
FLINT.Cloud Technical Guide

Sneha Mishra, Arnav Tiwari & contributors

Apr 14, 2022

CONTENTS

| | | |
|----------|--|-----------|
| 1 | FLINT.Cloud API Docs | 3 |
| 1.1 | GCBM: New simulation | 3 |
| 1.2 | GCBM: Upload files to simulation | 3 |
| 1.3 | GCBM: Run dynamic simulation | 7 |
| 1.4 | GCBM: Download simulation | 8 |
| 1.5 | GCBM: List simulations | 8 |
| 2 | FLINT.Cloud Development Setup | 11 |
| 2.1 | Git and Github guide | 11 |

The project aims to build a continuous deployment pipeline to offer FLINT as a SaaS over cloud. The FLINT-GCBM scripts are converted to a REST Api service built using Flask. This API Service is designed to be consumed by Gcloud.

This Documentation is meant for Developers wishing to contribute to FLINT.Cloud repository. If you would like to get in touch with the maintainers for other reasons, please drop a mail at info@moja-global.com.

FLINT.CLOUD API DOCS

This section guides first-time contributors through the v1 API consumed by FLINT.Cloud. Data is returned in JSON Format and HTTP response status codes have been designated to declare Success or Failure. The API is created using Flask and flask-restful.

1.1 GCBM: New simulation

Creates a new GCBM simulation by passing a title for the simulation in the request form-data. If the simulation is already existing, an error message is returned with 400 HTTP Status Code.

Example request:

Bash

```
$ curl --location --request POST 'http://0.0.0.0:8080/gcbm/new' \  
--header 'Content-Type: application/json' \  
--header 'accept: application/json' \  
--form 'title="example_new_1"'
```

Example response:

```
{  
  "data": "New simulation started. Please move on to the next stage for   
↪uploading files at /gcbm/upload."  
}
```

Request FormData Object

- **title** (*string*) – required title of simulation.

1.2 GCBM: Upload files to simulation

Upload input files for the GCBM simulation by passing the files as request FormData. If the upload has any issue, an error message is returned with 400 HTTP Status Code.

Example request:

Bash

```
$ curl --location --request POST 'http://0.0.0.0:8080/gcbm/upload' \  
--header 'Content-Type: multipart/form-data; boundary=---  
↪011000010111000001101001' \  
--header 'accept: application/json' \  
--form 'db=@"/home/kalilinux/Documents/Medium_Local_Run/input_database/gcbm_  
↪input.db"' \  
--form 'title="example"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/  
↪provider_config.json"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/  
↪localdomain.json"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/  
↪logging.conf"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/  
↪modules_output.json"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/  
↪variables.json"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/  
↪spinup.json"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/pools_  
↪cbm.json"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/  
↪modules_cbm.json"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/  
↪internal_variables.json"' \  
--form 'config_files=@"/home/kalilinux/Documents/Medium_Local_Run/config/gcbm_  
↪config.cfg"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/study_  
↪area.json"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/LdSpp_  
↪moja.tiff.aux.xml"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/LdSpp_  
↪moja.tiff"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/LdSpp_  
↪moja.json"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/  
↪initial_age_moja.tiff.aux.xml"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/  
↪initial_age_moja.tiff"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/  
↪initial_age_moja.json"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/  
↪mpb2000_moja.tiff.aux.xml"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/  
↪mpb2000_moja.tiff"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/  
↪mpb2000_moja.json"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/  
↪mpb1999_moja.tiff.aux.xml"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/  
↪mpb1999_moja.json"' \  
--form 'input=@"/home/kalilinux/Documents/Medium_Local_Run/layers/tiled/  
↪mpb1992_moja.tiff.aux.xml"' \  
↪
```

(continues on next page)

(continued from previous page)

```

--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1992_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1992_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1990_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1990_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1990_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1999_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1996_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1996_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1996_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1994_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1994_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1994_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1993_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1993_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1993_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1995_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1995_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1995_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1991_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1991_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪mpb1991_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/mean_
↪annual_temperature_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/mean_
↪annual_temperature_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/mean_
↪annual_temperature_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪2000_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪2000_moja.tiff"' \

```

(continues on next page)

(continued from previous page)

```

--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪2000_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1995_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1995_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1995_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1994_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1994_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1994_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1993_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1993_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1993_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1992_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1992_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1992_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1991_moja.tiff.aux.xml"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1991_moja.tiff"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/fire_
↪1991_moja.json"' \
--form 'input=@" /home/kalilinux/Documents/Medium_Local_Run/layers/tiled/
↪bounding_box.tiff"'

```

Example response:

```

{
  "data": "All files uploaded sucessfully. Proceed to the next step of the
↪API at gcbm/dynamic."
}

```

Request FormData Object

- **db** (*file*) – required input db for gcbm (gcbm_input.db).
- **title** (*string*) – required title of simulation (this was set in /new).
- **config_files** (*files*) – required config_files (cgf and json files in config folder).
- **input** (*string*) – required input files (tiff and json files in layers/tiled folder).

1.3 GCBM: Run dynamic simulation

Run the GCBM simulation by passing the files as request FormData.

Example request:

Bash

```
$ curl --location --request POST 'http://0.0.0.0:8080/gcbm/dynamic' \
--header 'Content-Type: application/json' \
--header 'accept: application/json' \
--form 'title="example_new_1"'
```

Example response:

```
{
  "data": {
    "execTime": 187.98220562934875,
    "exitCode": 0,
    "response": "Operation executed successfully. Downloadable links for
↳ input and output are attached in the response. Alternatively, you may also
↳ download this simulation input and output results by making a request at
↳ gcbm/download with the title in the body.",
    "urls": {
      "input": "https://storage.googleapis.com/simulation_data_flint-
↳ cloud/simulations/simulation-loop/input.zip?X-Goog-Algorithm=GOOG4-RSA-
↳ SHA256&X-Goog-Credential=authentication-bucket%40flint-cloud.iam.
↳ gserviceaccount.com%2F20210506%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-
↳ Date=20210506T201854Z&X-Goog-Expires=1800&X-Goog-SignedHeaders=host&X-Goog-
↳ Signature=9c8b3ff058646f1d5b6601cb9b68421e8ce11e9e71ced83fbc9948ba60de14620b8dd53a15e62e63915302
↳ ",
      "message": "These links are valid only upto 15 mins. Incase the
↳ links expire, you may create a new request.",
      "output": "https://storage.googleapis.com/simulation_data_flint-
↳ cloud/simulations/simulation-loop/output.zip?X-Goog-Algorithm=GOOG4-RSA-
↳ SHA256&X-Goog-Credential=authentication-bucket%40flint-cloud.iam.
↳ gserviceaccount.com%2F20210506%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-
↳ Date=20210506T201854Z&X-Goog-Expires=1800&X-Goog-SignedHeaders=host&X-Goog-
↳ Signature=60b0679176a15ac87a1aef22c90d0082330122c0e06cd7289e65266839f76d1c2113d5c98a36f634c88256
↳ "
    }
  }
}
```

Request FormData Object

- **title** (*string*) – required title of simulation (this was set in /new).

1.4 GCBM: Download simulation

Download the GCBM simulation input and output zipped folders by passing the files as request FormData. The downloadable links returned are only valid for 30 mins.

Example request:

Bash

```
$ curl --location --request POST 'http://0.0.0.0:8080/gcbm/download' \
--header 'Content-Type: multipart/form-data; boundary=---
↪011000010111000001101001' \
--header 'accept: application/json' \
--form 'title="example_new_1"'
```

Example response:

```
{
  "input": "https://storage.googleapis.com/simulation_data_flint-cloud/
↪simulations/simulation-simulations/examplenew1/input.zip?X-Goog-
↪Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=authentication-bucket%40flint-
↪cloud.iam.gserviceaccount.com%2F20210506%2Fauto%2Fstorage%2Fgoog4_request&X-
↪Goog-Date=20210506T200813Z&X-Goog-Expires=1800&X-Goog-SignedHeaders=host&X-
↪Goog-
↪Signature=6bf4f1d50e2fd9b2b6453d245db5ed14e23074027f3d029f6a2873e42b555b15ee31b4cd7ca561910e8820
↪",
  "message": "These links are valid only upto 15 mins. Incase the links
↪expire, you may create a new request.",
  "output": "https://storage.googleapis.com/simulation_data_flint-cloud/
↪simulations/simulation-simulations/examplenew1/output.zip?X-Goog-
↪Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=authentication-bucket%40flint-
↪cloud.iam.gserviceaccount.com%2F20210506%2Fauto%2Fstorage%2Fgoog4_request&X-
↪Goog-Date=20210506T200813Z&X-Goog-Expires=1800&X-Goog-SignedHeaders=host&X-
↪Goog-
↪Signature=233e90633d613ca6bd830682ec5014e9b89e9e28b1995cb04249e158ff0677cec730f342b17dfe1e99f571
↪"
}
```

Request FormData Object

- **title** (*string*) – required title of simulation (this was set in /new).

1.5 GCBM: List simulations

List all the GCBM simulations. Before creating a new simulation, it might be a good idea to check whether the simulation already exists or not using this endpoint. You may also utilize the title of the simulation available here to download results for that specific simulation.

Example request:

Bash

```
$ curl --location --request POST 'http://0.0.0.0:8080/gcbm/download' \  
--header 'Content-Type: application/json' \  
--header 'accept: application/json' \  

```

Example response:

```
{  
  "data": [  
    "simulation-example",  
    "simulation-exampleneu",  
    "simulation-exampleneu1",  
    "simulation-localrun"  
  ],  
  "message": "To create a new simulation, create a request at gcbm/new. To  
→access the results of the existing simulations, create a request at gcbm/  
→download."  
}
```


FLINT.CLOUD DEVELOPMENT SETUP

This section guides first-time contributors through installing FLINT.Cloud development environment.

The recommended method for installing the FLINT.Cloud development environment is using App Engine on Gcloud.

Before Setting Up FLINT.Cloud

- Please make sure that all the prerequisites required have been installed and configured correctly.
- The repository has been forked and cloned following the Git and Github guide [here](#) .
- It is highly recommended to first setup the FLINT.Cloud API locally to understand the endpoints.

Contents:

2.1 Git and Github guide

This guide is to help new contributors setup git, github and navigate their way through making contributions to moja global repositories. It covers the entire process of contributing right from installing git to opening pull requests.

2.1.1 Setup this project using Git

Before setting up this project using Git make sure you have installed and configured git by following the instructions [here](#).

2.1.2 Fork and Clone this project

- In your browser, visit <https://github.com/moja-global/FLINT.Cloud>. In the upper left corner, there is a **Fork** button. Please click on it to create a fork/copy of the repository on your profile.
- In the terminal screen, clone this repo by running the command where `your-username` represents your Github username.

```
git clone https://github.com/<your-username>/FLINT.Cloud/
```

- Enter into the newly created project folder by running the command

```
cd FLINT.Cloud
```

- Configure upstream for the fork so that git can sync work from the upstream if it is updated by running the command

```
git remote add upstream https://github.com/moja-global/FLINT.Cloud/
```

- Check if upstream is configured by running the command and check if upstream is shown or not.

```
git remote -v
```

- Now, the project is setup using Git. Please carry on with instructions on how to set this up on [Windows](#) or [Linux](#) here. You can revisit this section when you are ready to make a contribution.

2.1.3 Claim an issue

This section will demonstrate how to claim an issue to work on using botmojaglobal.

To work on an issue, claim it by adding a comment with `@botmojaglobal claim` to the issue thread. botmojaglobal is a GitHub workflow bot forked from the [zulipbot](#); it will assign you to the issue and label the issue as `in progress`. Some additional notes:

- You can only claim issues with the `Good for newcomers` or `Help Wanted` labels. botmojaglobal will give you an error if you try to claim an issue without one of those labels.
- Please feel free to ask questions on how to approach the issue or if the tests are failing. The maintainers/reviewers will try to get back to you as soon as possible. You can reach us on the [moja-global slack](#), or through Github.
- If your pull request has some requested changes, after working on it don't forget to leave a comment asking for a review since the reviewers aren't notified when a pull request is updated.

2.1.4 Make a contribution

This section will show you step-by-step how to make a contribution to FLINT.Cloud using git.

- FLINT.Cloud stable branch is **master**. All pull requests should be against **master** branch only. Make sure you are in the project directory and checkout to master branch with this command.

```
git checkout master
```

- Choose an issue to work on. We have issues specifically labelled `Good for newcomers` and `Help Wanted` for new contributors to claim. Before starting to work on any issue, make sure you have claimed it.
- Create a new feature branch from master branch to work on. The feature branch should have a short name that is relevant to the issue that you will be working on. For example, if you are working on improving documentation in the readme for adding a badge, the branch can be name `add_badge_readme`.

```
git checkout -b <feature-branch-name>
```

- Work on the task. Add tests and documentation for your changes if required. When you are done with your changes, you can check all the files changes using the following command.

```
git status
```

- Add the relevant files and commit the changes. Please make sure that only those files required for this contribution are added. You can later modify your pull request to add other files as per your requirement.

```
git add <file> <file> ...
```

- While committing the changes, make sure your commit message follows our commit-message guidelines mentioned here.


```
git commit -m "relevant commit message"
```

- Make sure your fork is in sync with the latest changes of master. For this rebase your branch against the latest master by following the commands below.

```
git checkout master
git pull origin master
git checkout <your-branch-name>
git rebase master
```

- In case there are any merge conflicts on running the rebase command, follow this guide to resolve them.
- You can now push your changes onto your feature branch using the command below.

```
git push origin <your-branch-name>
```

2.1.5 Create a pull request for your contribution

You can now create a pull request to get your changes merged into the upstream master branch. Follow this step-by-step guide to create a pull request on Github.

- Navigate to the pull requests tab under FLINT.Cloud. Click on the **New pull request** button. Compare your feature branch against the **master** branch to create the pull request. Fill the pull request template by linking the issue number solved.
- In case your pull request is a work in progress, don't forget to add "WIP" in the title of your pull request to let the maintainers know that the pull request is not ready for review yet.
- Please be patient, someone from our team will review your pull request shortly and provide feedback. In case there are changes requested, you can follow the section below on how to update/modify your pull request.
- Also make sure that your pull request is in sync with the latest master at all times.

NOTE: Don't forget to get credits for your contributions once it gets merged by following this guide [here](#).

2.1.6 Modify your pull request

In case your pull request needs further changes, you can update your pull request by following the steps below.

- Checkout on your feature branch of the pull request.
- Add the changes as required and commit using the amend flag. This will update the last commit thus keeping the commit history clean and within a single commit.

```
git add <file1> <file2>
git commit -amend
```

- Push this onto your feature branch but this time with force flag. This will update the pull request automatically. The reviewer won't be notified about this updation, so leave a comment in your pull request if you want a review.

```
git push origin <your-branch-name> --force
```

2.1.7 About Hadolint

Hadolint is a Dockerfile linter that helps you build best practice Docker images.