

---

# **FLINT-UI**

***Release 0.1.0***

**moja global docs team**

**Mar 25, 2023**



# CONTENTS

|          |                          |          |
|----------|--------------------------|----------|
| <b>1</b> | <b>Installation</b>      | <b>3</b> |
| <b>2</b> | <b>Development Guide</b> | <b>7</b> |
| 2.1      | Prerequisites . . . . .  | 7        |
| 2.2      | Outline . . . . .        | 7        |



FLINT-UI is a web application which makes it easier for users to run FLINT Docker and modules easily on their local device or through the [FLINT Cloud API](#).

The main users of the FLINT are scientists, policymakers, forest officers, GIS specialists, developers to name a few. This project provides an easy to use web interface for scientists, policymakers, and forest officers to solve issues around the technical setup and usage of standard FLINT. This project is compatible with the FLINT Cloud API as well, so that the user can manage their FLINT Cloud deployments.

This project also provides a base for governments to build their own application on top of the FLINT. Every government possess a different FLINT implementation, exclusive and custom modules and configuration, and hence the tool can be customised for every module.



## INSTALLATION

To set up this project locally, follow the procedure below:

1. Pull the docker image below to run `flint.example`:

```
docker pull shubhamkarande13/flint.example:bionic
```

2. Clone the `FLINT.UI` repository:

When using submodules, the installation code needs to be:

```
git clone --recursive https://github.com/moja-global/flint-ui
```

Or, if you've already initialized the repository without the submodule:

```
git submodule update --init --recursive
```

3. Now get inside the repository directory and build the `docker-compose.yml` file by following the below command:

```
docker-compose up
```

This will build all the docker images inside the `docker-compose.yml` file. It will take some time and after that, you can see all images listed in the Docker app.

`docker-compose.yml` file is composed of three services: `flint.example.api`, `flint.gcbm.api`, and `flint.ui`.

Now you can start all the containers by clicking on the `START` button or using the command `docker-compose up`. You can also use the command `docker start {name of the container}` to start one specific container.

The list of all containers will look like this in the Docker app:

If you want to shut down a specific container, click on the `STOP` button of that specific container in the Docker app. You can also use the command line to stop a specific container by using the command `docker stop {name of the container you want to stop}`.

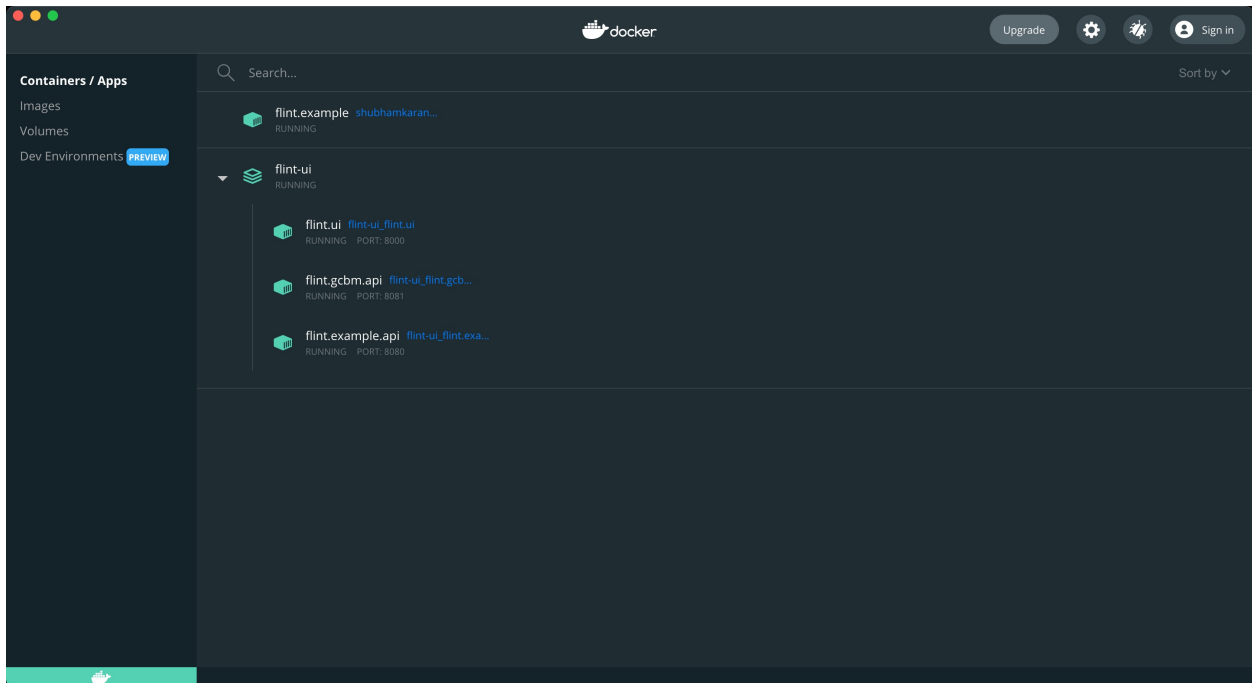
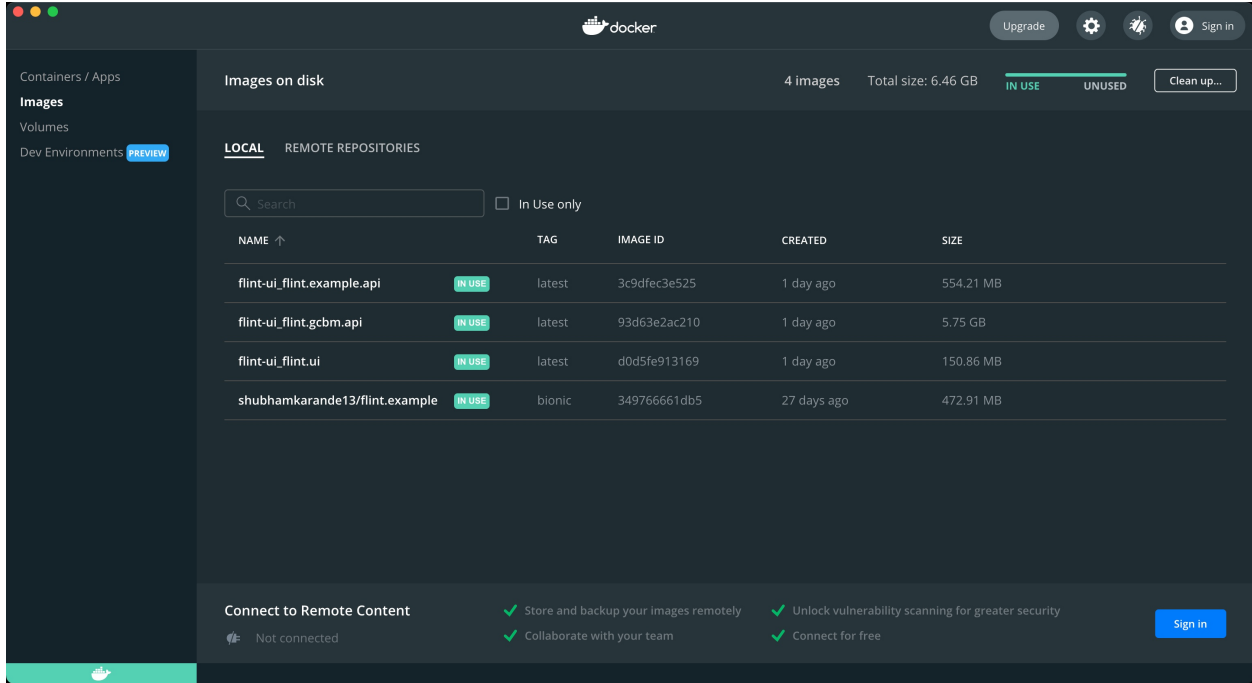
To shut down the entire `docker-compose.yml` file and all the services inside it, use the command `docker-compose down`.

---

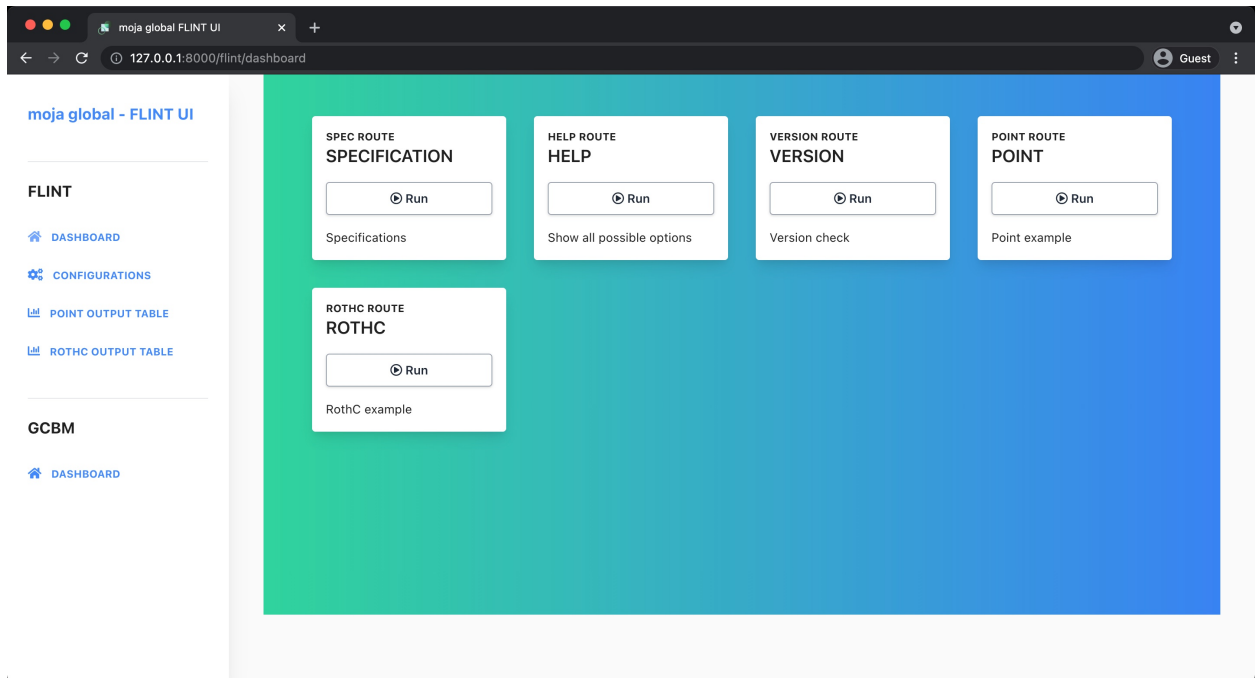
**Note:** You can also run `docker-compose.yml` in Detached mode. Detached mode, shown by the option `--detach` or `-d`, means that a Docker container runs in the background of your terminal. It does not receive input or display output. The command for running `docker-compose up` in Detached mode is `docker-compose up -d`.

---

4. To view the UI, please navigate to <http://127.0.0.1:8000>. It will look something like this:









## DEVELOPMENT GUIDE

This documentation is an entry level guide into the architectural disposition of the FLINT UI. Its role is to show the building blocks of the FLINT UI and the rationale behind it. More technical explanations of the source code should be found within the source directory of FLINT UI.

This documentation is strongly related to a series of guides detailing how to *set up* and operate the FLINT UI and develop further on the same. We recommend that you have a look at them as well for a more fuller understanding of the domain.

### 2.1 Prerequisites

|   | Prerequisite  |
|---|---|
| 1 | We need to have sufficient VueJS development skills |
| 2 | We need to have sufficient Docker skills            |

### 2.2 Outline

|   | Prerequisite   |
|---|--|
| 1 | The major design elements that make up the FLINT-UI            |
| 2 | The functional decomposition of the above elements             |
| 3 | The execution elements of the functionally decomposed elements |
| 4 | The projects's source code organization                        |
| 5 | The long term vision and direction for continuous improvement  |
| 6 | The logical next areas for improvement                         |

We will explore each of the above concepts in detail in the specific parts of the documentation.

## 2.2.1 Development Notes

### Docker

Run commands from FLINT UI repo folder:

FLINT Example

1. To build image

```
docker build -t flint-api ./flint.cloud/local/rest_api_flint.example
```

2. To run image

```
docker run --rm -d -p 8080:8080 flint-api
```

GCBM

1. To build image

```
docker build -t gcbm-api ./flint.cloud/local/rest_api_gcbm
```

2. To run image

```
docker run --rm -d -p 8081:8080 gcbm-api
```

### Node

This project requires Node version  $\geq 14$

1. To check the version of installed Node, run the following command:

```
node -v
```

2. You can also replace `-v` with `--version` and it should work the same.

3. To update Node to the latest version, ensure `nvm` is installed (see [this](#)) and run the following command:

```
nvm install <version>
```

(replacing `<version>` with version of Node you want to install.)

### Yarn

It is recommended to install Yarn through the `npm`.

1. Once you have `npm` installed you can run the following both to install and upgrade Yarn:

```
npm install --global yarn
```

2. To check the version of Yarn installed by running the following command:

```
yarn --version
```

## Vue app

To run the Vue app locally, you can follow the below procedure:

1. Go to the `flint.ui` folder and run `yarn install`.
2. Run `yarn serve` to start the Vue app server.
3. Go to `127.0.0.1:8000` to view the Vue app.

## E2E Testing

To test the Vue app locally, you can follow the below procedure:

1. Go to the `flint.ui` folder
2. Run `yarn test:e2e`
3. This will run the app in production mode at `127.0.0.1:8000` for E2E testing.
4. The Cypress interactive GUI will be opened where E2E testing can be run

## Storybook in local development

Storybook dependencies and configuration are located under the `flint.ui/.storybook` directory.

To build and launch Storybook locally, we have to follow the below procedure:

Go to the `flint.ui` directory.

1. Install Storybook dependencies:

```
yarn install
```

2. Start the Storybook in development mode:

```
yarn storybook
```

Go to `localhost:6006` to view the Storybook.

## Storybook in production

We have deployed Storybook on [Chromatic](#). It is made by Storybook maintainers. Chromatic automates gathering UI feedback, visual testing, and documentation, so developers can iterate faster with less manual work.

- Get a permalink to FLINT-UI's latest published Storybook on a given branch. Replace `<branch>` with the designated branch.

```
https://<branch>--612f3734da98c9003a398626.chromatic.com
```

- Get a permalink to the latest published component or library on a given branch. Add the `&branch=<branch>` query parameter to the url.

```
https://chromatic.com/library?appId=612f3734da98c9003a398626&branch=<branch>
```

Storybook live (master branch): <https://master-612f3734da98c9003a398626.chromatic.com>

Storybook documentation can be found on the index page of <https://master-612f3734da98c9003a398626.chromatic.com/>.

### Dependency management

We are following [Yarn](#) package manager.

Our `package.json` file is gitignored to enforce strict management of our client dependencies.

In order to add a new dependency, please:

1. use `yarn add <your-package>`
2. `git checkout -b dependency/<your-package>`
3. `git add --force flint.ui/package.json`
4. `git push` and submit a PR with only the `package.json` changes

Our CI build will test for conflicts and your new dependency will be added after your PR has been reviewed.

### Code Style

The Vue.js style guide has been used to maintain a consistent style throughout the project. Click [here](#) to learn more about the style guide.

One exception is the case used to describe the ecological models exposed to our FLINT client. Because model names, and sometimes model variables, often use acronyms these remain capitalized.

Obvious examples are:

- GCBM: Generic Carbon Budget model
- FLINT: Full Lands Integration Tool
- RothC: Rothamstead Carbon model

We recommend using ESLint and the Prettier plugins to make conforming to the Vue style guide easy. There is [official integration](#) for VSCode and other editors.

If you have ESLint installed globally, you can run it from the `flint.ui` directory. For example, this will detect any errors in the `src` folder:

```
eslint --ext .js,.vue src
```

Minor errors can be fixed automatically:

```
eslint --fix --ext .js,.vue src
```