08

**Fall**

**FSGIM Implementation Support**

**Contract No. SB1341-14-CQ-0043/ Order No.:16-443**



**FSGIM Weather Client Application (Simple Client)**

**Draft v1**

**April 19, 2018**

**Prepared by**

**ESTA International, LLC**

**2214 Rock Hill Road, Suite 180**

**Herndon, Virginia, 20170-4234, USA**



[www.ESTAInternational.com](http://www.ESTAInternational.com)

ESTA International, LLC has prepared this document under contract to NIST under contract number SB1341-14-CQ-0043/ order no.:16-443. Unless otherwise noted all information in this document is confidential and proprietary.

This work was completed through close cooperation between

NIST and ESTA International Project Team members.

NIST Project Team Members includes:

Dr. Martin J. Burns Project Technical Lead

Mr. Paul Boynton Project Manager

ESTA team members for this project include:

Mr. Nader Farah Project Manager

Mr. Allen J. Jones FSGIM Model Subject Matter Expert

Mr. Donald F. Coffin OpenESPI/ Green Button Subject Matter Expert

Mr. Abhinav Verma Software Developer

Please address all inquiries to:

Mr. Nader Farah

ESTA International, LLC

[nader.farah@ESTAinternational.com](mailto:nader.farah@ESTAinternational.com)

Table of Contents

[1 Overview 3](#_Toc512246228)

[1.1 Objective 3](#_Toc512246229)

[1.2 Application Output 3](#_Toc512246230)

[1.3 Versioning 3](#_Toc512246231)

[2 Application Structure 4](#_Toc512246232)

[3 Supported Webpages and URL’s 6](#_Toc512246233)

[3.1 Login Page 6](#_Toc512246234)

[3.1.1 Creating admin user 7](#_Toc512246235)

[3.1.1.1 Windows 7](#_Toc512246236)

[3.1.1.2 Linux 8](#_Toc512246237)

[3.2 FSGIM Endpoint Access page 9](#_Toc512246238)

[3.3 Django Administration 10](#_Toc512246239)

[4 Setup 10](#_Toc512246240)

[4.1 Linux 10](#_Toc512246241)

[4.2 Windows 12](#_Toc512246242)

[5 References 14](#_Toc512246243)

Table of Figures

[Figure 1: Application Structure 4](https://estainternationalllc-my.sharepoint.com/personal/abhinav_verma_estainternational_com/Documents/NIST/FSGIM%20Simple%20Client%20Delivery%20Document.docx#_Toc512248326)

[Figure 2: Supported url patterns in urls.py in 'fsgim\_oauth' package 4](https://estainternationalllc-my.sharepoint.com/personal/abhinav_verma_estainternational_com/Documents/NIST/FSGIM%20Simple%20Client%20Delivery%20Document.docx#_Toc512248327)

[Figure 3: Static package inside 'fsgim\_oauth' package 5](#_Toc512248328)

[Figure 4: HTML pages 5](#_Toc512248329)

[Figure 5: Supported URI Table 6](#_Toc512248330)

[Figure 6: Login page 6](#_Toc512248331)

[Figure 7: Windows Command Prompt Window 7](#_Toc512248332)

[Figure 8: Create Super User Command 7](#_Toc512248333)

[Figure 9: Successfully created Admin user 8](#_Toc512248334)

[Figure 10: Linux Terminal Window 8](#_Toc512248335)

[Figure 11: Create Super User in Linux 8](#_Toc512248336)

[Figure 12: FSGIM endpoint access page 9](#_Toc512248337)

[Figure 13: Django Administration page 10](#_Toc512248338)

[Figure 14: Create & Navigate into 'nist' directory 10](#_Toc512248339)

[Figure 15: Create, navigate and activate Virtual Environment 11](#_Toc512248340)

[Figure 16: Install Django 11](#_Toc512248341)

[Figure 17: Clone git repo and start local server 11](#_Toc512248342)

[Figure 18: Setup Simple Client 12](#_Toc512248343)

[Figure 19: Create and Navigate into the directory 'NIST' 12](#_Toc512248344)

[Figure 20: Clone git repository 12](#_Toc512248345)

[Figure 21: Navigate and run server on Windows 13](#_Toc512248346)

# Overview

## Objective

The purpose of ‘OpenFSGIM-SimpleClient’ is to consume the endpoints of the Facilities Smart Grid Information Model (FSGIM) Weather Web Server. This application solves this purpose by creating a website using one of the most popular framework called Django[[1]](#footnote-2). Django provides a framework for development of this website. Django is fast, secure and scalable.

## Application Output

The output of this application is a website; this website is secured by user credentials and demonstrates the GET and POST functionality of the FSGIM weather web server.

## Versioning

This application uses Python version 3.5[[2]](#footnote-3). Django framework version 2.0.1 is used for the development of the website.

The interactive development environment used to develop this application is Pycharm Python IDE 2017.3.1 (Community Edition)[[3]](#footnote-4).

The ‘Simple Client’ also uses Bootstrap 3.3.7[[4]](#footnote-5) for styling the website.

# Application Structure

The application consists of a ‘weather\_client\_app’ package which consists of two packages: ‘fsgim\_oauth’ and ‘weather\_client\_app’.

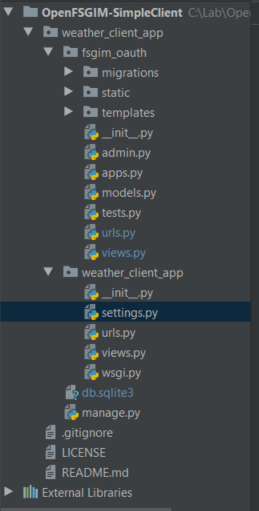
This application also uses a DBL SQL Lite database to store user credentials and permissions.

Figure : Application Structure

‘weather\_client\_app’ package inside the ‘weather\_client\_app’ package consists of three main files:

1. Settings.py
2. Urls.py
3. Views.py

‘fsgim\_oauth’ package is generated by the Django framework and share the structure with ‘weather\_client\_app’ package. ‘fsgim\_oauth’ package has the following main files and packages:

1. Urls.py
2. Views.py
3. Static package
4. Template package

Urls.py contains the urls that this package supports and controls (Refer to Figure 2). Refer to section 3 for the list of all the URL’s supported by this application.

Views.py contains the logic which controls this application.

This views.py file contains functions which provide routing and links between different webpages.

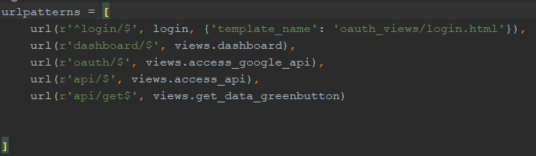


Figure : Supported url patterns in urls.py in 'fsgim\_oauth' package

Static package consists of the styling options provided by the Bootstrap version 3.3.7. The Bootstrap styling options are present in the three packages at the bottom of the figure 3: “css, fonts and js”.

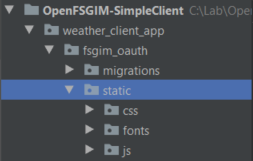


Figure 3: Static package inside 'fsgim\_oauth' package

The template package contains another package called ‘oauth\_views’ which contains all the HTML pages used by this application as shown in figure 4.

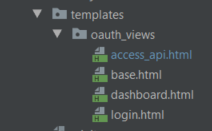


Figure 4: HTML pages

# Supported Webpages and URL’s

The Simple Client supports the following URL’s:

|  |  |
| --- | --- |
| **URL** | **Description** |
| /fsgim/login/ | Login page |
| /fsgim/dashboard/ | Dashboard page: This page is a mediatory page. It has a welcome message and a next button. The next button starts the Oauth flow. |
| /fsgim/oauth/ | Internal URI, used to send an oauth request to the server |
| /fsgim/api/ | FSGIM GET and POST endpoint access page |
| /fsgim/api/get/ | Internal URI: It is used after a successful request has been fulfilled by the oauth URI |
| /admin | django user administration website |

Figure 5: Supported URI Table

## Login Page

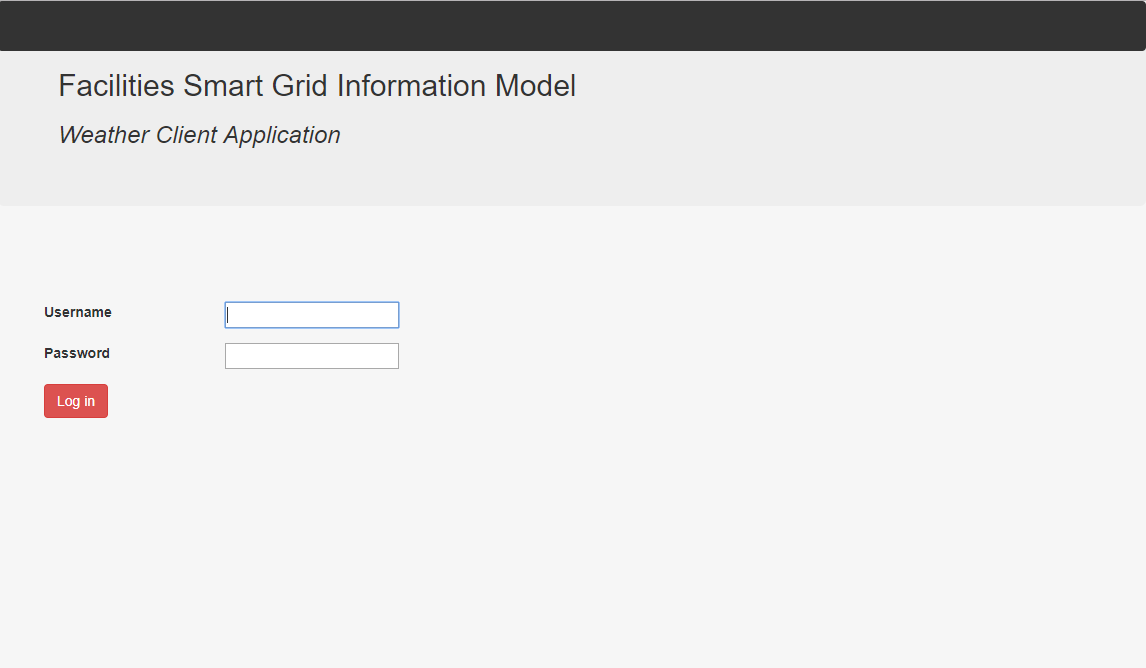


Figure 6: Login page

Login page consists of the username and password input form. The first user for this application can be created by using the super user command of the Django framework[[5]](#footnote-6). The first user to this site should be created using the Section 3.1.1. All the other user credentials can be created by Django user administrative page. (URI: /admin)

### Creating admin user[[6]](#footnote-7)

Please follow the following steps to create an administrative user after installing this application.

#### Windows

Steps to create an administrative user in windows:

1. Open ‘Command Prompt’
2. Navigate to the folder where the git repository is cloned or downloaded.
3. Navigate into ‘weather\_client\_app’, the command prompt window should look like the command prompt window show in the Figure 7

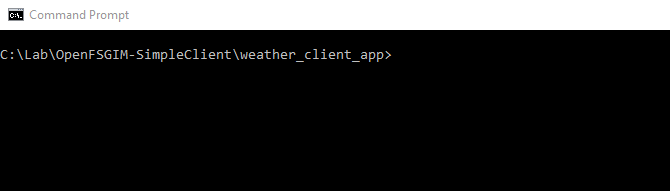


Figure 7: Windows Command Prompt Window

1. Once you are in the correct directory, type the following command (Refer to Figure 8)

Python manage.py createsuperuser

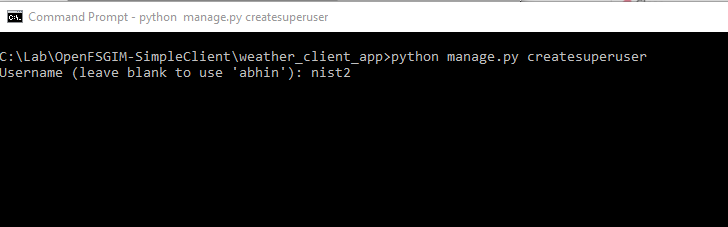


Figure 8: Create Super User Command

1. After typing the above command, press enter, the system is going to ask for the following parameters:
   1. Username: <enter any username of your choice>
   2. Email address: <enter email address or leave it blank>
   3. Password: <enter your password here>
   4. Password (again): <enter your password here again>

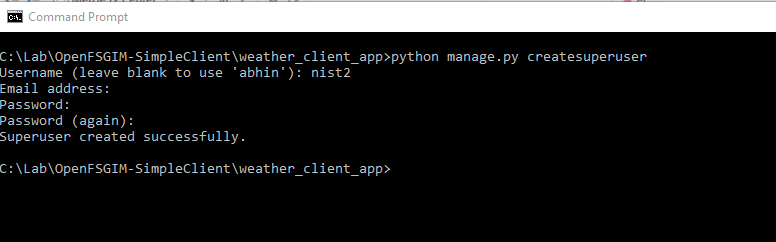


Figure 9: Successfully created Admin user

1. A message will be received after the user has been created, refer to Figure 9
2. Use these user credentials to log into the Figure 6: Login page

#### Linux

This process is similar to the process discussed in the above section (3.1.1.1). Please follow the following steps to create a super user with administrator privileges in a Linux environment as follows:

* + - 1. Open the terminal window
      2. Navigate to the folder where git repository is cloned or downloaded.
      3. Navigate into ‘weather\_client\_app’, the terminal window will look like Figure 10

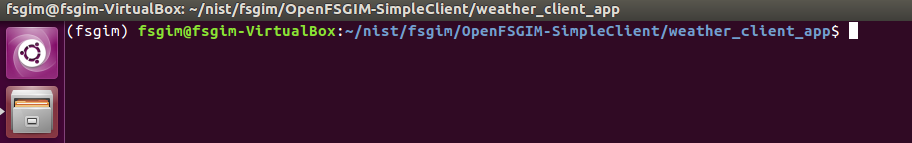


Figure 10: Linux Terminal Window

* + - 1. Type the following command to create a super user (Refer to Figure 11)

Python manage.py createsuperuser

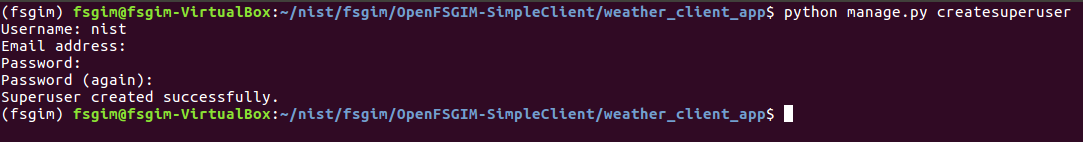


Figure 11: Create Super User in Linux

* + - 1. After typing the above command, press enter, the system is going to ask for the following parameters:

1. Username: <enter any username of your choice>
2. Email address: <enter email address or leave it blank>
3. Password: <enter your password here>
4. Password (again): <enter your password here again>
   * + 1. A message will be received after the user has been created, refer to Figure 9
       2. Use these user credentials to log into the Figure 6: Login page

## FSGIM Endpoint Access page

The endpoint access page is used to access the endpoints of FSGIM Weather Web server. The endpoint access page can be reached by using the URI (/fsgim/api) or by clicking on the ‘next’ button on the dashboard page (URI: /fsgim/dashboard/). Refer to Figure 12

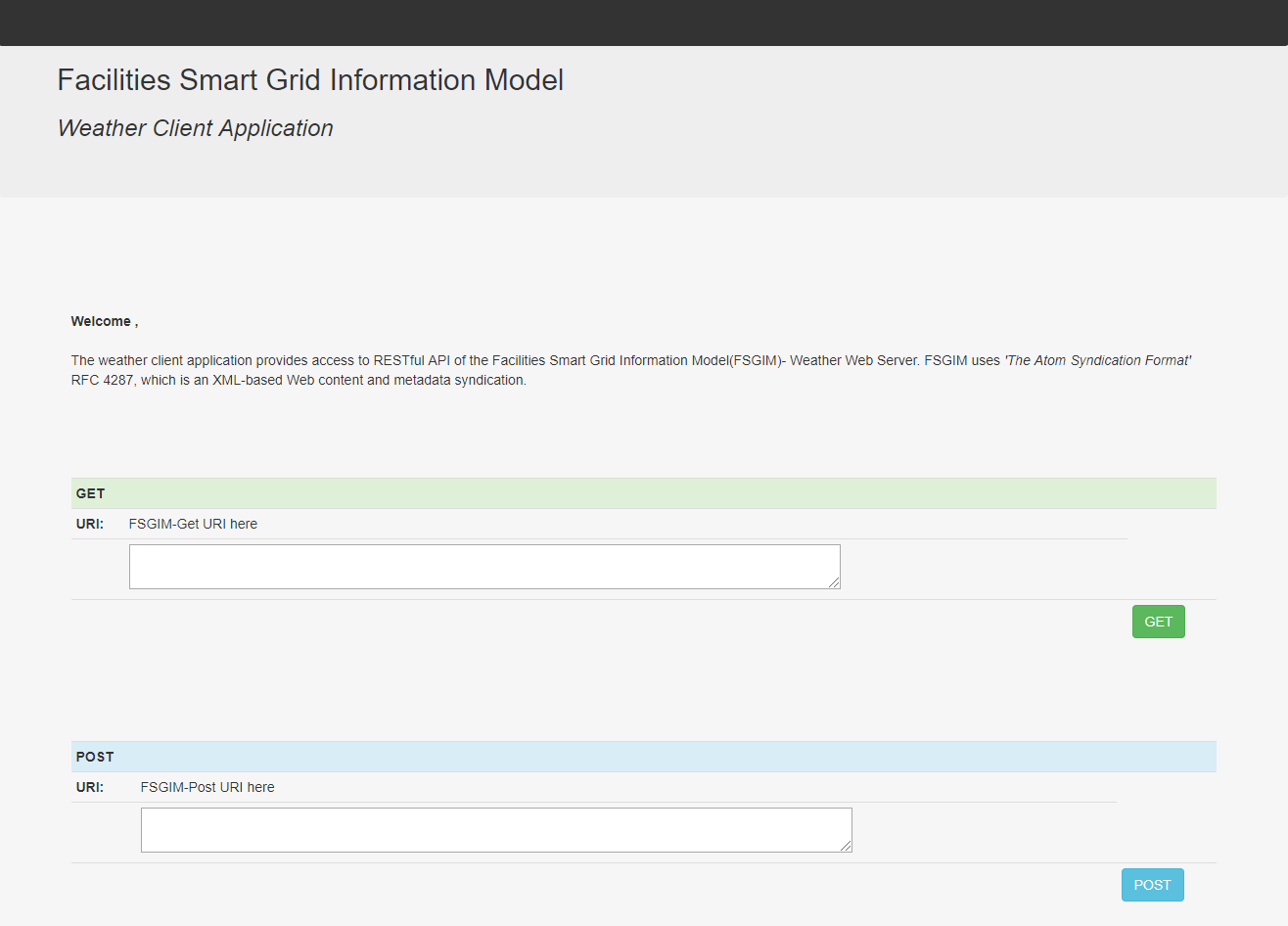


Figure 12: FSGIM endpoint access page

This page has both GET and POST functionality for the FSGIM Weather Web Server.

## Django Administration

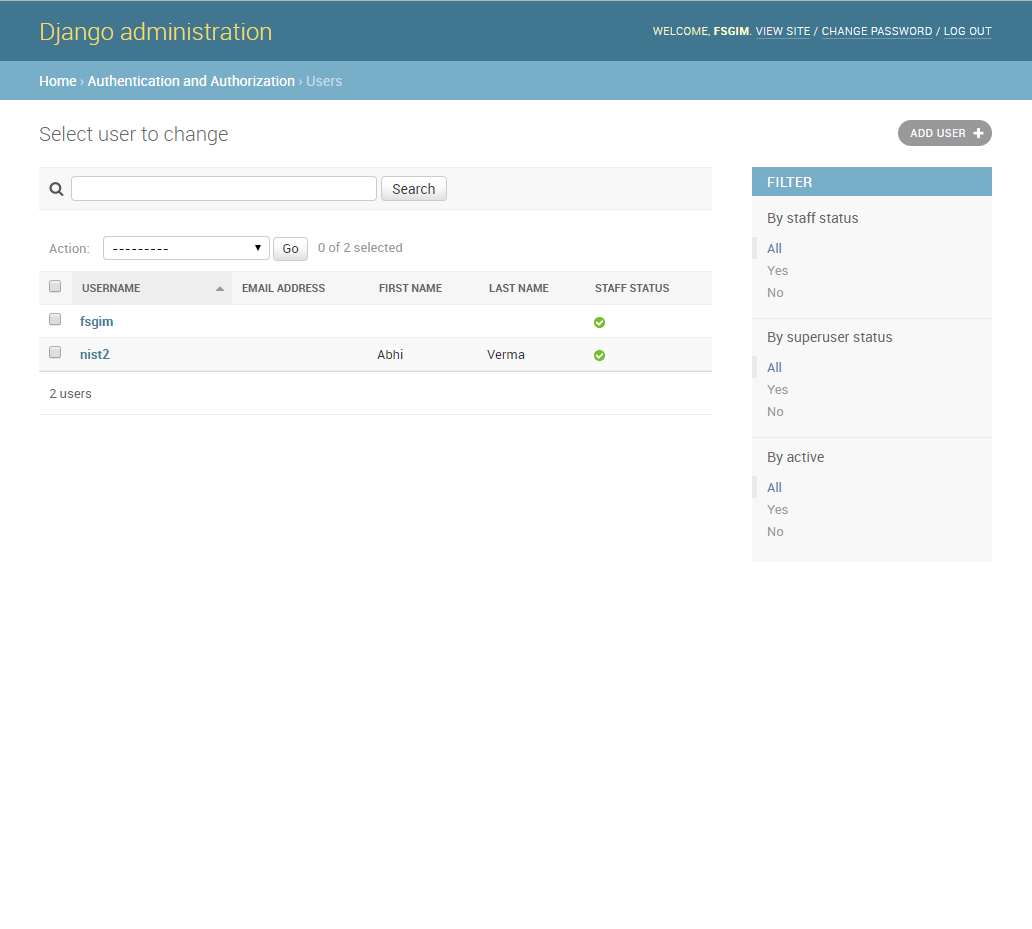
This page can be used to create more users. Different permissions can be assigned to different users. Refer to Figure 13

Figure 13: Django Administration page

# Setup

This section defines the steps to install and run this application on Windows and Linux platforms. The best practice to run any python application is to have a virtual environment. Virtual environment as the name suggests is an environment that can have different versions of the same python libraries on the same machine.[[7]](#footnote-8) In this demonstration, the windows installation is done without setting up a virtual environment and Linux installation is done with a virtual environment.

## Linux

Please refer to the following steps to setup Simple client on your Linux environment:

Create a new directory call ‘nist’ using the following command:

Navigate into the new directory ‘nist’



Figure 14: Create & Navigate into 'nist' directory

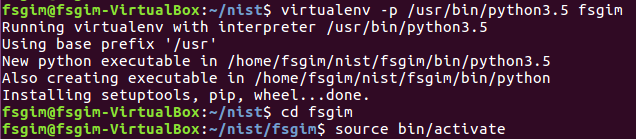
Create a virtual environment

Figure 15: Create, navigate and activate Virtual Environment

Navigate into the virtual environment

Activate the virtual environment

Install Django 2.0.4

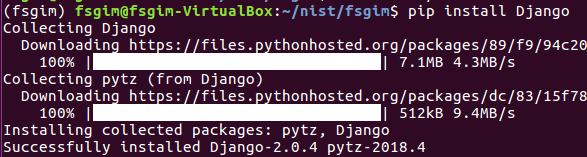


Figure 16: Install Django

Clone the git repository

Navigate into ‘OpenFSGIM-SimpleClient’

Navigate into ‘weather\_client\_app’

Run the server

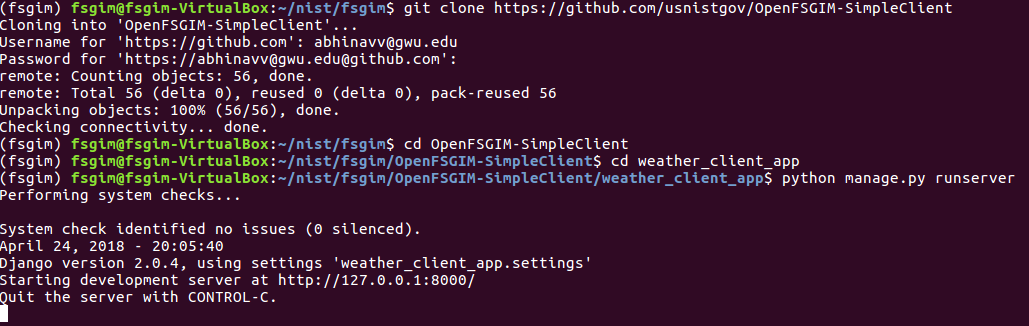


Figure 17: Clone git repo and start local server

Go to the browser and use this link http://localhost:8000/fsgim/login/

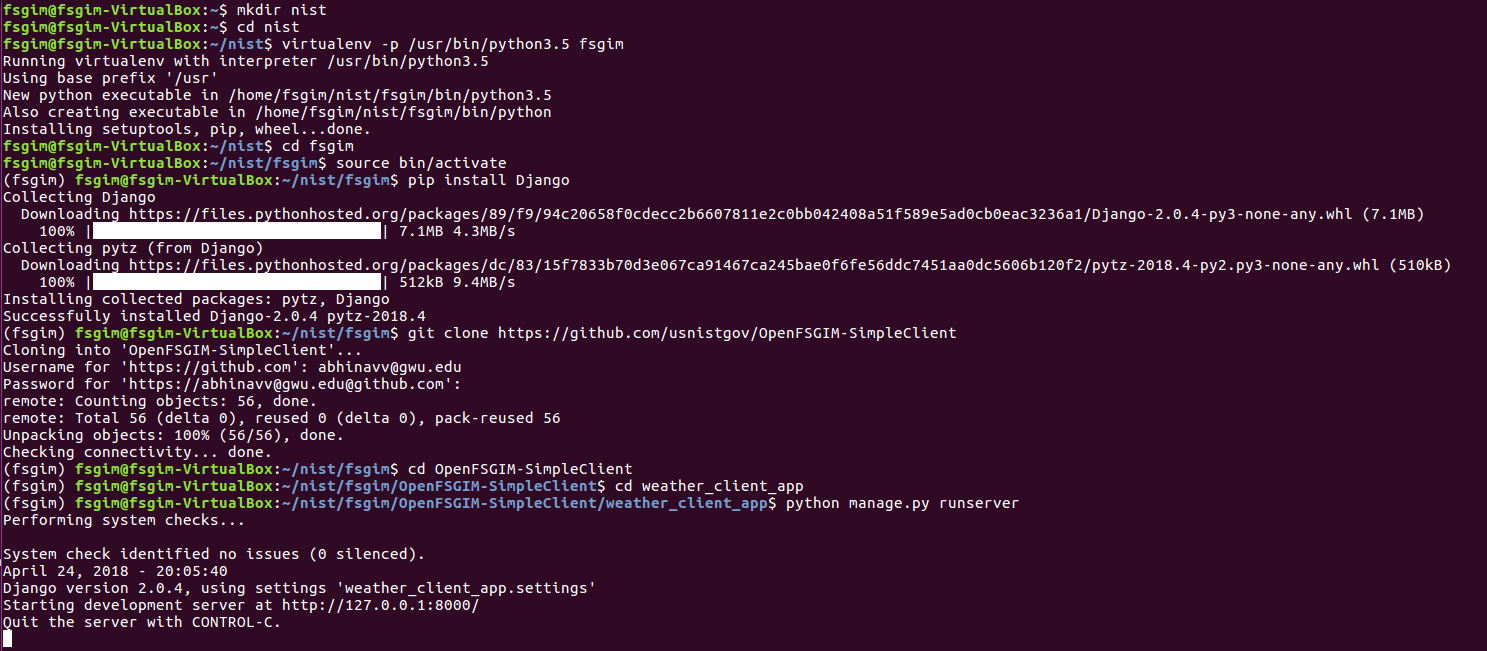


Figure 18: Setup Simple Client

## Windows

The steps will be similar to the steps discussed in the Linux section above. The windows setup in this document is done without setting up the virtual environment. It is recommended to use the virtual environment to run the Simple Client. Before using the following steps, please ensure that Python 3.5 and Django 2.0.1 is installed on your machine.

1. Create a new directory
2. Navigate into the new directory

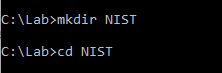


Figure 19: Create and Navigate into the directory 'NIST'

1. Clone the git repository

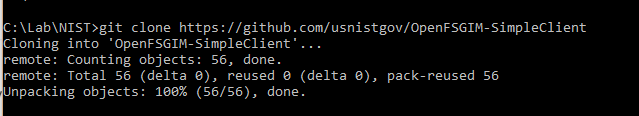


Figure 20: Clone git repository

1. Navigate into ‘OpenFSGIM-SimpleClient’
2. Navigate into ‘weather\_client\_app’
3. Run the server

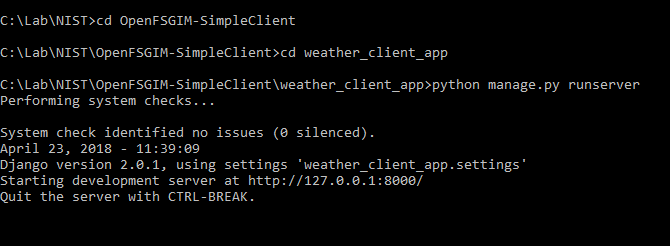


Figure 21: Navigate and run server on Windows

1. Go to the browser and use this link: http://localhost:8000/fsgim/login/

# References

|  |  |
| --- | --- |
| **Resource** | **Link** |
| Django framework | <https://www.djangoproject.com/> |
| Python 3.5 | https://www.python.org/downloads/release/python-350/ |
| Pycharm Python IDE 2017.3.1 (Community Edition) | https://www.jetbrains.com/pycharm/download/#section=windows |
| Bootstrap 3.3.7 | https://getbootstrap.com/ |
| Django- Create admin user documentation | https://docs.djangoproject.com/en/2.0/intro/tutorial02/#creating-an-admin-user |
| Python Virtual Environment Documentation | https://docs.python.org/3/tutorial/venv.html |

1. https://www.djangoproject.com/ [↑](#footnote-ref-2)
2. https://www.python.org/downloads/release/python-350/ [↑](#footnote-ref-3)
3. https://www.jetbrains.com/pycharm/download/#section=windows [↑](#footnote-ref-4)
4. https://getbootstrap.com/ [↑](#footnote-ref-5)
5. https://docs.djangoproject.com/en/2.0/intro/tutorial02/#creating-an-admin-user [↑](#footnote-ref-6)
6. https://docs.djangoproject.com/en/2.0/intro/tutorial02/#creating-an-admin-user [↑](#footnote-ref-7)
7. https://docs.python.org/3/tutorial/venv.html [↑](#footnote-ref-8)