UNIVERSITY OF MICHIGAN

Flask Video Streaming API

Global Challenges

Version 1.0

UNIVERSITY OF MICHIGAN

Table of Contents

UNIVERSITY OF MICHIGAN

Revision History

Version	Changes
1.0	Initial Version

1. Introduction

This document goes into detail how the API for the Flask Video Streaming[1] server works. Anything or anyone that wishes to get information to and from the server will need to understand what the server is looking for. Also included are some guidelines when things need to be added to the server's API.

[1] More information about the Flask Video Streaming server's internals can be found in Flask Video Streaming Project Information

2. API Table

Route	Arguments	Description	Notes
/		Gives the homepage.	
/autofocus	status (true | false)	Turns autofocus on or off.	Only on supported cameras
/database		Gives the database editor page	
/database/add	filename (string)	Adds a file to the database.	
/database/new		Deletes old database and builds a new one	
/database/remove	filename (string)	Removes a file from the database and deletes it from the server.	
/get_database		Gives the database as a JSON object	
/grayscale	status (true | false)	Turns grayscale filter on or off.	
/light	status (true | false)	Turns light on or off.	Only if light is attached
/logs		Gives logs as JSON object	Should only be available for debugging/development
/media/<filename>		Sends the file requested if it is in the media folder	
/reboot	key (string)	Reboots the server if the key is valid	
/resolution	status (true | false)	Toggles between normal and scaled down image.	
/set_brightness	value (int)	Sets brightness of lights.	Only if light is attached
/set_color	value (string)	Sets color of lights.	Only if light is attached
/set_focus	value (float)	Sets focal depth of camera.	Only on supported cameras
/shutdown	key (string)	Shuts down the server if the key is valid	
/snapshot	filename (string) date	Saves an image from the stream.	

/step_focus	direction	Changes focal depth by 5.	Only on supported cameras
/video_capture	status (true | false) filename (string) date	Marks the start and stop of when to save video from the stream.	
/video_feed		Gives a constant video stream as M-JPEG	Does not work on Internet Explorer

3. API Breakdown

a. /

This gives the landing page for the server. It displays the video stream with various controls to perform actions on the server. Unless you are a web browser, this won't be very useful to any software accessing the server.

b. /autofocus

Using a GET on this with either true or false passed as status will toggle the autofocus on the camera as long as the camera supports that feature. The server will accept any of True, true, False or false, anything else will be ignored.

Passing false will turn autofocus on, and true will turn it off.

Returns 200 if action happened without errors (or is ignored), or 403 if the camera doesn't support changing autofocus.

c. /database

This gives a page with the listing of the database and controls to manipulate the server. Unless you are a web browser, this won't be very useful to any software accessing the server.

d. /database/add

Using a GET on this with a any string passed as filename will have the server search for that file in the media folder and add it to the database.

Returns 200 if the file is found and added to the database, otherwise returns 404 if it can't find the file requested.

e. /database/new

Using a GET on this will cause the current database to be dropped and a new one created. It doesn't change anything in the media folder.

Returns 200 always.

f. /database/remove

Using a GET on this with a any string passed as filename will have the server search for that file in the media folder and remove it from the database as well as delete the file on the server.

Returns 200 if the file is found, removed from the database and deleted, otherwise returns 404 if it can't find the file requested.

g. /get_database

Using a GET on this will give you a JSON object of the contents of the database. The JSON is formatted like so:

```
[{"date":"2018-1-31", "filename":"file1.jpg"},
 {"date":"2018-1-31", "filename":"file2.avi"},
 …]
```

Returns 200 and the JSON object

h. /grayscale

Using a GET on this with either true or false passed as status will toggle the grayscale filter on the video feed. The server will accept any of True, true, False or false, anything else will be ignored.

Passing false will turn the grayscale filter on, and true will turn it off.

Returns 200 if action happened without errors (or is ignored).

i. /light

Using a GET on this with either true or false passed as status will toggle the light(s) on the server assuming there is light on the server. The server will accept any of True, true, False or false, anything else will be ignored.

Passing false will turn the light(s) on, and true will turn it off.

Returns 200 if action happened without errors (or is ignored), or 403 if there are no lights attached to the server.

j. /logs

Using a GET on this will give you a JSON object of the contents of the server logs and boot logs. The JSON is formatted like so:

```
[{"System":"<log information>"},
 {"Server":"<log information>"}]
```

Returns 200 and the JSON object or 500 if there was a problem fetching the logs

k. /media/<filename>

Using a GET on this with a filename filled in will give that file from the server as long as that file exists.

Returns 200 and file requested.

l. /reboot

Using a GET on this with a string passed in as key will make the server reboot as long as the key is valid. Should the server find the key valid, the server will reboot one minute after this request is made.

Returns 200 if the server is going to reboot, 500 if there was a problem with trying to reboot, and 401 if the key is invalid.

m. /resolution

Using a GET on this with either true or false passed as status will toggle the resolution of the image sent from the server. It will either be the full resolution off the camera, or half as tall and wide. The server will accept any of True, true, False or false, anything else will be ignored.

Passing false will toggle to the half resolution image, true toggles to the full resolution image.

Returns 200 if action happened without errors (or is ignored).

n. /set_brightness

Using a GET on this with an integer value passed as value will set the brightness of the light(s) attached to the server, assuming the server has a light attached. Value should be from 0 to 255, all other values are ignored.

Returns 200 if action happened without errors (or is ignored), or 403 if there are no lights attached to the server.

o. /set_color

Using a GET on this with a string passed as value will change the color of the light(s) attached to the server, assuming the server has a light attached. The string should be a hexcode for the color desired in the form RRGGBB, anything too short will be ignored, and too long will have the first six characters evaluated. The color will be remembered so that when the light is toggled off and back on, it will be the color the lights come back as.

Returns 200 if action happened without errors (of is ignored), or 403 if there are no lights attached to the server.

p. /set_focus

Using a GET on this with an float passed as value will set the focal depth of the camera, if the camera supports changing focus. Value should be a number from 0.0 to 1.0, all values higher or lower will be assumed to be 1.0 or 0.0 as appropriate.

Returns 200 if action happened without errors, or 403 if the camera does not support changing focus.

q. /shutdown

Using a GET on this with a string passed in as key will make the server shut down as long as the key is valid. Should the server find the key valid, the server will shut down one minute after this request is made.

Returns 200 if the server is going to reboot, 500 if there was a problem with trying to shut down, and 401 if the key is invalid.

r. /snapshot

Using a GET on this with a string passed as filename and a date passed as date will have the server save a picture from the stream. The picture will be saved

onto the server with the filename that was given as filename, and the database will be updated with the filename and date the picture was taken. Date can be left blank, in which case the server will use its internal clock for noting when the picture was taken.

Returns 200 if the picture was saved, or 500 if there was a problem saving the picture.

s. /step_focus

Using a GET on this with a signed integer or float passed as direction will change the focal depth by a set amount if the camera supports changing focus. The server is only checking the sign of the number passed by direction, so the type doesn't matter.

A positive direction will increase the focal depth, a negative direction will decrease the focal depth.

Returns 200 if action happened without errors, or 403 if the camera does not support changing focus.

t. /video_capture

Using a GET on this with either true or false passed as status, a date as date, and a string passed as filename will mark the start and end points of what to save from the stream as a video. The video will be saved with the given filename, saved on the server and added to the database with the given date and filename.

Passing false will mark the start of the recording, true marks the end.
Date only needs to be supplied when status=true.

Returns 200 if the video markers have be set properly, 500 if there was an error saving the video.

u. /video_feed

Using a GET on this will give a M-JPEG stream from the server. This route will not be too helpful unless you can process M-JPEG streams.

This route will not do anything on Internet Explorer.

Returns 200 and M-JPEG stream.

4. API Guidelines

When it comes time to add to the server's API, the following can be useful to keep the additions consistent with what is currently in use. These are not absolutes, and as always do what is best for making the new features work.

- Naming a new endpoint should use underscores to split up words (No camelCase).
- Arguments should be descriptive, so that a user can understand what needs to be entered should they be doing things by hand.
- For anything that needs to receive a boolean value, False should enable the action and True should disable the action.
- Any returned JSON messages should be wrapped up in an array.

5. Appendix

More information about how the server works can be found in Flask Video Streaming Project Information.