# Ember Web Portal API

By Matt Hubert, Jason Lefley, & Carl Minden

### Overview

This document describes the communication protocol between the web service, the printer, the web client, and the slicer. It is designed to be an exhaustive list of the web service <> printer requirements, but there is additional information sent from the printer for use by the Spark API that is not documented here. It touches on the interaction between the web service and web client, however is not exhaustive in that respect, and it also describes the protocol for a slicer and repackager.

### Pre-requsities

When POSTing to endpoints that take JSON as the request format, don't forget to add the header Content-Type: application/json

## Authentication

### Server Authentication

All requests except for POST /api/v1/printers/registration\_code require authentication. Authentication can be set by adding thePrinter-Auth-Token header to all of your HTTP requests with the value set to the auth\_token for the Printer or Web Client depending on the request

### Faye Authentication

Faye authentication\_token can be set using a client extension, for example:

class ClientAuth

def initialize(token)

@authentication\_token = token

end

def outgoing(message, callback)

# Again, leave non-subscribe messages alone

unless message['channel'] == '/meta/subscribe'

return callback.call(message)

end

# Add ext field if it's not present

message['ext'] ||= {}

# Set the auth token

message['ext']['authentication\_token'] = @authentication\_token

# Carry on and send the message to the server

callback.call(message)

end

end

client.add\_extension(ClientAuth.new(<AUTH TOKEN>))

More information here:   
[http://faye.jcoglan.com/ruby/clients.html](http://clients.bitmatica.com/redirect?http://faye.jcoglan.com/ruby/clients.html)

### Slicer and Repackager Authentication

Assuming a Slicer/Repackager key of "test123"

The "Authorization" Header must be set to "Token token=\"test123\""

### Setup

POST /api/v1/printers/registration\_code Printer

Called when a printer registers itself to the server. If the printer provides an authentication token, the printer will be re-associated with its account. If the printer does not provide an authentication token, it will be given a new id, a new authentication token, as well as a registration code it should display on the screen. The firmware and serial\_number fields in the request will be set for the printer, and a registration code is returned with every response until the registration has been completed for the printer with a given auth token.

Example Request: auth\_token=asdf&serial\_number=4321&firmware=2

Example Response: {"printer\_id": 1, "registration\_code" : "4321", "auth\_token": "asdf"}

Request Parameters:

| Parameter | Description |
| --- | --- |
| auth\_token | Authentication Token of the printer if it has already been registered |
| serial\_number | Serial Number of the printer, will be saved to the Printer object in the database |
| firmware | Firmware Version of the printer, will be saved to the Printer object in the database |

SUB /printers/ID/users Printer

Will notify the printer when a user has added the printer to his or her account.

POST /api/v1/printers/register Web Client

The client types the registration code into the web interface to register with a new printer.

Example Request: {"registration\_code": 4321}

Example Response: [{"printer\_id": 1, "status": "confirmed"}]

Request Parameters:

| Parameter | Description |
| --- | --- |
| registration\_code | Registration Code of the printer the user is attempting to register |

PUB /printers/ID/users Server

The server notifies the printer of a successful user registration.

Example Payload: {"registration": "success", "type": "primary"}

### Printer Status

POST /api/v1/printers/ID/status Printer  
The printer should POST to the server its latest status, including any errors that may be occurring. Status updates should also be sent periodically to inform the Server that the Printer is "alive".

Example Request: {"data": {"state": "state\_name", "change": "entering", "ui\_sub\_state": "sub\_state", ...}}

Example Response: None

Request Parameters:

| Parameter | Description |
| --- | --- |
| data.state | the current state of the printer's state machine |
| data.change | whether the printer is "entering" or "leaving" that state, or "none" if there has been no state change |
| data.ui\_sub\_state | a sub-state in which a different UI is shown, or "NoUISubState" if none |
| data.is\_error | true if and only if an error is being reported |
| data.error\_code | the last Ember error code |
| data.errno | the last Linux errno |
| data.error\_message | the last error message |
| data.layer | the current layer number (1-indexed) when printing, or 0 when not printing |
| data.total\_layers | the total number of layers in the current print, or 0 when not printing |
| data.seconds\_left | the estimated remaining print time in seconds |
| data.temperature | the current temperature inside the printer, in degrees Celsius |
| data.job\_id | id of the job being printed, will either be an id given by the server or an id created locally on the printer for local jobs |
| data.job\_progress | the fraction of the print completed, when printing |
| data.print\_rating | indicates the result of a print as either "successful" or "failed" |

PUB /printers/ID/frontend Server  
Server passes the "data" sent from the Printer in the /status request to the Web Client

### Command Interface

#### Overview

The printer and web service communicate via a series of commands. The server issues outbound commands to the printer as pub/sub notifications. Because the pub/sub system is asynchronous and not bi-directional, the printer cannot return a direct response to an outbound command. Therefore, the printer must make an acknowledgement request to the server containing the response to the command. Every publish will include a command\_token, which is sent back with the acknowledgement so the acknowledgement can be associated with the appropriate command

SUB /printers/ID/command Printer

The printer listens for commands from the service.

POST /api/v1/printers/ID/command Web Client  
Web Client asks the Server to send the specified Printer a command

Example Request: command=command&command\_token=12345

Example Response: None

| Parameter | Description |
| --- | --- |
| command | the name of the command you want to send to the printer |
| command\_token | a unique id which the printer will use for its acknowledgement of receipt of the command |
| settings | used for some commands that require settings, like "settings" and "print\_data" |
| package\_url | link to new firmware, used by the "firmware\_upgrade" command |
| file\_url | used by the "print\_data" command |

List of valid commands:

| Command | Description |
| --- | --- |
| print\_data | ask the printer to print a specific file |
| settings | sends a set of printer or print settings for the printer to use |
| request\_settings | asks the printer to send back its current settings |
| logs | asks the printer to upload its logs so they can be sent in with feedback |
| pause | pauses the current print |
| resume | resumes the current print |
| cancel | cancels the current print |
| reset | resets the printer |

PUB /printers/ID/command Server  
The server sends a command to the printer. Additional command specific parameters listed above may be specified in addition to the command and command token.

Example Payload: {"command": "<command>", "command\_token": "<token>", "<command specific parameter>": "<parameter value>", ...}

POST /api/v1/printers/command/<command\_token> Printer  
The printer acknowledges that it has received, successfully handled, or failed to handle a given command.

Example Request: {"data": {"command": "<command name>", "message": "<message containing request specific data (like the logs url)>", "state": "received"|"completed"|"failed", "settings": "<settings are sent if the "request\_settings" command is sent"}}

Example Response: None

PUB /printers/ID/frontend Server  
Server passes the command acknowledgement sent from the Printer in the /command request through to the Web Client

Example Payload: {"command": "<command name>", "command\_token": "12345"}

### Command Examples

#### Set Printer Settings

PUB /printers/ID/command Server  
The server commands the printer to apply the specified settings.

Example Payload: {"command": "settings", "settings": {"JobName": "my print", "LayerThicknessMicrons": 25, "FirstExposureSec": 5.0}, "command\_token": "<token>"}

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges receipt of the command.

Example Request: {"command": "settings", "message": null, "state": "received"}

Example Response: None

POST /api/v1/printers/command Printer  
The printer acknowledges that the settings have been successfully applied.

Example Request: {"command": "settings", "command\_token": "<token>", "message": null, "state": "completed"}

Example Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that the settings could not be successfully applied.

Example Request: {"command": "settings", "message": "<error message>", "state": "failed"}

Example Response: None

#### Request Settings

PUB /printers/ID/command Server  
The server commands the the printer to respond with its current settings.

Example Payload: {"command": "request\_settings", "command\_token": "<token>"}

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges receipt of the command.

Example Request: {"command": "request\_settings", "message": null, "state": "received"}

Example Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that it has successfully queried its current settings and includes them in the acknowledgment.

Example Request: {"command": "request\_settings", "message": {"JobName": "my print", "LayerThicknessMicrons": 25, "FirstExposureSec": 5.0}, "state": "completed"}

Example Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that the settings could not be successfully queried.

Example Request: {"command": "settings", "message": "<error message>", "state": "failed"}

Example Response: None

#### Upload logs

PUB /printers/ID/command Server  
The server commands the printer to upload an archive of its logs

Example Payload: {"command": "logs", "command\_token": "<token>"}

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges receipt of the command.

Example Request: {"command": "logs", "message": null, "state": "received"}

Example Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that it has successfully uploaded an archive of its log files and includes the location of the upload in the acknowledgment.

Example Request: {"command": "logs", "message": {"url": "http://s3.backed.cloudfront/logs.tar.gz"}, "state": "completed"}

Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that it could not successfully uploaded an archive of its log files.

Request: {"command": "logs", "message": "<error message>", "state": "failed"}

Response: None

#### Load print data

PUB /printers/ID/command Server  
The server commands the printer to download the specified print data file and apply the specified settings.

Example Payload: {"command": "print\_data", "file\_url": "http://s3.backed.cloudfront/link", "settings": {"Settings": {"JobName": "my print", "JobID": 123, "LayerThicknessMicrons": 25, "FirstExposureSec": 5.0}}, "command\_token": "<token>"}

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges receipt of the command.

Request: {"command": "print\_data", "message": null, "state": "received"}

Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that it has successfully downloaded and begun processing the specified print file and settings.

Request: {"command": "print\_data", "message": null, "state": "completed"}

Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that it could not successfully download or begin processing the specified print file and settings.

Request: {"command": "print\_data", "message": "<error message>", "state": "failed"}

Response: None

#### Printer control

PUB /printers/ID/command Server  
The server commands the printer to perform a print related task. Examples include pausing or canceling a print or going through the calibration routine.

Example Payload: {"command": "pause"|"resume"|"cancel"|"reset", "command\_token": "<token>"}

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges receipt of the command.

Request: {"command": "pause"|"resume"|"cancel"|"reset", "message": null, "state": "received"}

Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that it has successfully sent the specified command to the printer control firmware.

Request: {"command": "pause"|"resume"|"cancel"|"reset", "message": null, "state": "completed"}

Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that it could not successfully send the specified command to the printer control firmware.

Example Request: {"command": "pause"|"resume"|"cancel"|"reset", "message": "<error message>", "state": "failed"}

Example Response: None

#### Firmware upgrade

PUB /printers/ID/command Server  
The server commands the printer to download the specified firmware package and install it.

Example Payload: {"command": "firmware\_upgrade", "package\_url": "http://s3.backed.cloudfront/link", "command\_token": "<token>"}

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges receipt of the command.

Example Request: {"command": "firmware\_upgrade", "message": null, "state": "received"}

Example Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that it successfully downloaded and installed the specified firmware package.

Example Request: {"command": "firmware\_upgrade", "message": null, "state": "completed"}

Example Response: None

POST /api/v1/printers/command/12345 Printer  
The printer acknowledges that it could not successfully download or install the specified firmware package.

Example Request: {"command": "firmware\_upgrade", "message": "<error message>", "state": "failed"}

Example Response: None

### Slicing

POST /api/v1/jobs/ID/slice Web Client  
Web Client asks the Server to slice the specified job

POST <http://slicer.autodesk.com/slicing_job> Server

The server makes a request to the slicer to slice the file, if use\_deprecated\_firmware is true the slicer will package up the sliced files and settings into a tar.gz and call the callback\_url, if use\_deprecated\_firmware is false the slicer will just slice the files and send them and the settings to the repackager to package them up into a zip file.

Example Request: {"file\_url": "http://s3.backed.cloudfront/1234", "callback\_url": "http://this.server.autodesk.com/callback", "setting": {...json...}, "token": "abc123", "job\_name": "job", "use\_deprecated\_firmware": false}

200 Example Response: None

4XX Example Resonse: {"errors": {"file\_url": "invalid file", "callback\_url": "invalid url"}}

5XX Example Response: {"errors": {"error\_1234": "the server is on fire"}}

| Parameter | Description |
| --- | --- |
| file\_url | link to the model file |
| callback\_url | url to call when slicing/repackaging are complete |
| setting | hash of the slicer settings |
| token | unique token used to authenticate and correlate callback requests with the job that was asked to be sliced |
| job\_name | name of the job being sliced, this is used to set content disposition on the sliced files |
| use\_deprecated\_firmware | indicates if the job should be sliced for a printer running pre or post 2.0 firmware |

POST <http://repackager.autodesk.com/repackage> Slicer

Takes a sliced archive and adds a settings file to it, this is used by the slicer for initial packaging of jobs that are not using deprecated\_firmware (pre 2.0)

Example Request: {"file\_url": "http://s3.backed.cloudfront/1234", "callback\_url": "http://this.server.autodesk.com/callback", "setting": {...json...}, "token": "abc123", "job\_name": "job", "use\_deprecated\_firmware": false}

Example Response: None

| Parameter | Description |
| --- | --- |
| file\_url | link to the model file |
| callback\_url | url to call when slicing/repackaging are complete |
| setting | hash of the slicer settings |
| token | unique token used to authenticate and correlate callback requests with the job that was asked to be repackaged |
| job\_name | name of the job being sliced, this is used to set content disposition on the sliced files |
| use\_deprecated\_firmware | indicates if the job should be sliced for a printer running pre or post 2.0 firmware |
| num\_images | number of images that the slicer created when slicing the model |
| images\_directory | the location of the sliced images |

POST <http://this.server.autodesk.com/callback> Slicer or Repackager  
The slicer or Repackager calls the callback URL once slicing and packaging is complete

Example Success Request: {"file\_url": "http://s3.backed.cloudfront/1234\_sliced", "numOfImages": 5, "images\_directory": "http://s3.backed.cloudfront/1234\_images"}

Example Failure Request: {"errors": {"error\_4321": "unslicable file for some reason"}}

| Parameter | Description |
| --- | --- |
| file\_url | link to the model file |
| numOfImages | number of images in the sliced file |
| images\_directory | location of the sliced file |
| error | error if any that occurred in a failed attempt to slice |

Response: None

### Repackaging

Along with its use in the normal packaging of post 2.0 firmware sliced jobs, the repackaging server also helps with reprinting jobs. Jobs that were sliced post 2.0 that get reprinted and do not having any slicer settings change can be repackaged instead of re-sliced. The repackager simply takes existing sliced and packaged zip file from the source job, and updates the zip with the new printer settings.

POST /jobs/ID/repackage Web Client

Asks the server to send a repackage request to the repackager for the specified job

POST <http://repackager.autodesk.com/repackage> Server

Repackages the sliced archive with new settings

Example Request: {"file\_url": "http://s3.backed.cloudfront/1234", "callback\_url": "http://this.server.autodesk.com/callback", "setting": {...json...}, "token": "abc123", "job\_name": "job", "use\_deprecated\_firmware": false}

Example Response: None

| Parameter | Description |
| --- | --- |
| file\_url | link to the model file |
| callback\_url | url to call when slicing/repackaging are complete |
| setting | hash of the slicer settings |
| token | unique token used to authenticate and correlate callback requests with the job that was asked to be repackaged |
| job\_name | name of the job being sliced, this is used to set content disposition on the sliced files |
| use\_deprecated\_firmware | indicates if the job should be sliced for a printer running pre or post 2.0 firmware |
| num\_images | number of images that the slicer created when slicing the model |
| images\_directory | the location of the sliced images |

POST <http://this.server.autodesk.com/callback> Slicer  
The repackager calls the callback URL once slicing and packaging is complete

Example Success Request: {"file\_url": "http://s3.backed.cloudfront/1234\_sliced", "numOfImages": 5, "images\_directory": "http://s3.backed.cloudfront/1234\_images"}

Example Failure Request: {"errors": {"error\_4321": "unslicable file for some reason"}}

| Parameter | Description |
| --- | --- |
| file\_url | link to the model file |
| numOfImages | number of images in the sliced file |
| images\_directory | location of the sliced file |
| error | error if any that occurred in a failed attempt to slice |