCP server).

HTTP error code:

* 446: Printer busy
* 450: Update in progress

### Get ip configuration

Get the internet protocol configuration.

Syntax: /getnetworkip

Return value:

Json formatted string (RFC 4627):

{"ip":"*ip\_address*",

"gateway":"*gateway\_address*",

"dns":"*DNS\_server\_address – first if multiple DNS server address have been defined*",

“mac”: "*mac\_address*"}

HTTP error code:

* 446: Printer busy
* 450: Update in progress

### Get visible SSID

Get broadcasted SSID.

Syntax: /listssid

Return value:

Json formatted string (RFC 4627):

["ssid1", "ssid2","ssid3"… ]

HTTP error code:

* 450: Update in progress

### Configure

Set the network configuration.

Syntax: /setnetwork?*configuration*

*configuration*: URL encoded Json formatted string (RFC 4627):

* WiFi access point

{"topology":"p2p,

"medium":”wifi”,

"ssid":"*user\_defined\_ssid*",

"oldpassword":"*old\_password\_(to\_be\_left\_blank\_if\_not\_specified)*",

"password":"*password*"}

* WiFi network

{"topology":"network",

"medium":”wifi”,

"ssid":"*user\_network\_ssid*",

"password":"*password*"}

* Ethernet network

{"topology":"network",

"medium":"eth",

["ipv4":{“user\_assigned\_address”:"*1st\_IPV4\_address*",

“user\_assigned\_gateway":”*1st\_gateway\_IPV4\_address*",

“user\_assigned\_mask":”*1st\_mask\_IPV4\_address*"}]}

Note:

For network Ethernet mode, address, gateway and mask will be fetched from a DHCP server if they are not provided.

In case of configuration change, no response should be actually expected due to the possible loss of connectivity.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 446: Printer busy
* 450: Update in progress

### Reset

Reset the network configuration.

Syntax: /resetnetwork

Note:

The behavior is the same as the use of the network reset button of the printer. No response should be actually expected due to the loss of connectivity.

HTTP error code:

* 446: Printer busy
* 450: Update in progress

## Printer states

### Get

Printer parameters will be retrieved using to the “get” group function:

General syntax:

/get?*parameters*

Get printer information.

Syntax: /get?p=info

Return value:

Json formatted string (RFC 4627): {"ver": 1, "type": "*type*", "sn":"*serial number*", "extruder": number of extruder(s), “xmax”: *maximum printing breadth (in mm)*, “ymax”: *maximum printing depth (in mm)*, “zmax”: *maximum printing height (in mm)*, “name”: “*printer friendly name*”, “hostname”: “*hostname*”, “account”: “*account email*”}

Note:

The serial number of the printer will be its Ethernet MAC address (without colon).

The printer friendly name is initially blank and given at printer activation.

#### Printer status

Get printer status

Syntax: /status

Return value:

Json formatted string (RFC 4627) :

{"status":"*status*",

["p":*percentage*],

["d":*duration*],

["eXtended\_parameters":*eXtended\_parameters*],

["time-lapse": *time-lapse\_information*]}

*status*: “upgrading”|”to\_be\_connected”|”idle”|”slicing”|”sliced”|”canceling”|”printing”|”printed”|”loading\_left”|”loading\_right”|”unloading\_left”|”unloading\_right”|”recovery”|”usb\_connected”

*percentage* (optional): current state completion percentage

*duration* (optional): seconds remaining

*eXtended\_parameters* (optional): parameters depending upon state

*time-lapse* (optional): information about time-lapse video

Note:

For “sliced” state, extended parameters will be a Json formatted string (RFC 4627):

{"l\_temperature": *left\_temperature (in °C),* "l\_length": *left\_filament\_length\_needed (mm)*, "r\_temperature": *right\_temperature (in °C)*, "r\_length": *right\_filament\_length\_needed (mm)*}

If only one extruder is in use, the information concerning the other one won’t be provided.

For “printing” state, extended parameters will be a Json formatted string (RFC 4627) containing nozzles temperatures:

{"l\_temperature": *left\_temperature (in °C)*, "r\_temperature": *right\_temperature (in °C)*, "operation": ["heating"|"printing"|"lowering\_platform"]}

If only one extruder is in use, the information concerning the other one won’t be provided.

(to avoid any inconsistency after a power-cut - the G-code being stored in memory - the “sliced” state will be returned only if there’s actually a sliced model available)

The "usb\_connected” state is returned when the printer is connected to a computer through the USB cable. In this case, all user interfaces must return an information message preventing their usage.

If a slicing error happened, the printer will return in an “idle” state with extended parameter set to {"slicing\_error": "*error message*"}.

*time-lapse* is provided only for "idle" and "sliced" states and only if a time-lapse video is currently being generated or available:

* Currently being generated: "generation”
* Available: *time-lapse\_URL*

#### Acceleration

Get printer acceleration

Syntax: /get?p=acceleration&a=*axis*

* *axis*: x|y|z

Return value:

acceleration on the *axis* axis in mm/s²

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Cartridge information

Get cartridge information.

Syntax: /get?p=cartridgeinfo&v=*cartridge*

* *cartridge*: r|l (right/left)

Return value:

Json formatted string (RFC 4627) : {“type”:”*type of cartridge*”,

“material”:”*type of material*”,

“diameter”:*diameter*,

“color”:”*color”*,

“initial”:*initial material quantity in mm*,

“used”:*used material quantity in mm*,

“temperature”:*default extrusion temperature in Celsius degree*,

“setup”:*setup date (ISO 8601 – e.g.: 2014-01-01T15:12Z - see* [*http://en.wikipedia.org/wiki/ISO\_8601*](http://en.wikipedia.org/wiki/ISO_8601)*)*

}

Note:

If the cartridge is not present an empty structure (i.e. {}) will be returned. Colors are given by name, based on <http://www.w3.org/TR/css3-color/#svg-color>, or by 24 bits hexadecimal RGB code.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 446: Printer busy

#### Cold extrusion

Security mechanism that prevents extrusion below xxx °C sate.

Syntax: /get?p=coldextrusion

Return value:

“on” or “off”

Note:

This setting is shared by both extruders for dual head printer.

#### Current extruder

Get the current extruder.

Syntax: /get?p=extruder

Return value:

« l » or « r » (left/right)

Note:

For single head printer the function will always return l.

HTTP error code:

* 446: Printer busy

#### End stops status

Get end stops status.

Syntax: /get?p=endstop&axis=*axis*

* *axis*: xmin|xmax|ymin|ymax|zmin|zmax

Return value:

“on” or “off”

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Extrusion/reverse speed

Get extrusion and reverse speed

Syntax: /get?p=extrusionspeed

Return value :

extrusion speed in mm/s

Note:

For dual head printer, the returned extrusion speed will be the current extruder's one.

#### Filament state

Check if filament is loaded

Syntax: /get?p=filament&v=*filament*

* *filament:* l|r (left/right)

Return value:

“true” or “false”

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Head light

Get the blue light atop the printer head status

Syntax: /get?p=headlight

Return value:

“on” or “off”

#### Movement speed

Get movement speed

Syntax: /get?p=speed&a=*axis*

* *axis*: x|y|z

Return value :

speed on the *axis* axis in mm/s

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Offset

Get nozzle offsets adjustment.

Syntax: /get?p=offsetadjustment&axis=*axis*

* *axis*: x|y

Return value:

* nozzle offset adjustment in tenths of millimeters for the corresponding axis (integer)

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Position

Get printer head and platform position

Syntax: /get?p=position

Return value:

Json formatted string (RFC 4627) :

{"x": x head position in mm,

“y”: y head position in mm,

“z”: z platform position in mm}

#### Picture

Get a high definition picture from the onboard camera.

Syntax: /get?p=picture[?password=*xxx*]

* *password*: the password limiting the use of the onboard camera if the password protection is activated

Note:

The body of the response will contain a JPG picture of the platform.

HTTP error code:

* 436: Missing / Incorrect password

#### Remote control (SSH)

Get the printer SSH remote control state.

Syntax: /get?p=remotecontrol

Return value:

* “on” or “off”

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Remote control (tromboning)

Get the tromboning state of the printer.

Syntax: /get?p=tromboning

Return value:

* “on” or “off”

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Slicer preset

Get current slicer preset.

Syntax: /get?p=slicerpreset

Return value:

Slicer preset identifier (string)

#### Temperature

Get temperatures of extruder(s) or heater bed.

Syntax: /get?p=temp&<e|h>[&v=*extruder*]

* “e” for extruder, “h” for heater bed
* *extruder:* l|r (left/right) (for dual head extruder temperature retrieving)

Return value:

Temperature (integer in Celsius degree)

Note:

-1 will be returned if the asked extruder or heater bed is not present.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Tromboning

Get printer tromboning status.

Syntax: /get?p=tromboning

Return value:

* “on” or “off”

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Statistics

Get printer statistics status.

Syntax: /get?p=stats

Return value:

* “on” or “off”

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Strip led

Get strip led status.

Syntax: /get?p=stripled

Return value:

“on” or “off”

#### Upgrade

Get printer upgrade status.

Syntax: /get?p=upgrade

Return value:

* “on” or “off”

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Video broadcasting

Get the state of the onboard camera broadcasting engine.

Syntax: /get?p=video

Return value:

current mode (“3gp”/”mp4”/etc.) or “off”.

### Set

Printer parameters will be set using to the “set” group function:

General syntax:

/set?*parameters*

#### Acceleration

Set printer acceleration

Syntax: /set?p=acceleration&a=*axis*&v=*value*

* *axis*: x|y|z
* *value*: acceleration in mm/s² (between xxx and yyy)

Default value :

* X : xxx mm/s²
* Y : yyy mm/s²
* Z : zzz mm/s²

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Camera password

Set or reset the password limiting the use of the onboard camera.

Syntax: /set?p=password&v=*password*[&o=*old-password*]

* *v*: the new password, left blank to cancel password protection
* *o*: the previous password ; can be not provided or left blank if the password protection is not active

Note:

The password is left blank at first use and after each network reset.

Passwords must be URL encoded.

Current video broadcasting, if any, will be stopped if the new password is not blank.

HTTP error code:

* 432: Missing parameter
* 436: Missing / Incorrect password

#### Cold extrusion

Enable or disable the security mechanism that prevents extrusion below xxx °C.

Syntax: /set?p=coldextrusion&v=<on|off>

Default value:

* off

Note:

This setting concern both extruders for dual head printer.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Current extruder

Set the current extruder.

Syntax: /set?p=extruder&v=*extruder*

* *extruder:* l|r (left/right)

Default value:

* l (left)

Note:

Subsequent head movement will perform the offset adjustment needed to center the select extruder nozzle to the current position.

For single head printer, only l value will be accepted.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 446: Printer busy

#### Extrusion/reverse speed

Set extrusion and reverse speed

Syntax: /set?p=extrusionspeed&v=*value*

* *value*: speed in m/s (between xxx and yyy)

Default value:

5 mm/s

Note:

For dual head printer, the speed will be set for the current extruder.

Extrusion and reverse will be performed at the same speed.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Movement speed

Set movement speed

Syntax: /set?p=speed&a=*axis*&v=*value*

* *axis*: x|y|z
* *value*: speed in mm/s (between xxx and yyy)

Default value :

* X : xxx mm/s
* Y : yyy mm/s
* Z : zzz mm/s

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Motors of

Set motors off until next move (M84 G-code)

Syntax: /set?p=motor&v=off

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Offset

Set nozzles offset adjustment.

Syntax: /set?p=offsetadjustment&axis=*axis*&adjustment=*adjustment*

* *axis*: x|y
* *adjustment*: nozzle offset adjustment in tenths of millimeters for the corresponding axis (integer)

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Printer name

Set printer friendly name.

Syntax: /set?p=name&name=*name*

* *name*: printer friendly name

Note:

Calling functions are responsible to maintain coherence between the local name and the SSO stored name (/addprinter.ashx - Directory server).

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Remote control

Set the ability of the printer to be remotely controlled.

Syntax: /set?p=remotecontrol&v=<on|off>

Default value:

* on

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Remote control

Enable/disable printer SSH remote control.

Syntax: /set?p=remotecontrol&v=<on|off>

Default value:

* off

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Slicer preset

Set current slicer preset.

Syntax: /set?p=slicerpreset&id=*id*

* *id*: preset slicer identifier (see Slicer / List function)

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Slicer parameter

Modify current slicer parameter.

Syntax: /set?p=slicerparameter[&density=*density*][&skirt=*0|1*][&raft=*0|1*][&support=*0|1*]

* *density*: fill density in percentage (between 0 and 100)

Note:

This function will modify the current preset slicer parameters (changes will be lost if the preset is changed).

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Statistics

Enable/disable statistics sending by the printer.

Syntax: /set?p=stats&v=<on|off>

Default value:

* on

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Switch on/off strip led

Switch on or off the strip led above the platform.

Syntax: /set?p=stripled&v=<on|off>

Default value:

* off

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Switch on/off head light

Switch on or off the blue light atop the printer head

Syntax: /set?p=headlight&v=<on|off>

Default value:

* off

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Temperature

Set temperatures of extruder(s) and heater bed.

Syntax: /set?p=temp&<e|h>&v=*value*

* “e” for extruder, “h” for heater bed
* *value*: temperature in °C (between 100 and 250)

Default value:

* Extruder: 200 °C
* Heater bed: 100 °C

Note:

For dual head printer, the temperature will be set for the current extruder.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 446: Printer busy

#### Tromboning

Enable/disable tromboning.

Syntax: /set?p=tromboning&v=<on|off>

Default value:

* on

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Upgrade

Enable/disable printer upgrades.

Syntax: /set?p=upgrade&v=<on|off>

Default value:

* on

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Video broadcasting

Start or stop the onboard camera broadcasting engine.

Syntax: /set?p=video&v=<on|off>&m=<*ffmpeg command line*>[&s=*password*]

* *m*: the video mode
* *s*: the password limiting the use of the onboard camera if the password protection is activated

Return value:

None.

Note:

While performing time-lapse capture, the server shouldn’t be interrupt nor its parameters modified.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 436: Missing / Incorrect password

## Printer movements

### Homing

Reset head and platform position to 0.

Syntax: /home[?a=*axis*]

* *axis*: x|y|z

Note:

If the *axis* parameter is not provided the homing will occur for all axes in one go.

HTTP error code:

* 433: Incorrect parameter
* 446: Printer busy

### Move

Move printer head and platform

Syntax: /move?a=*axis*&v=*value*

* *axis*: x|y|z
* *value*: distance in mm (between -150 and 150)

Note:

Value can be decimal (e.g.: -1.5)

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 446: Printer busy

## Extrusion

### Extrude

Syntax: /extrude[?v=*value*]

* *value*: filament length to extrude in mm (between xxx and yyy)

Default value:

* 5 mm

Note:

For dual head printer, extrusion will be performed by the current extruder.

Reverse will be performed by entering a negative value.

HTTP error code:

* 433: Incorrect parameter
* 446: Printer busy

## Cartridges

### Load

Syntax: /load?v=*cartridge*

* *cartridge*: r|l (right/left)

Note:

Procedure will vary depending upon filament material type:

* PLA / ABS : 500 mm/min for the first 5 cm then 1000 mm/min
* PVA : 100 mm/min for the first 5 cm then 500 mm/min

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 434: Already loaded / unloaded
* 438: Cartridge missing (before complete action)
* 439: Filament missing (after action completed)
* 446: Printer busy

### Unload

Syntax: /unload?v=*cartridge*

* *cartridge*: r|l (right/left)

Note:

Procedure will vary depending upon filament material type:

* PLA / ABS : heats the nozzles up to the first layer extrusion temperature; extrude 10 mm at 150 mm/min; rewind at 1000 mm/min
* PVA : heats the nozzles up to the first layer extrusion temperature; extrude 10 mm at 50 mm/min; rewind at 500 mm/min

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 434: Already loaded / unloaded
* 446: Printer busy

### Prime

Will perform priming (50mm extrusion to ensure proper cartridge operations and clean out of the previous filament) for a new cartridge. “First” priming will home the printer head, center it, move down the platform and extrude 50mm at the temperature specified in the cartridge. “Subsequent” priming, will only extrude 50mm at the requested temperature (avoiding any head movement). The extrusion speed will depend upon the filament material type (PLA/ABS: 150 mm/min; PVA: 50 mm/min).

Syntax: /prime?v=*cartridge*&t=*type*

* *cartridge*: r|l (right/left)
* *type*: f|s (first/subsequent)

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 439: Filament missing
* 446: Printer busy

### Raise platform

Syntax: /raiseplatform

HTTP error code:

* 446: Printer busy

## Print list

The print list will contain a short list (10 to 20) of pre-sliced models stored in the memory of the printer to provide a “fast-print” functionality.

### List

List models in the print list

Syntax: /listmodel

Return value:

Json formatted string (RFC 4627) containing an array listing all models information

{"ver": 2,

"models": [{“mid”:*model identifier*,

“name”:

{"*lang1*”: "*name1*” *(50 char. max.)*,

"*lang2*”: "*name2*” *(50 char. max.)*,

…},

“description”:

{"*lang1*”: "*description1*” *(255 char. max.)*,

"*lang2*”: "*description2*” *(255 char. max.)*,

…},

“duration”: *duration (in seconds)*,

“c1”:”*color”*,

“l1”: length *(in mm)*,

“c2”:”*color”*,

“l2”: length *(in mm)*,

“picture”:[“*picture\_url\_1*”, “*picture\_url\_2*”…]},

…

]}

Note:

Languages identifiers will follow the ISO 639-1 norm. Filament needed length are ordered by objet order within AMF file. Pictures URL will call Get picture function. Colors are given by name, based on <http://www.w3.org/TR/css3-color/#svg-color>, or by 24 bits hexadecimal RGB code.

HTTP error code:

* 446: Printer busy

### Get picture

Return a picture of a model from the print list.

Syntax: /getpicture?id=*mid*&p=*picture\_number*

* *mid*: model identifier
* *picture\_number*: picture number

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 446: Printer busy
* 448: Unknown model
* 449: Unknown picture

### Pre-sliced print

Print a pre-sliced model from the print list.

Syntax: /preslicedprint?id=*mid*[&filament=<*straight*|*crossover*>]

* *mid*: model identifier (see “list” function)
* “filament” default value: straight (right filament for single color models)

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 442: Left filament missing
* 443: Right filament missing
* 444: Not enough left filament
* 445: Not enough right filament
* 446: Printer busy

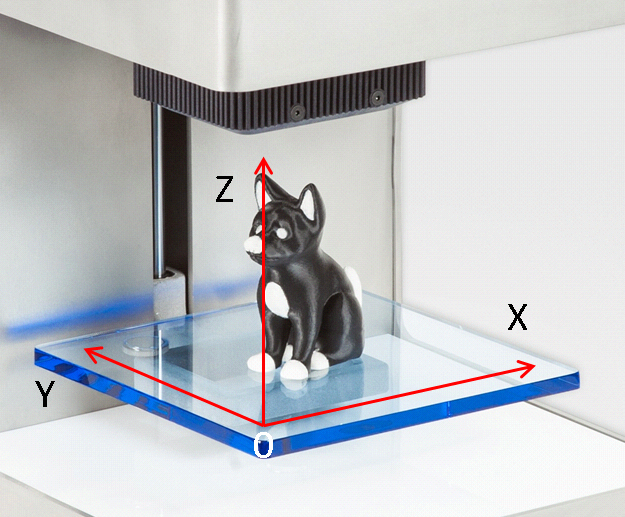
## Platform

The platform is the print tray.

Several models can be placed on it; they can be modified (position and scale) and be rendered as an all.

Models must not exceed the printing volume limits.

Platform axes present themselves in the following manner:



All coordinates (about X, Y or Z axis) must fit [0, 150[ range (in mm).

#### Get 3D file

Get STL, OBJ or AMF 3D file of a specified model.

Syntax: /getmodel?id=*id*

* *id*: the model identifier (see “list” function)

Note:

The 3D file (either AFM, OBj or STL depending on the source file) will be send zipped with an “application/zip” MIME type.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Level

This function will “level” all models on the platform so their lowest point will align to the 0 Z-axis.

Syntax: /level

Note:

This function will automatically be called before printing.

#### List

Get platform models list

Syntax: /platformmodel

Return value:

Json formatted string (RFC 4627) containing an array listing all models information

[{“id”:*identifier*,

“xpos”:*x\_position(minimum in mm – must be positive)*,

“ypos”:*y\_position(minimum in mm – must be positive)*,

“zpos”:*z\_position(minimum in mm – must be positive)*,

“xrot”:*x\_rotation(in degree)*,

“yrot”:*y\_rotation(in degree)*,

“zrot”:z*\_rotation(in degree)*,

“s”:*scale* (in percent),

“color”: Json formatted string (RFC 4627) linking color ranks and extruders number *“[first color extruder number, second color extruder number, N color extruder number]”* ; eg: “[0, 1]”

}…

]

#### Remove

Remove a specified model.

Syntax: /removemodel?id=*id*

* *id*: the model identifier (see “list” function)

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Set position, rotation, scale and color

Set current position, rotation and color of a given model.

Syntax: /setmodel?id=*id*[&xpos=*xpos*][&ypos=*ypos*][&zpos=*zpos*][&xrot=*xrot*][&yrot=*yrot*][&zrot=*zrot*][&s=*scale*][&c=*color*]

Note:

* *id*: the model identifier (see “list” function)
* *xpos*: minimum x position of the model (in mm)
* *ypos*: minimum y position of the model (in mm)
* *zpos*: minimum z position of the model (in mm)
* *xrot*: rotation about x axis (in degree)
* *yrot*: rotation about y axis (in degree)
* *zrot*: rotation about z axis (in degree)
* *scale*: scale (in percent)
* *color*: Json formatted string (RFC 4627) linking color ranks and extruders number “[first color extruder number, second color extruder number, N color extruder number]” ; eg.: “[0, 1]”

The scale factor application must respect the limits of the printing area. The 433 error is returned otherwise.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

#### Upload

Upload a model on the platform

Syntax: /upload

Return value:

Json formatted string (RFC 4627) containing the identifier of the model

{“id”:*identifier*}

Note:

The store function will be called by POST method with the following parameters:

or

* *f*: 3D file of the model (GCODE, STL, OBJ or AMF)

or

* *s1*: first 3D file part of the model (STL)
* *s2*: second 3D file part of the model (STL)

Each 3D file mustn’t exceed 100Mo uncompressed.

Both parts of a two parts model must be uploaded at the same time. Parts of a two parts model will be merged and loaded as if the model has been uploaded like a single AMF file.

The upload function will automatically scale down the model if it exceeds the printing volume limits and “level” the model so its lowest point will align to the 0 Z-axis.

In version 1.0 of the REST web service, upload of a new model will provoke the deletion of the previous one so that only one model could be print at a time.

Files ending with a .GCODE extension will be integrally parsed in order to verify that each line is syntactically correct (an error, will be returned otherwise). Size will be checked in order to verify the model printability. Correct files will be analyzed to determine the number of extruders used and associated temperatures. Printer state will be set to “sliced”, with eXtended\_parameters filed accordingly.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 440: Incorrect file format
* 450: File to big
* 455: Pre-sliced model too big

#### Upload without resizing

Upload a model on the platform without adapt its size

Syntax: /uploadnoresize

Return value:

Json formatted string (RFC 4627) containing the identifier and information about of the model

{“id”:*identifier,*

“xsize”: *model breadth (in mm)*,

“ysize”: *maximum model depth (in mm)*,

“zsize”: *maximum model height (in mm),*

*“scalemax“*: *maximum model scale (according to the printing volume - in %)*}

Note:

The store function will be called by POST method with the following parameters:

or

* *f*: 3D file of the model (STL, OBJ or AMF –GCODE will be rejected)

or

* *s1*: first 3D file part of the model (STL)
* *s2*: second 3D file part of the model (STL)

Each 3D file mustn’t exceed 100Mo uncompressed.

Both parts of a two parts model must be uploaded at the same time. Parts of a two parts model will be merged and loaded as if the model has been uploaded like a single AMF file.

The upload function will automatically scale down the model if it exceeds the printing volume limits and “level” the model so its lowest point will align to the 0 Z-axis.

In version 1.0 of the REST web service, upload of a new model will provoke the deletion of the previous one so that only one model could be print at a time.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter
* 440: Incorrect file format
* 450: File to big

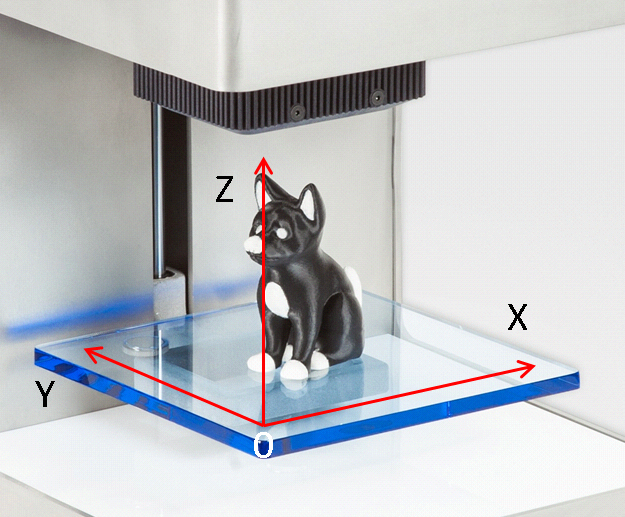
## Rendering

The platform is the print tray.

Several models can be placed on it; they can be modified (position and scale) and be rendered as an all.

Models must not exceed the printing volume limits.

Platform axes present themselves in the following manner:



All coordinates (about X, Y or Z axis) must fit [0, 150[ range (in mm).

#### Get 3D file

Get STL, OBJ or AMF 3D file of a specified model.

Syntax: /getmodel?id=*id*

* *id*: the model identifier (see “list” function)

Note:

The 3D file (either AFM, OBj or STL depending on the source file) will be send zipped with an “application/zip” MIME type.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

## Slicer

### Cancel

Cancel current slicing

Syntax: /cancelslicing

Note:

Cancel the current slicing (and clean up every resources).

HTTP error code:

* 454: No current slicing

### List

List slicer presets.

Syntax: /slicerlistpreset

Return value:

Json formatted string (RFC 4627) containing an array listing all models information

{{“id”:*preset identifier*, “name”: ”*name*” *(30 char. max.)*}

### Slice

Slice models present on the tray print.

Syntax: /slice

HTTP error code:

* 441: Platform empty
* 446: Printer busy

## Printing

### Calibration

Print the calibration model.

Syntax: /printcalibration

HTTP error code:

* 439: Filament missing
* 446: Printer busy

### Print

Launch printing of the G-code previously sliced.

Syntax: /platformprint[&filament=<*straight*|*crossover*>]

Default value:

* straight (right filament for single color models)

HTTP error code:

* 442: Left filament missing
* 443: Right filament missing
* 444: Not enough left filament
* 445: Not enough right filament
* 446: Printer busy
* 452: No sliced model

### Suspend

Suspend current printing

Syntax: /suspend

Note:

Suspending the printing will lower the platform, retract the filament and cool nozzles down.

HTTP error code:

* 437: No current printing

### Resume

Resume current printing

Syntax: /resume

Note:

Resume the previously suspended printing.

HTTP error code:

* 437: No current printing

### Cancel

Cancel current printing

Syntax: /cancel

Note:

Canceling the current printing will lower the platform, retract the filament and cool nozzles down.

HTTP error code:

* 437: No current printing

## Miscellaneous

### Logging

Statistics logging interface.

Syntax: /stat

Note:

The login function will be called by POST method with the following URL-encoded parameters:

* *printersn*: printer serial number
* *category*: name supplied for the group of objects to be tracked
* *action*: string that is uniquely paired with each category, and used to define the type of user interaction
* *label (optional)*: optional string to provide additional information on the event data
* *value (optional)*: integer that can be used to provide numerical data about the event

Messages will forwarded by POST to the URL:

<https://stat.service.zeepro.com/log.ashx>

Requests will have to be run asynchronously and error trapped silently.

HTTP error code:

* 432: Missing parameter
* 433: Incorrect parameter

### Send G-code

Send G-code to the printer.

Syntax: /gcode[?v=*gcode*]

* *gcode*: URL-encoded G-code if GET request (see below)

Note:

If the G-code length is more than 1024 characters, the request must be sent by POST, with a parameter named “v” containing the URL-encoded G-code.

G-code is case-sensitive.

Return value:

If G-code is sent by GET method: buffered responses for each instruction, separated by a tabulation character

If G-code is sent by POST method: nothing

### Send G-code file

Send a G-code file to the printer.

Syntax: /gcodefile

Note:

This function allows to send a G-code file to the printer by POST. If the request is made by GET, a HTML form letting an HTTP client to select a local file to be specified will be sent in return.

---

*The following feature is still in discussion (depending upon the general responsibility of the filament compensation):*

*Before printing, the G-code will be analyzed in order to determine:*

* *how many extruders will be used*
* *there temperature*

*These information will be used to:*

* *heat nozzle(s)*
* *compensate the 20mm retraction (compensation should only be performed with hot nozzles to avoid cold extrusion)*

*After printing, a 20mm retraction will be performed for each used extruders.*

---

HTTP error code:

* 442: Left filament missing
* 443: Right filament missing
* 446: Printer busy

### Reboot

Reboot the printer.

Syntax: /reboot

### Shut down

Shut the printer down

Syntax: /shutdown

## Model Storage

### Store STL

Store a STL model on the printer.

Syntax: /libstorestl

* *name*: the model name
* *file1*: the model (100MB limit)
* *file2*: the second part of the model (optional – 100MB limit)

Note:

This function allows to store a one or a two file STL on the printer by POST.

---

Get the rendering image from /sliceupload/preview\_ajax (

rho = ZIMAPI\_VALUE\_DEFAULT\_RHO; 0

delta = ZIMAPI\_VALUE\_DEFAULT\_DELTA; 0

theta = ZIMAPI\_VALUE\_DEFAULT\_THETA; 0

color\_right = 'PRINTERSTATE\_VALUE\_DEFAULT\_COLOR'; white

color\_left = 'PRINTERSTATE\_VALUE\_DEFAULT\_COLOR'; red)

---

HTTP error code:

* 432: Missing parameter
* 443: Incorrect parameter
* 450: File too big
* 456: Disk full

### Store G-code

Store a G-code file on the printer.

Syntax: /libstoregcode

* *name*: the model name

Note:

This function allows to store the last printed G-code on the printer.

---

Get the last G-code printed and the screen capture of the end of printing in the /tmp folder.

---

HTTP error code:

* 432: Missing parameter
* 443: Incorrect parameter
* 456: Disk full
* 457: G-code file not found
* 458: Image file not found

### List STL

Get the STL models list.

Syntax: /libliststl

Return value:

Json formatted string (RFC 4627) containing an array listing all models information

[{id: id, name: name, imglink: link}, …]

### List G-code

Get the G-code models list.

Syntax: /liblistgcode

Return value:

Json formatted string (RFC 4627) containing an array listing all models information

[{id: id, name: name, imglink: link}, …]

### Rename STL

Rename the STL model on the printer.

Syntax: /librenamestl?id=id&name=name

* *id*: id
* *name*: the model name

HTTP error code:

* 432: Missing parameter
* 443: Incorrect parameter
* 448: Unknown model

### Rename G-code

Rename the G-code model on the printer.

Syntax: /librenamegcode?id=id&name=name

* *id*: id
* *name*: the model name

HTTP error code:

* 432: Missing parameter
* 443: Incorrect parameter
* 448: Unknown model

### Delete STL

Delete the STL model from the printer.

Syntax: /libdeletestl?id=id

* *id*: id

HTTP error code:

* 432: Missing parameter
* 443: Incorrect parameter
* 448: Unknown model

### Delete G-code

Delete the G-code model from the printer.

Syntax: /libdeletegcode?id=id

* *id*: id

HTTP error code:

* 432: Missing parameter
* 443: Incorrect parameter
* 448: Unknown model

### Print STL

Load the STL model on the printer.

Syntax: / libprintstl?id=id

* *id*: id

HTTP error code:

* 432: Missing parameter
* 443: Incorrect parameter
* 448: Unknown model
* Slicer error codes

### Print G-code

Print the G-code model on the printer.

Syntax: / libprintgcode?id=id

* *id*: id

HTTP error code:

* 432: Missing parameter
* 443: Incorrect parameter
* 448: Unknown model
* Printer error codes

# HTTP error code

* 432: Missing parameter
* 433: Incorrect parameter
* 434: Already loaded / unloaded
* 436: Missing / Incorrect password
* 437: No current printing
* 438: Left cartridge missing
* 439: Right cartridge missing
* 440: Incorrect file format
* 441: Platform empty
* 442: Left filament missing
* 443: Right filament missing
* 444: Not enough left filament
* 445: Not enough right filament
* 446: Printer busy
* 447: Print list full
* 448: Unknown model
* 449: Unknown picture
* 450: File too big
* 451: Model too big
* 452: No sliced model
* 453: Remote control not allowed
* 454: No current slicing
* 455: Pre-sliced model too big
* 456: Disk full
* 457: G-code file not found
* 458: Image file not found
* 499: WIP