

# ВИКОРИСТАННЯ ФРЕЙМВОРКУ DJANGO ДЛЯ РОЗРОБКИ WEB-САЙТІВ

Крохін І. В., Абдулов О. Р.

*ДДМА, м. Краматорськ*

Django - фреймворк, що розповсюджується на вільній основі та дозволяє розробляти сайти на мові Python. Як більшість сучасних веб-фреймворків, Django використовує шаблон MVC (Model-View-Controller). У порівнянні з іншими PHP-фреймворками (Yii2, Laravel, Symfony), які використовуються на серверній стороні, Django фактично має всі ті переваги і недоліки, які має мова програмування Python перед мовою програмування PHP, тому в даному випадку постає вибір між мовами. Якщо порівнювати Django з іншим відомим Python-фреймворком - Flask'ом, Django утримує планку лідерства в популярності, однак для великих проєктів рекомендується використовувати Flask, через слабку реалізацію ORM (Object-Relational Mapping - об'єктно-реляційне відображення) в Django.

Для створення сайту або додатку на Django, нам буде потрібні:

- інтерпретатор Python версії 3.x;
- встановлений Django;
- редактор текстових файлів (Sublime Text, Notepad ++);
- IDE для роботи з Python-модулями (PyCharm, PyScripter, Eclipse з плагіном PyDev).

Встановити Python можна з офіційного сайту шляхом скачування потрібної версії, відповідно до своєї ОС і її розрядності. На додаток до стандартного пакету Python, починаючи з версій 2.7.9/3.4, в комплекті має встановитися менеджер пакетів pip. Далі, через менеджер пакетів pip завантажити модуль virtualenv для створення віртуального середовища python. Віртуальне середовище не є невід'ємною частиною проєкту на Django, проте все ж її рекомендується використовувати, оскільки вона дозволяє розмежувати відносини між інтерпретатором і використовуваними пакетами. Таким чином, якщо ми маємо два проєкти, які працюють на різних версіях Django, за допомогою віртуального середовища ви зможете уникнути конфліктів версії пакетів на одній машині. Створюємо робочу директорію в тому місці, де необхідно, і відкриємо її в терміналі.

Встановлення Django виконується командою:

```
pip install django ~ = 1.11.0
```

Як видно, в команді вказана версія фреймворку. Можна її не вказувати, в такому випадку завантажиться більш нова версія.

Для створення проєкту Django, в директорії з віртуальної середньої щільності (папка venvdir) в терміналі виконаємо наступну команду:

```
django-admin.exe startproject site.
```

де site - назва проєкту і папки з файлами. Крапка в кінці команди - показчик на поточну директорію, тобто в якій буде розгорнуто проєкт. В папці site у файлі settings.py необхідно внести зміни у змінну STATIC\_URL. Відразу після її ініціалізації додамо нову:

```
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
```

Тим самим ми вказали, що статичні файли для нашого сайту (картинки, файли стилів, шрифти та ін.) будуть лежати в каталозі `\static`. Варто звернути увагу на змінну-прапор `Debug` в цьому ж файлі. Її значення за замовчуванням має бути `true`. Цей прапор, відповідно до назви, говорить нам про те, що перебуває в режимі дебаггера, який фактично заміняє всі функції сервера, а також обробляє виключення. Також варто відзначити, статичні файли будуть довантажувати в шаблон і коректно працювати, тільки якщо включений прапор `Debug`, оскільки пошук статички по шляху, який ми зазначали вище (`STATIC_ROOT`) - це робота сервера. Щоб проект нормально працював без режиму дебаггера, необхідно поставити його в зв'язці з сервером.

Створення WEB-додатку виконується за допомогою команди:

```
python manage.py startapp firstapp
```

де `firstapp` - ім'я додатку. Після створення додамо його до списку додатків в файл настройки проекту (`settings.py`). Знайдемо список `INSTALLED_APPS` і додамо в нього свій `'firstapp'`. Як видно, тут вже встановлені деякі зі стандартними програмами.

URL-адреса являють собою визначник місцезнаходження ресурсу, тобто так чи інакше URL-адресу є "шляхом до файлу", будь то IP-адреса, або шлях до директорії на жорсткому диску. URL-адреси в проекті на основі Django можна налаштувати у файлі `urls.py` (та ж папка, в якій лежить `settings.py`), який являє собою URL dispatcher.

Для налаштувань відображення необхідно відкрити файл `view.py` в папці програми, і підключимо в нього деякий вбудований функціонал:

```
from django.views.generic.edit import FormView
from django.contrib.auth.forms import UserCreationForm
```

Як видно з коду вище, ми додали `FormView` і `UserCreationForm` (тобто прості форми, які нам потрібні для створення користувачів). Тепер опишемо в цьому ж `views.py` наступний клас, який успадковується від вбудованого класу `FormView`:

Для реєстрації користувачів необхідно налаштувати базу даних, куди будуть додаватися користувачі після реєстрації. В Django є стандартна SQLite база даних, в якій визначена таблиця з зареєстрованими користувачами. Перебуваючи в кореневій папці проекту міграція бази даних виконується командою:

```
python manage.py migrate
```

Таким чином, в роботі було описано порядок дій для створення WEB-додатку з використанням фреймворку Django. Наведені основні команди для встановлення всіх необхідних пакетів. У встановленому додатку реалізовано функціонал, який дозволяє реєструвати та авторизувати користувачів. Цей функціонал може бути використаний в інших проектах, де передбачається реєстрація користувачів.