

Инструкция № 1 по созданию Веб-приложения на **Django**

Оглавление

1 . Создаем проект(project) прописывая инструкцию в терминале	1
2. Создаем приложение(app) прописывая инструкцию а также регистрируем его в settings.py	1
3. Создаем HTML шаблон (start.html) в app/templates/app.....	2
4. Пишем функцию, которая будет запускать HTML шаблон в views.py	2
5. Создаем ссылку которая будет запускать функцию в urls.py , для этого создаем файл urls.py в app	2
6. Запускаем веб приложение.....	3
#Результат.....	3

Для создания веб приложения изначально требуется:

- **Pycharm Community Edition** www.jetbrains.com
- **Python** www.python.org
- **Django**

1 . Создаем проект(project) прописывая инструкцию в терминале

```
django-admin startproject project
```

```
cd project
```

#Заходим в папку

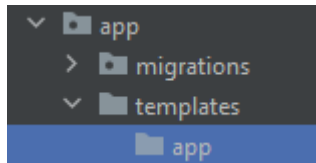
2. Создаем приложение(app) прописывая инструкцию а также регистрируем его в settings.py

```
python manage.py startapp app
```

+ а также регистрируем его в settings.py

```
INSTALLED_APPS = [  
    "django.contrib.admin",  
    "django.contrib.auth",  
    "django.contrib.contenttypes",  
    "django.contrib.sessions",  
    "django.contrib.messages",  
    "django.contrib.staticfiles",  
    "app",  
]
```

3. Создаем HTML шаблон (start.html) в app/templates/app



#Создаем папки строго согласно документации

+ html – шаблон (start.html)

```
start.html x
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8      <h1>Тема проекта</h1>
9      <p>Описание проекта</p>
10 </body>
11 </html>
```

4. Пишем функцию, которая будет запускать HTML шаблон в views.py

```
views.py x
1  from django.shortcuts import render
2
3
4  def start(request):
5      return render(request, 'app/start.html')
```

5. Создаем ссылку которая будет запускать функцию в urls.py , для этого создаем файл urls.py в app

! на данном шаге требуется отредактировать 2 файла urls.py (в project и в app)

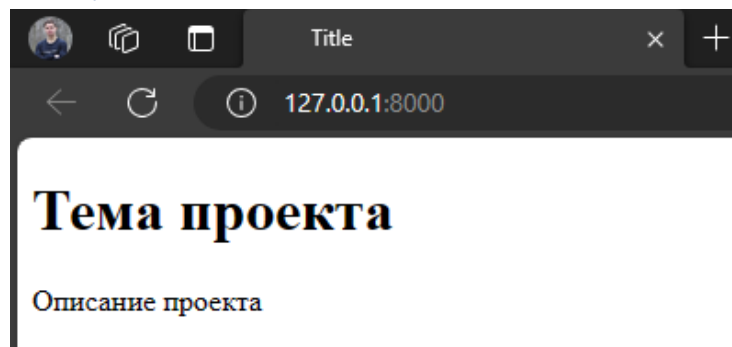
```
app\urls.py x
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path("", views.start),
6  ]
```

```
project\urls.py x
1  from django.contrib import admin
2  from django.urls import path, include
3
4  urlpatterns = [
5      path("admin/", admin.site.urls),
6      path("", include('app.urls')),
7  ]
```

6. Запускаем веб приложение

```
python manage.py runserver
```

#Результат



Структура проекта

- `manage.py` – утилита командной строки, используемая для управления проектом. Это минимальная обертка над файлом `django-admin.py`. Мы не будем редактировать этот файл;
- `__init.py` – пустой файл, который говорит Python о том, что `mysite` является Python-пакетом;
- `settings.py` – содержит конфигурацию нашего проекта, в нем уже заданы базовые настройки;
- `urls.py` – здесь будут храниться *шаблоны адресов* (Uniform Resource Locator – URL). Каждый URL, определенный в этом файле, будет связан с конкретным обработчиком;
- `wsgi.py` – конфигурация для запуска проекта как WSGI-приложения.

Структура приложения

- `admin.py` – здесь мы регистрируем модели для добавления их в систему администрирования Django (использование сайта администрирования Django не является обязательным);
- `apps.py` – файл, содержащий основную конфигурацию приложения
- `migrations` – папка, содержащая миграции базы данных приложения. Миграции позволяют Django отслеживать изменения моделей и синхронизировать их со схемой данных базы;
- `models.py` – модели данных приложения. В любом Django-приложении должен быть этот файл, но он может оставаться пустым;
- `tests.py` – этот файл предназначен для создания тестов для приложения;
- `views.py` – вся логика приложения описывается здесь. Каждый обработчик получает HTTP-запрос, обрабатывает его и возвращает ответ.

Ссылки

Антонио Меле Django 2 в примерах