

REST API Quick start guide

Currently the only thing you can do with the REST API is to add new markers and change angle. If you have any additional request, please let us know.

Multicam Logger uses port 8888 by default, but this can be configured in the preferences of Multicam Logger.

Which document will "reply"?

As Multicam Logger is a "document based" application, you can have multiple documents opened. You should note that it's the front most document that will answer to your requests.

General requests

Start and Stop logging

Once a document is created (you can not do that with the API), you can start and stop logging using the following requests.

GET /start

GET /stop

Retrieve the list of available inputs

GET /inputs

Returns a list of inputs that are available for the "front" document. So it represents that inputs that were available when the document was created and thus can be different to what you can see in the preferences.

Example response:

```
[
  "INPUT 1",
  "INPUT 2",
  "INPUT 3",
  "INPUT 4",
]
```

Retrieve the current state

GET /status

Returns a few information, such as which input is selected in program and which one is in preview (their zero based index). Or the current timecode, the front document name and if it is currently logging or not.

Example response:

```
{
  "preview" : "7",
  "document_name" : "Untitled",
  "logging_state" : false,
  "program" : "8",
  "timecode" : "14:25:33;01"
}
```

Working with Markers

Creating markers using "POST"

To add a marker, you can either use GET or POST.

When using the POST method, you should pass a json dictionary with "name" to set the title of the marker.

POST add_marker

```
{
  "name" : "This is my Marker",
}
```

You can also add an offset to your marker. Send a negative value (-HH:mm:ss:ff) if you want to add an offset in the past (for example 5 seconds before the current time). Make sure to format this properly.

The "reference": "live" key should be specified here.

POST add_marker

```
{
  "name" : "This is my Marker",
  "reference": "live",
  "offset": "00:00:05:00" }
```

You can also add a marker at a specific time.

The "reference": "live" key should be specified here, and you must indicate a time.

POST add_marker

```
{
  "name" : "This is my Marker",
  "reference" : "absolute",
  "time" : "00:00:05:00" }
```

Creating markers using "GET"

When using the GET method, you don't need to pass a JSON, everything can be done in one url, so it is pretty convenient.

GET /add_marker?name=This%20is%20my%20Marker

Retrieving markers

You can also retrieve the list of markers that were already created:

GET /markers

Returns a list of markers already created.

Example response:

```
[
  {
    "name" : "Marker 1",
    "time" : "01:00:05;00"
  },
  {
    "name" : "Marker 2",
    "time" : "01:00:15;00"
  },
  {
    "name" : "Marker 1",
    "time" : "01:00:24;29"
  }
]
```

```
}  
]
```

You can also retrieve just a specific marker, by indicating its index (again it's zero based)

GET /markers/1

Returns the specified marker information.

Example response:

```
{  
  "name" : "Marker 2",  
  "time" : "01:00:15;00"  
}
```

Working with Angles

Changing angles

You can also change the selected angle in program and preview.

IMPORTANT NOTE: If you are working with an ATEM or Tricaster, changing in the UI of Multicam Logger also changes in the production switcher. So it means you can control your production switcher from a simple web page.

You can either set which input is in program, or which input is in preview, or both. Or you can just task to do a "TAKE".

Note that the index is 0 based. This also can be done either with a POST or a GET.

Assigning Preview and Program

In this case, indicate which input should be used for program and which one for preview.

```
POST change_angle  
{  
  "program" : "0",  
  "preview" : "1",  
}
```

And here with a GET.

GET /change_angle?program=0&preview=1

Assigning just the Program, or the Preview

Here either indicate which input to use in program OR which one in preview (following example for program change only)

```
POST change_angle  
{  
  "program" : "0",  
}
```

And here with a GET.

GET /change_angle?program=0

Doing a "TAKE"

And finally if you just want to do a "TAKE" (swap program and preview), just do not specify any program or preview.

```
POST change_angle
```

And here with a GET.

GET /change_angle

Retrieving angles

You can also retrieve the list of angle changes that were done:

GET angle_changes

This will return a dictionary with the list of angle changes.

Example response:

```
[
  {
    {
      "out_point" : "01:00:11;00",
      "in_point" : "01:00:01;00",
      "program" : 0,
      "duration" : "00:00:10;00"
    },
    {
      "out_point" : "01:00:20;29",
      "in_point" : "01:00:11;00",
      "program" : 1,
      "duration" : "00:00:09;29"
    },
    {
      "out_point" : "01:00:30;29",
      "in_point" : "01:00:20;29",
      "program" : 0,
      "duration" : "00:00:10;00"
    },
    {
      "out_point" : "01:00:40;29",
      "in_point" : "01:00:30;29",
      "program" : 1,
      "duration" : "00:00:10;00"
    },
    {
      "out_point" : "01:00:50;29",
      "in_point" : "01:00:40;29",
      "program" : 0,
      "duration" : "00:00:10;00"
    }
  }
]
```

Note that the "program" value shows the index of the inputs that was used in program. You can use GET /inputs to retrieve the names of these.

For more information

If you need more info or support about Multicam Logger, you can find it on [our support desk](#). And if you need additional information on the API and/or the websockets, submit a ticket and we'll be happy to guide you.