
culturemeshFFB Documentation

Release beta

Stanford Code the Change

Dec 18, 2018

Table of Contents

1	Developer Quick Start	3
1.1	Getting Started	3
1.2	Environment Variables	6
2	API Client	7
2.1	API Spec	7
2.2	Acknowledgements	7
3	Deploying CultureMesh FFB	9
3.1	How it works now	9
3.2	How it should probably work in the future	9
4	Identity	11
4.1	Colors	11
5	Testing	13
5.1	Unit Tests	13
5.2	Integration Tests	13
6	Developer Notes	15
7	Contact	17

CultureMesh FFB is mobile web site for CultureMesh designed for the [Facebook Free Basics](#) platform – you can also think of it as CultureMesh ‘lite’.

Because Facebook Free Basics strives to provide services to a very wide range of phones and data plans, the mobile website needs to function properly without:

1. **JavaScript**
2. `iframes`
3. Video and large images
4. Flash and Java applets

You can find more technical details about these requirements [here](#).

CultureMesh FFB started as a [Stanford Code the Change](#) project.

This section will get you up and running with CultureMesh FFB.

1.1 Getting Started

CultureMesh FFB is a project to bring CultureMesh to the Facebook Free Basics platform. Facebook Free Basics (FFB) is an initiative at Facebook that provides free light weight (no JavaScript, no large images, etc.) web services to as many cellphones around the world as possible. CultureMesh FFB is a Python-Flask web app.

Because the intended audience for this service may not be very technically inclined, it is very important to design the site and it's functions to be as straightforward as possible.

It is equally important to keep in mind that the site is to use **NO JAVASCRIPT**, and that it will benefit from being as light (in memory and computation) as possible. Please keep this in mind as you contribute to the project.

1.1.1 Running Locally

All our code (save secrets) lives on [GitHub](#).

Follow these steps to run the website locally.

1. Get the code – choose a directory and run

```
$ git clone https://github.com/Code-The-Change/culturemeshFFB
```

2. Install python from <https://python.org> or via your favorite package manager
3. Install virtualenv

```
$ pip3 install virtualenv
```

4. If you get a note from `pip` about `virtualenv` not being in your `PATH`, you need to perform this step. `PATH` is a variable accessible from any bash terminal you run, and it tells bash where to look for the commands you enter. It is a list of directories separated by `:`. You can see yours by running `echo $PATH`. To run

virtualenv commands, you need to add python's packages to your PATH by editing or creating the file `~/.bash_profile` on MacOS. To that file add the following lines:

```
PATH="<Path from pip message>:$PATH"
export PATH
```

5. Then you can install dependencies into a virtual environment

```
$ cd culturemeshFFB
$ virtualenv .env
$ source .env/bin/activate
$ pip install -r requirements.txt
```

6. You also need to set some required environment variables (see [Environment Variables](#))

```
$ export WTF_CSRF_SECRET_KEY=...
$ export CULTUREMESH_API_KEY=...
```

7. Start the Flask App.

```
$ python -u run.py
```

As an alternative to combine the above two steps, you can create a simple bash script to set the environment variables and start up the app for you. To do so, create a file `run.sh` with the following contents, filling in the missing information:

```
#!/usr/bin/env bash

export CULTUREMESH_API_KEY=<API Key>
export WTF_CSRF_SECRET_KEY=<CSRF Secret>
export CULTUREMESH_API_BASE_ENDPOINT=<API Base>

python -u run.py
```

Then make the app executable:

```
chmod 700 run.sh
```

Whenever you want to start the app, just execute the script:

```
./run.sh
```

You'll see something like this on the terminal:

```
$ python run.py
* Restarting with stat
* Debugger is active!
* Debugger PIN: 202-914-549
* Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
```

You can then head over to your browser and type in `http://127.0.0.1:8080/` on the address bar.

Note: By default, the website (even if running locally) really communicates with the live CultureMesh API. However, the CultureMesh API itself currently reads and writes from a staging/dev database. “Flipping the switch” and enabling CultureMesh FFB in production would mean making the CultureMesh API point to the production database. No changes in FFB are required.

1.1.2 Contributing

Note: Before contributing or writing code, be sure to scan the codebase first. There are certain recurring paradigms (e.g. blueprint-specific util and config files) that you should follow.

All changes you make to the directory should go into a separate branch which you push and submit a pull request for:

1. Install dependencies

```
$ cd culturemeshFFB
$ virtualenv .env
$ source .env/bin/activate
$ pip install -r requirements.txt
```

2. Create a new branch

```
$ git checkout -b my-new-branch
```

3. Set environment variables (see *Environment Variables*)

```
$ export WTF_CSRF_SECRET_KEY=...
$ export CULTUREMESH_API_KEY=...
```

4. Make some awesome commits

5. Push the branch:

```
$ git push -u origin my-new-branch
```

6. Make sure there are no merge conflicts with master

7. Submit a pull request.

Warning: When opening the Pull Request choose the Code-The-Change base fork, not ericshong's

8. Select your reviewers

9. Wait until at least one other person submits a positive review (or make the requested changes). Once a positive review is submitted, you can merge the branch yourself from the GitHub website if your reviewer has not already done so. You should also make sure that your Travis CI build is green.

10. Update your local master branch and delete the old one

```
$ git checkout master && git pull
$ git branch -d my-new-branch
```

CultureMesh FFB is a Python-Flask webapp. I will not go into the details of the Flask microframework (blueprints, templates, routes, etc.) – there is already plenty of documentation for all of this online.

1.1.3 Simulating Mobile Web

CultureMesh FFB is meant to be accessed from low-end mobile devices and it runs without JavaScript.

You can simulate this type of environment from Chrome.

1. Run the webapp locally
2. Open the developer tools on chrome
3. Select the mobile view option (top left on the developer tools pane)
4. On the 'Network' tab, switch from 'Online' to 'Slow 3G' on the drop-down menu
5. Click on the three vertical dots on the top right of the developer tools pane
6. Go to 'settings' and select 'Disable JavaScript' under the 'Debugger' section

1.2 Environment Variables

You need to define two environment variables before you can start the application.

Variable	Purpose
WTF_CSRF_SECRET_KEY	A secret of your choosing for generating and validating CSRF tokens
CULTUREMESH_API_KEY	The key to access the CultureMesh API (contact us for the key)

CHAPTER 2

API Client

CultureMesh FFB communicates with CultureMesh via the CultureMesh API. CultureMesh FFB contains a `client` class that gives easy programmatic access to the resources available via the CultureMesh API – these are things like users, posts, networks, etc.

A typical usage of the `client` looks like this:

```
c = Client(mock=False)
post = c.get_post(current_post_id)
...
replies = c.get_post_replies(post["id"], NUM_REPLIES_TO_SHOW)

# Both 'post' and 'replies' contain real data from the CultureMesh API.
```

Dig into `culturemesh/client` to get a better sense of how the client is structured and to see all of the available methods.

2.1 API Spec

We use a Swagger spec to define the CultureMesh API. The file is publicly available in the CultureMesh API [repo](#). Link to file [here](#).

Note: To view the the methods, data, and objects in a nice format, copy and paste the contents of the spec into the editor at <https://editor.swagger.io/>.

2.2 Acknowledgements

The CultureMesh Client class was modeled heavily after the client in [this](#) project.

Deploying CultureMesh FFB

3.1 How it works now

We have been using Heroku to deploy CultureMesh FFB for developing and testing – the site can be accessed at <https://culturemesh-ffb.herokuapp.com/>. If you'd like to deploy the application as well, you can just make your own Heroku account, download the CLI tools, and follow one of the myriad tutorials online. Don't forget to set the required *Environment Variables* on Heroku as well, or your deploy won't work.

Deploying to FFB is as easy as submitting the URL of CultureMesh FFB via the Facebook Free Basics website and making sure that you meet the requirements of the platform.

3.2 How it should probably work in the future

In the near future, CultureMesh FFB should be available at <https://lite.culturemesh.com> and run on the same serving infrastructure as the CultureMesh API and CultureMesh for web. Currently, this is Bluehost. Once this is the case, the `client` in CultureMesh FFB class could point to the localhost even directly speak to the database – this will greatly reduce latency in API calls.

It would also be a good idea to look into AWS hosting in the future. We also suggest Dockerizing the website, the FFB site, and the API.

CHAPTER 4

Identity

4.1 Colors

Color	Code
CultureMesh Orange	#EF5330
CultureMesh Green	#97CB38
CultureMesh Grey	#3C3C3C

5.1 Unit Tests

To run all unit tests:

```
$ nosetests --verbosity=2 test/unit/*
```

To run all unit tests in a single file:

```
$ nosetests --verbosity=2 test/unit/path/to/test_file.py
```

To run a single unit test:

```
$ nosetests --verbosity=2 test/unit/path/to/test_file.py:test_x_y_and_z.py
```

Add unit tests for `culturemesh/path/to/file.py` at `test/unit/path/to/file/test_file.py`.

Note: At the moment we unit test only the `client` class. To ease development, we initially created a ‘mock’ client that returns fake API data from `data/`, and we use this for unit testing. In the future, these tests should be updated to use the `mock` library and the ‘mock’ client should be deprecated.

5.2 Integration Tests

There are currently no integration tests for CultureMesh FFB. There are, however, tools to make live debugging easier. For example, if you want to check that the server has registered code changes and is serving the latest your changes, you can use the `/dev/note` endpoint. This endpoint serves the contents of the template `note.html`, which you can change to display custom text. If that new text is displayed at that endpoint, you know the server has updated to reflect your changes.

CHAPTER 6

Developer Notes

- We are currently using Python 3.6.1 to run the Flask app.
- This project along with the CultureMesh API, CultureMesh Android, and the original CultureMesh for web are all on [GitHub](#).

CHAPTER 7

Contact

ken [at] culturemesh [dot] com