




# Developer's Guide

Branch: master ▾

Find file Copy path



GreenVVZ / README.md

 **bbodo** Update README.md  
02ce985 just now

2 contributors  

Raw Blame History

110 lines (92 sloc) | 7.42 KB

# Green VVZ

---

## Overview

---

GreenVVZ: A web application to facilitate storage and display of modules related to sustainability topics, based on the course catalogue of the University of Zurich.

Built for the Informatics and Sustainability Research group at the Department of Informatics of the University of Zurich, under supervision of Prof. Dr. Lorenz Hilty.

This project consists of two main parts: a flask back end, interfacing with the UZH course catalogue, as well as a flask / Jinja2 / jQuery frontend, interfacing with the flask back end.

A foreword for the following guides: Trial and error, as long as you are not running a productive server, are a great tool to learn how software works. This guide might not be complete, and the technology you run it on might not be the exact same, but it should help you figure out the necessary steps to get it running. **Should anything need changes, please create a pull request!**

## Developer's Guide for hosting the project

---

(This is needed to test any database functionality. Best bring this guide to your UZH technician, he will be able to guide you where this guide might fail.)

To set up a server hosting this tool, which will be required to implement and test significant changes, follow these steps:

1. Request a Virtual Private Server (VPS) at the Department of Informatics (or set up your own). Root privileges are not required, with exception of privileges to (re)start certain services. More to that later. Make sure the VPS comes with the following software installed:
  - A linux distribution. *UZH uses Debian.*
    - A user / production folder with read/write privileges.
    - read / write privileges to the config folders of systemd/systemctl.
    - open ports, at least port 80, 8080, 443 and 8443.
  - Git.
  - A MySQL variant. *UZH uses MariaDB.* Also a MariaDB account which can create a database, tables, and read/write to it. Standard procedure is one DB user / application. *UZH supplied the accounts for me.*
  - Python  $\geq 3.5$ .
    - pip, venv.
  - An HTTP server. *UZH uses Apache.* Make sure to request privileges to run `sudo systemctl restart apache2`.
  - A working SSL certificate.
2. In your user directory, create a virtualenv (*UZH supplied one for me*):
  - To create a local environment to host the needed python libraries, if necessary and applicable, either `python3 -m pip install virtualenv`, Or `apt-get install python3-venv`.
  - Create a virtual python environment on your VPS with either `python3 -m virtualenv venv_[choose a name]` Or `python3 -m venv venv_[choose a name]` in your user directory. For this example, let's say you chose `greenvvz` as the name.
    - This will create a folder `venv_greenvvz`
    - use `source venv_greenvvz/bin/activate` to activate the virtual env.
3. Set up this project to be served:
  - First of all, get a copy of this project on the server: `git clone https://github.com/bbodo/greenvvz.git`, to clone this repository into the folder `greenvvz`. Then `cd greenvvz` to navigate to the project directory.
  - With the virtualenv activated, use `pip install -r requirements.txt` to install all necessary packages to host the server.
  - Once you have have a database user example: `dbuser` with password `dbuserpassword`, set up a database named `dbname`, and decided on a unique secret key `yoursecretkey`, only known to you, and not saved in the repository, proceed to the next step.
  - add the following lines to the `venv_greenvvz/bin/activate_this.py` file, to make sure the server has the necessary information:

```
os.environ["FLASK_ENV"]="development" # Comment line out for production!
os.environ["DB_USER"]="dbuser"
```

```
os.environ["DB_PASSWORD"]="dbuserpassword"
os.environ["DB_NAME"]="dbname"
os.environ["SECRET_KEY"]="yoursecretkey"
```

4. Set up the HTTP server to serve files using the Web Server Gateway Interface (WSGI) in your user/www directory. For this example, let's say your username is `USER`.

- Put the application.wsgi file there, with the following contents(*the helpful techs at UZH did this for me*):

```
import sys

activate_this = '/home/USER/venv_greenvvz/bin/activate_this.py'
with open(activate_this) as file_:
    exec(file_.read(), dict(__file__=activate_this))

sys.path.insert(0, '/home/USER/greenvvz')
from main import app as application
```

5. After this, you should be good to go! Check out if your server is running at <https://yourserver.ifi.uzh.ch>! If you run into errors, make sure to check out the logs you set up in step 4, or UZH kindly set up for you.

## Developer's Guide for a local setup

1. Make sure you have git and python  $\geq 3.5$  installed.
2. Set up a local dev environment in a folder of your liking: `python3 -m virtualenv venv_[choose a name]` Or `python3 -m venv venv_[choose a name]`. For this example, let's say you chose `greenvvz` as the name.
  - This will create a folder `venv_greenvvz`.
  - use `venv_greenvvz/bin/activate` to activate the virtual env.
3. Git clone this project to a folder of your liking: `git clone https://github.com/bbodo/greenvvz.git`.
4. Activate the virtualenv, and navigate to the project folder. `python 3 -m pip install -r requirements.txt` to get all necessary packages.
5. Add the following information to your environment, either using the virtualenv file or manually (this depends on your system, google how to add environmental variables):

```
os.environ["FLASK_ENV"]="development"
os.environ["DB_USER"]="dbuser"
os.environ["DB_PASSWORD"]="dbuserpassword"
os.environ["DB_NAME"]="dbname"
os.environ["SECRET_KEY"]="yoursecretkey"
```

5. Once this is done, you are good to go! Use `flask run` to use your local server on localhost:5000!

## Guide to set up a working instance on Magnolia:

6. Now, to host your server on an UZH page in Magnolia, do the following:
- Create a page `greenvvz-admin` for the administrator view. I recommend Inhaltsseite 1-Spaltig.
    - In the Content subsection, add an HTML element with the following contents:

```
<iframe id="greenvvz-admin-iframe" class="mod mod-iframe" src="https://yourserver.ifi.uzh.ch/st

<script type="text/javascript" src="https://yourserver.ifi.uzh.ch/st

<script>
  iFrameResize({
    log: false,
    // heightCalculationMethod: 'max',
  },
  '#greenvvz-admin-iframe')
</script>
```

- Create a page `greenvvz-public` for the public view
  - In the Content subsection, add an HTML element with the following contents:

```
<iframe id="greenvvz-public-iframe" class="mod mod-iframe" src="http

<script type="text/javascript" src="https://yourserver.ifi.uzh.ch/st

<script>
  iFrameResize({
    log: false,
    // heightCalculationMethod: 'max',
  },
  '#greenvvz-public-iframe')
</script>
```