

Создание сервера Django в редакторе Atom

Романов Даниил Алексеевич

Приамурский государственный университет имени Шолом-Алейхема

студент

Аннотация

Целью данной статьи является создание сервера с помощью языка программирования Python и его библиотеки Django. Проект выполняется в текстовом редакторе Atom. Результатом исследования станет готовый веб-сервер страницы, на которой можно будет добавить внешний вид и функционал.

Ключевые слова: сервер, Python, Django, Atom

Creating a Django server in the Atom Editor

Romanov Daniil Alekseevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to create a server using the python programming language and its Django library. The project is executed in the Atom text editor. The result of the study will be a ready-made web server of the page on which it will be possible to add appearance and functionality.

Keywords: server, Python, Django, Atom

1 Введение

1.1 Актуальность

На сегодняшний день почти каждому программисту, работающему в сфере web-сайтов необходимо уметь создавать и подключать сервер, на основе которого можно создавать web-приложения и сайты.

1.2 Обзор исследований

В своей работе Д. А. Ковалёв и В. Н. Татищева описывают методы создания веб-сервера на основе библиотеки Django для веб-приложения [1]. Схожими вопросами занималась и К. М. Рябова [2]. В. В. Романченко и Л. Н. Амосова делали обзор на текстовый редактор Atom и его использование для создания веб-сайта [3].

1.3 Цель исследования

Цель исследования - создать и понять принцип работы web-сервера, написанного на языке программирования Python с использованием библиотеки Django, в текстовом редакторе Atom.

2 Материалы и методы

Для создания программы нам потребуется несколько вещей. Во-первых, это текстовый редактор Atom скачать который вы сможете по ссылке [4]. Во-вторых, язык программирования Python [5] и библиотека Django [6].

3 Результаты и обсуждения

После того как вы установили Atom, Python и Django, необходимо зайти в Atom и установить плагин терминала. Для этого нужно зайти в Preferences(настройки), Install(установить) и с помощью поисковой строки устанавливаете плагин platformio-ide-terminal (рис.1). После установки желательно перезапустить Atom и внизу вы увидите плюсики, который будет выдвигать терминал.

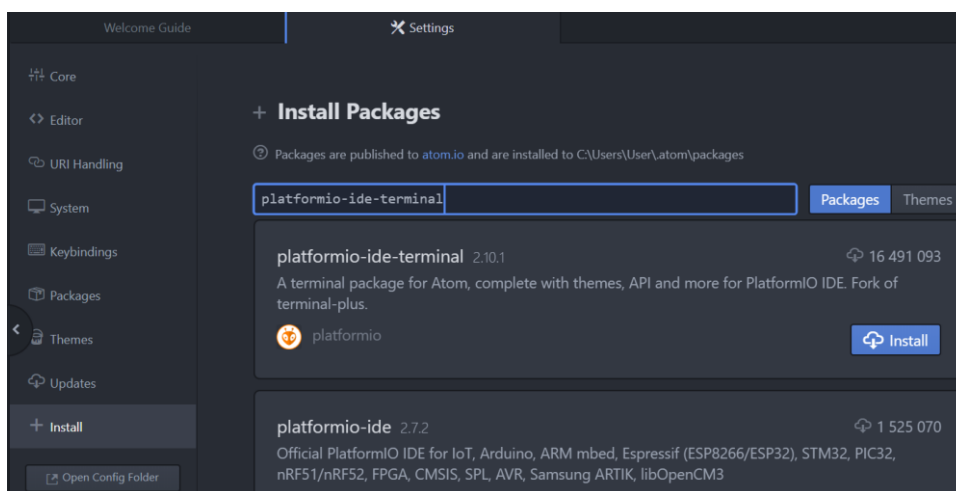


Рисунок 1 - Установка дополнительного плагина

После установки желательно перезапустить Atom и внизу вы увидите плюсики, который будет выдвигать терминал (рис.2).

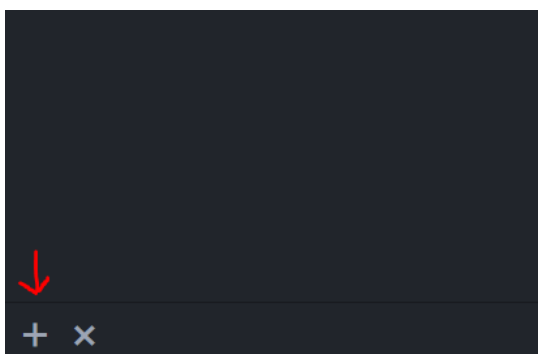


Рисунок 3 - Значок терминала в нижнем левом углу

Далее необходимо создать django-проект, для этого прописываем команду “django-admin startproject WeatherApp(название проекта)” (рис.3).

```
django-admin startproject WeatherApp
```

Рисунок 3 - Команда создания django-проекта

Затем переходим внутрь этой папки, прописывая в терминале “cd WeatherApp” (рис.4).

```
PS D:\> cd WeatherApp
PS D:\WeatherApp> |
```

Рисунок 4 - Команда перехода внутрь проекта

В правом верхнем углу нажимаем на File, Add Project Folder и выбираем наш проект WeatherApp. Таким образом открывается проект внутри Atom (рис.5).

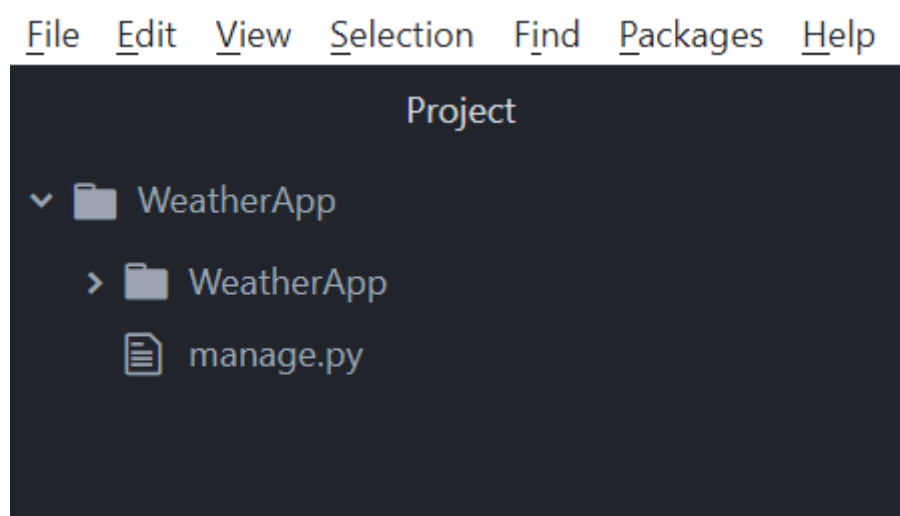


Рисунок 5 - Открытие проекта внутри Atom

Теперь необходимо создать приложение внутри django-проекта. Это приложение будет выдавать информацию о погоде в разных городах мира. Для этого прописываем в терминале команду “python manage.py startapp weather(приложение, которое мы создаём)” (рис.6).

```
PS D:\WeatherApp> python manage.py startapp weather
PS D:\WeatherApp> |
```

Рисунок 6 - Команда создания приложения

Внутри проекта WeatherApp добавится папка weather(рис.7).

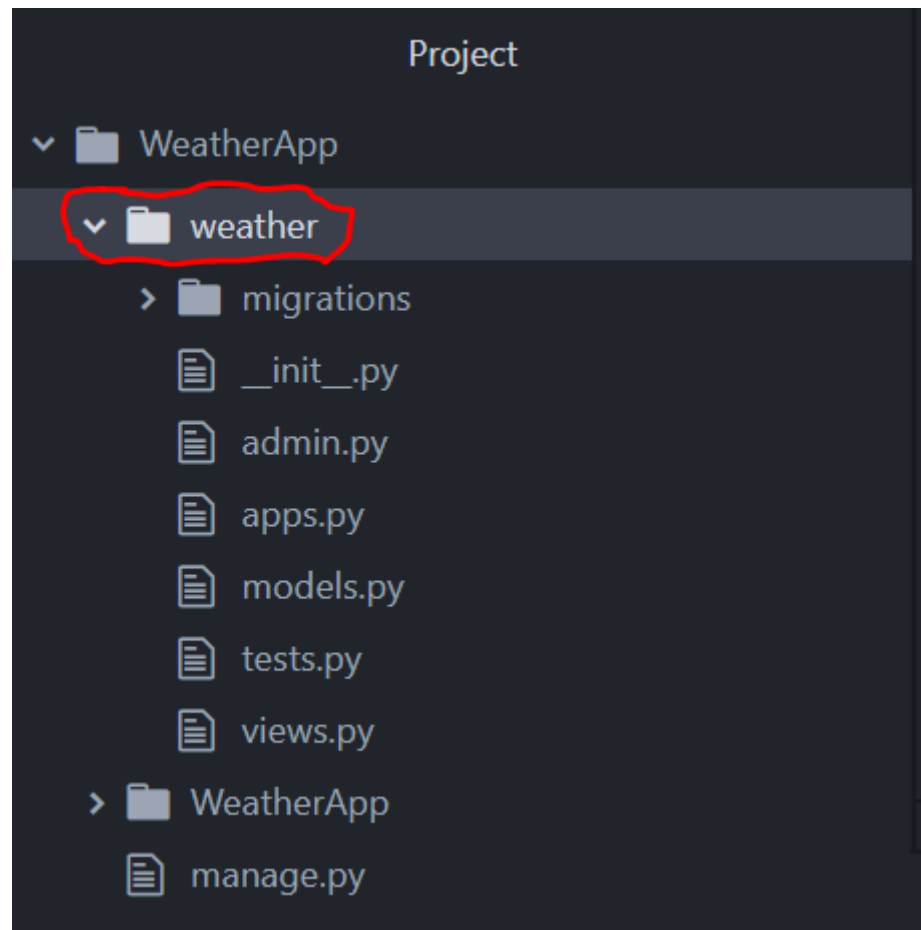


Рисунок 7 - Папка weather внутри проекта WeatherApp

Каждый раз создавая новое приложение его необходимо регистрировать в файле settings.py (рис.8).

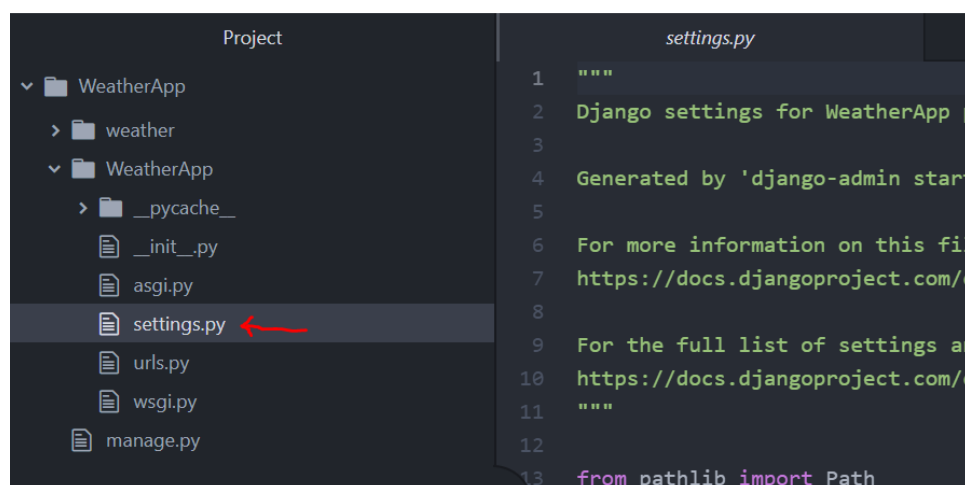


Рисунок 8 - Файл settings.py

Внутри списка INSTALLED_APPS нужно прописать новое приложение, которое будет использоваться внутри django-проекта (рис.9).

```
33  INSTALLED_APPS = [  
34      'django.contrib.admin',  
35      'django.contrib.auth',  
36      'django.contrib.contenttypes',  
37      'django.contrib.sessions',  
38      'django.contrib.messages',  
39      'django.contrib.staticfiles',  
40      'weather',  
41  ]
```

Рисунок 9 - Добавление в список нового приложения

Так же дополнительно нужно создать супер-юзера (пользователь админ-панели). Для этого в терминале прописываем команду миграции (рис.10).

```
PS D:\WeatherApp> python manage.py migrate
```

Рисунок 10 - Команда миграции

Если всё работает корректно, то выполнятся миграции (рис.11).

```
Applying auth.0002_alter_permission_name_max_length... OK  
Applying auth.0003_alter_user_email_max_length... OK  
Applying auth.0004_alter_user_username_opts... OK  
Applying auth.0005_alter_user_last_login_null... OK  
Applying auth.0006_require_contenttypes_0002... OK  
Applying auth.0007_alter_validators_add_error_messages... OK  
Applying auth.0008_alter_user_username_max_length... OK  
Applying auth.0009_alter_user_last_name_max_length... OK  
Applying auth.0010_alter_group_name_max_length... OK  
Applying auth.0011_update_proxy_permissions... OK  
Applying auth.0012_alter_user_first_name_max_length... OK  
Applying sessions.0001_initial... OK  
PS D:\WeatherApp>
```

Рисунок 11 - Миграции

Далее вводим команду создания супер-юзера и заполняем характеристики: никнейм, почта, пароль (рис.12).

```
PS D:\WeatherApp> python manage.py createsuperuser
Username (leave blank to use 'user'): master
Email address: danilka.romanov.2018@list.ru
Password:
Password (again):
Superuser created successfully.
PS D:\WeatherApp> |
```

Рисунок 12 - Создание супер-юзера

Далее следует запустить сервер (рис.13).

```
PS D:\WeatherApp> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 26, 2022 - 15:19:01
Django version 4.0.1, using settings 'WeatherApp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
|
```

Рисунок 13 - Команда запуска сервера

Чтобы проверить работает ли сервер, переходим в браузер и в поисковой строке пишем “localhost 8000”. После чего должна отобразиться страница сервера (рис.14).

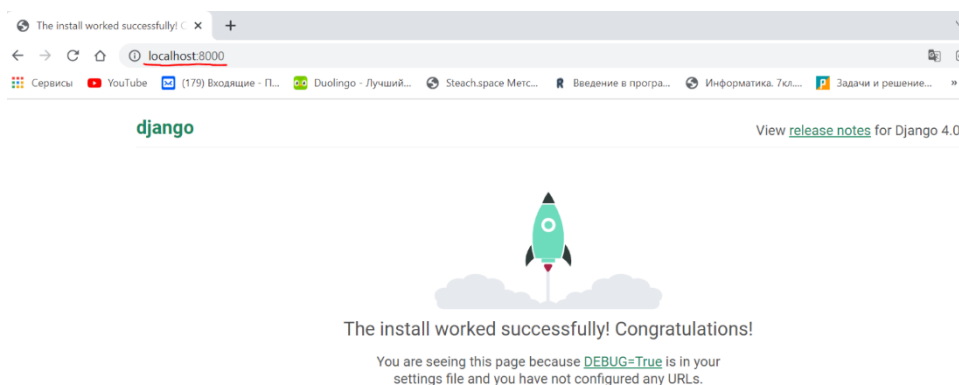


Рисунок 14 - Страница сервера

Не лишним будет проверить страницу админа. Для этого в поисковой строке прописываем “localhost 8000/admin”, после чего должна высветиться панель входа, где нужно ввести логин и пароль (рис.15).

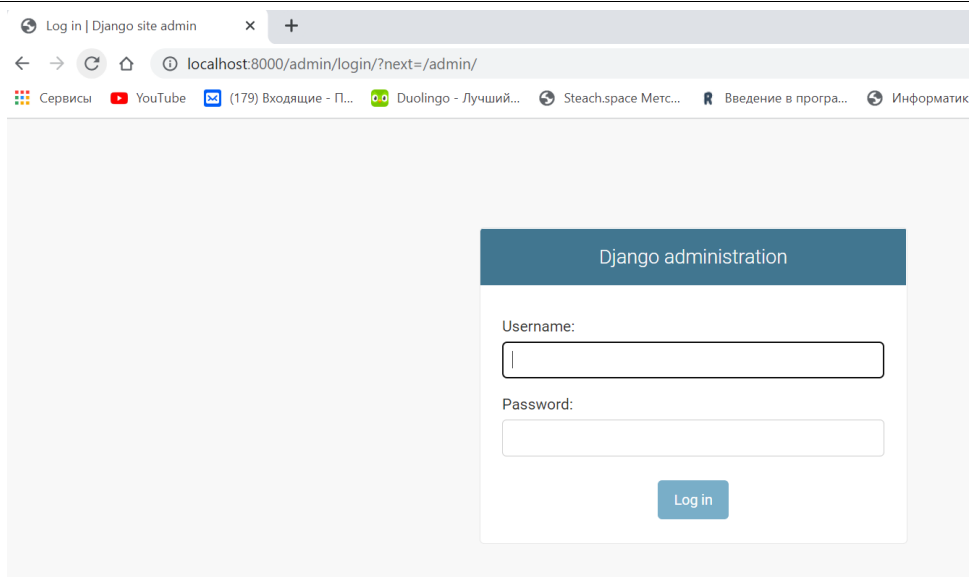


Рисунок 15 - Панель входа

После входа мы попадём в панель администратора (рис.16).



Рисунок 16 - Панель администратора

Выводы

В данной работе была создана серверная страница сайта, которая является фундаментом для создания полноценного веб-приложения.

Библиографический список

1. Ковалев Д. А. Методы и средства разработки электронных учебников. Технология Django для веб-приложений на языке Python //Вестник Волжского университета им. ВН Татищева. 2009. №13. С. 106-110.
2. Рябова К. М. Разработка веб-сайта с использованием фреймворка Django на языке Python //Инновационные научные исследования: теория, методология, практика. 2018. С. 96-99.
3. Романченко В. В., Амосова Л. Н. Обзор существующих технологий для создания web-сайта //Вестник научного общества студентов, аспирантов и молодых ученых. 2019. №2. С. 171-177.
4. Atom URL: <https://atom.io>
5. Python URL: <https://www.python.org/downloads>
6. Django URL: <https://www.djangoproject.com>