

---

**Traveller**

**FlaskCon**

**Jan 17, 2022**



# CONTENTS

<b>1</b>	<b>User Guide</b>	<b>3</b>
1.1	Documentation . . . . .	3
1.2	Readme . . . . .	3
1.2.1	Setup . . . . .	3
1.2.2	Setup Mail Dev Environment . . . . .	5
1.2.3	Running tests . . . . .	6
1.2.4	Some functionalities of app . . . . .	6
<b>2</b>	<b>Indices and tables</b>	<b>7</b>



Traveller is a conference software written for FlaskCon



## 1.1 Documentation

Sphinx and the other needed packages are included in **traveller/sphinx\_source/docs\_requirements.txt**.

To setup Sphinx:

Install the required packages using

```
$ python -m pip install -r docs_requirements.txt
```

then change directory to the folder containing the source and build files for the documentation

```
$ cd traveller/sphinx_source
```

Run the command below in the traveller/sphinx\_source folder to generate HTML pages for the documentation.

```
$ make html
```

You can view the generated HTML pages in **\_build/html** folder

## 1.2 Readme

### 1.2.1 Setup

Create venv named venv inside root folder

Activate it

Install requirements.txt

```
$ python -m pip install -r requirements.txt
```

You may also want to install dev\_requirements.txt

```
$ python -m pip install -r dev_requirements.txt
```

We are using MySQL but you can have a stab at a different db.

Create a db named traveller or whatever you want in your MySQL db.

- Start MySQL database

```
$ systemctl start mysql
```

(or)

```
$ service mysql start
```

- After starting MySQL database, login into the shell

```
$ mysql
```

- Create a database

```
mysql > CREATE DATABASE traveller;
```

- This will create the database in your local MySQL server, you can exit the Mysql shell and complete the remaining steps

Change directory to the traveller folder

```
$ cd traveller
```

Create folder called instance and a file called config.py in it

```
$ mkdir instance #auto ignored by git  
$ touch instance/config.py
```

In instance/config.py set the **SQLALCHEMY\_URI**. For MySQL it will be like this (the file should contain only that):

```
SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://root:root@localhost/traveller'
```

'mysql+pymysql://username:password@localhost/dbname'.

Create or edit traveller/config.json with the information needed for each environment.

for **development**:

```
{  
  "environment": "development",  
  "admin_user": {  
    "email": "admin@domain.com",  
    "password": "pass"  
  },  
  "settings": {  
    "APP_NAME": "Demo",  
    "ACTIVE_FRONT_THEME": "blogus",  
    "ACTIVE_BACK_THEME": "boogle",  
    "CURRENCY": "MUR"  
  }  
}
```

and for **production**:

```
{  
  "environment": "production",  
  "admin_user": {  
    "email": "admin@domain.com",  
    "password": "pass"  
  },  
}
```

(continues on next page)



(continued from previous page)

```

"settings": {
    "APP_NAME": "Demo",
    "ACTIVE_FRONT_THEME": "blogus",
    "ACTIVE_BACK_THEME": "boogle",
    "CURRENCY": "MUR"
}
}

```

Now in traveller/traveller run:

```
$ python manage.py initialise
```

Then, to get development example data (make sure requirements in dev\_requirements.txt are installed)

```
$ flask seed dev
```

Then

```
$ python manage.py rundebug
```

Migrations:

```

$ python manage.py db migrate
$ python manage.py db upgrade

```

More info can be found in the shopyo docs: [shopyo.readthedocs.io](https://shopyo.readthedocs.io)

## 1.2.2 Setup Mail Dev Environment

We are using flask-mailman.

If you have Node.js, use the [maildev](#) package. Install it using

```
$ npm install -g maildev
```

Then serve it using

```
$ maildev
```

Dev configs for this setup are (already in config.py):

```

class DevelopmentConfig(Config):
    """Configurations for development"""

    ENV = "development"
    DEBUG = True
    LOGIN_DISABLED = False
    # control email confirmation for user registration
    EMAIL_CONFIRMATION_DISABLED = False
    # flask-mailman configs
    MAIL_SERVER = 'localhost'
    MAIL_PORT = 1025
    MAIL_USE_TLS = False
    MAIL_USE_SSL = False
    MAIL_USERNAME = '' # os.environ.get("MAIL_USERNAME")

```

(continues on next page)

(continued from previous page)

```
MAIL_PASSWORD = '' # os.environ.get("MAIL_PASSWORD")
MAIL_DEFAULT_SENDER = 'ma@mail.com' # os.environ.get("MAIL_DEFAULT_SENDER")
```

Go to <http://127.0.0.1:1080> where it serves its web interface by default. See mails arrive in your inbox! Particularly useful when registering!

### 1.2.3 Running tests

Go to traveller/traveller

```
$ python -m pytest .
```

### 1.2.4 Some functionalities of app

Go to: <http://127.0.0.1:5000/dashboard>

Login with **username: admin@domain.com** and **password: pass**

Click on admin and create a new role called reviewer

Add new people and assign them the roles

Go to dashboard and click on conf

Create a new conf

Add reviewers to conf

Go to: <http://127.0.0.1:5000/y/2021/>

We'll db seed some folks soon

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`