caMicroscope API Guide

Contents of this Page

Naming

- case_id: Used to uniquely identify an image.
- execution_id: Used to uniquely identify an algorithm. A case_id can have multiple execution_ids associated with it.
- subject_id: Used to uniquely idenify a patient. A subject_id can have multiple case_ids associated with it.

/ImageLoader

The /ImageLoader resource is a Node application that extracts image metadata (such as width and height of the image) from the existing whole slide image and stores it in a Mongo database.

HTTP Verb	QueryEndpoint	Query Parameters(Input)	Output(Description)
POST	/submitImage	case_id, File_location, subject_id	{“status”: “Success”} on success

Example of /ImageLoader

```
curl -v -F case_id=TCGA-02-0001 -F file_location=@TCGA-02-0001-01Z-00-DX1.83fce43e-42ac-4dcd-b156-2908e75f2e47.svs http://localhost:32799/submitData
```

Optional `study_id` Return type: `json` On success returns: `{"status":"success"}`

case_id: The unique identifier for the image

file_location: Location of the image on file system

/uAIMDataLoader

Node application that converts Aperio and binary masks to GeoJSON objects and stores them in a Mongo database

HTTP Verb	QueryEndpoint	Query Parameters(Input)	Output(Description)
POST	/submitMaskOrder	Case_id, Execution_id, Height, Width, X, Y, Type(optional)	{“jobId”: 12}
GET	/job/:id	None	{“status”:”incomplete”}

Example of /uAIMDataLoader

Example cURL command (Aperio):

```
$ curl localhost:6000/submitMaskOrder -F mask=@1.png -F case_id=cbtc_test_11 -F execution_id=ganesh:test2 -F
width=1743 -F height=2017 -F x=0 -F y=0
```

Example cURL command (binary mask):

```
$ curl localhost:6000/submitMaskOrder -F mask=@1.xml -F case_id=cbtc_test_11 -F execution_id=ganesh:test2 -F
width=1743 -F height=2017 -F x=0 -F y=0 -F type= maskfile
```

Variable	Definition
case_id	The unique identifier for the image
execution_id	The name(label) of the algorithm/run
width	width of the image/roi
height	height of the image/roi
x	top left x coordinate of the image/roi (Default: 0)
y	top left y coordinate of the image/roi (Default: 0)

/ImageData

Bindaas APIs for handling Image Metadata. (Internal APIs, not meant to be exposed)

HTTP Verb	QueryEndpoint	Query Parameters (Input)	Output(Description)	Additional Description
GET	/fileLocation	case_id	Get file location of image	Used by the caMicroscope viewer to give the image location to its backend server.
	/mpp	case_id	Get mpp-x, mpp-y of the image	Used by the viewer to adjust the scales.
	/metadata	case_id	Get all metadata of the image(width, height, mpp etc.)	Used by dynamic services to get width, height, and file location of the image.
	/md5	case_id	Get MD5 of the image file	Used to check the MD5 to make sure that the image was loaded successfully.
POST	/json	metdata	metadata of the image in JSON format	Stored in Mongo(db: Camicroscope, collection: ImageData).

Example of /ImageData

POST

```
$ curl -H "Content-Type: application/json" -X POST -d '{"case_id": "TCGA-01-0001", "width": 30000, "height":
31000}' http://localhost:9099/Camicroscope/ImageData/json
```

GET

```
$ curl http://localhost:9099/Camicroscope/ImageData/fileLocation
```

/MarkupData

Bindaas

HTTP Verb	QueryEndpoint	Query Parameters (Input)	Output (Description)	Additional Information

GET	/multipleMarkups	case_id, x1,y1,x2,y2, footprint, algorithms	GeoJSON	x1, y1, x2, y2 are the coordinates of a rectangular region of interest. (x1,y1) is the top-left coordinate pair. (x2, y2) is the bottom-right coordinate pair.
N/A	/executionIds	case_id	List of execution IDs available for a case	The case_id uniquely identifies an image. The execution_ids identify the image segmentation algorithm that is run on images. The results of these images appear as green polygons in caMicroscope.

Examples of /MarkupData

/uAIMTemplates

Schema: https://github.com/joshfire/jsonform

HTTP Verb	QueryEndpoint	Query Parameters(Input)	Output(Description)
GET	/template	N/A	Jsonform template
POST	/json	Form in JSON format(see this)	200 OK

Examples of /uAIMTemplates

Examples of authoring a JSON form can be seen here.

Jobs

APIs for posting and retrieving tasks for caMicroscope. It uses Kue to maintain the tasks. It is well documented.

HTTP Verb	QueryEndpoint	Query Parameters(Input)	Output(Description)
POST	/job	Job data in JSON format	N/A
GET	/job	task_id	N/A

Examples of /Tasks

Example of POSTing a task

```
$ curl -H "Content-Type: application/json" -X POST -d \
    '{
       "type": "order",
       "data": {

       },
       "options" : {
         "attempts": 5,
         "priority": "high"
       }
     }' http://localhost:3000/job
```

will output:

```
{"message": "job created", "id": 3}
```

This ID can be used to query the status of the job.

GET /job/:id

```
curl http://localhost/job/3
```

Example of GETting status of a task

```
{"id":"3","type":"queue","data":{},"priority":-10,"progress":"100","state":"complete","attempts":null,"
created_at":"1309973155248","updated_at":"1309973155248","duration":"15002"}
```