

Создаём своё первое приложение с Django, часть 6

Продолжаем начатое в пятой части. Мы создали протестированное веб приложение для опросов и теперь нам надо добавить стили и картинку.

В отличие от HTML, генерируемого сервером, веб приложения обычно нуждаются в обработке дополнительных файлов, таких как изображения, JavaScript или CSS, которые нужны для отображения полной веб страницы. В Django мы называем такие файлы “статикой”.

Для маленьких проектов это не является проблемой, так как вы можете просто разместить статичные файлы где-то на своём веб сервере. Однако, для больших проектов, особенно состоящих из множества приложений, взаимодействие с несколькими наборами статичных файлов, предоставляемых каждым приложением, начинает походить на работу фокусника.

Именно для решения этой задачи существует `django.contrib.staticfiles`. Он собирает статичные файлы из каждого вашего приложения (и из других мест, которые вы укажете) в единое место, которое может легко применяться на боевом сервере.

Настройка вида вашего приложения

Сначала создайте каталог `static` в каталоге `polls`. Django будет искать статичные файлы в нём, аналогично тому как Django ищет шаблоны внутри `polls/templates/`.

Параметр конфигурации `STATICFILES_FINDERS` содержит список модулей, которые знают, как получать статичные файлы из различных источников. Одним из стандартных является `AppDirectoriesFinder`, который ищет каталоги “static” в каждом зарегистрированном приложении, например, созданный нами каталог в `polls`. Интерфейс администратора использует аналогичную структуру каталогов для своих статичных файлов.

Внутри созданного каталога `static` создайте ещё один каталог с именем `polls` и внутри него создайте файл `style.css`. Другими словами, ваши стили должны быть в файле `polls/static/polls/style.css`. Из-за особенностей работы `AppDirectoriesFinder` вы можете обращаться из шаблона к этому статичному файлу как `polls/style.css`, аналогично работе с шаблонами.

Пространство имен для статических файлов

Аналогично шаблонам, мы *можем* просто размещать наши статичные файлы прямо в каталоге `polls/static` (не создавая ещё один каталог `polls` внутри), но это будет плохой идеей. Django выбирает первый найденный статичный файл с указанным именем и если *другое* приложение имеет статичный файл с таким же именем, Django не сможет понять какой именно был вам нужен. Нам надо явно указать Django нужный, а вложенный каталог даёт нам *именованное пространство* для этого. Следовательно, просто размещайте статичные файлы приложения внутри *другого* каталога с именем, как у приложения.

Разместите следующий код в файле стилей (`polls/static/polls/style.css`):

```
polls/static/polls/style.css
li a {
    color: green;
}
```

Затем добавьте следующие строки в начало `polls/templates/polls/index.html`:

```
polls/templates/polls/index.html
{% load staticfiles %}

<link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}" />
```

Тэг `{% load staticfiles %}` загружает шаблонный тег `{% static %}` из шаблонной библиотеки `staticfiles`. Шаблонные тег `{% static %}` создаёт абсолютный URL на статичный файл.

Это всё, что вам требуется сделать. Перегрузите страницу <http://localhost:8000/polls/> и вы должны увидеть, что ссылки опроса стали зелёными (стиль Django!). Это означает, что ваш файл стилей загрузился нормально.

Добавление фонового изображения

Теперь надо создать каталог для изображений. Создайте каталог `images` в каталоге `polls/static/polls/`. Внутри созданного каталога разместите изображение с именем `background.gif`. Другими словами, поместите ваше изображение в `polls/static/polls/images/background.gif`.

Затем добавьте следующие строки в файл стилей (`polls/static/polls/style.css`):

```
polls/static/polls/style.css
body {
    background: white url("images/background.gif") no-repeat right bottom;
}
```

Перегрузите страницу <http://localhost:8000/polls/> и вы должны увидеть, что фоновое изображение появилось в нижнем правом углу экрана.

Предупреждение

Естественно, что шаблонный тег `{% static %}` не доступен для использования в статичных файлах, таких как ваш файл стилей, так как эти файлы не создаются с помощью Django. Вы должны всегда использовать **относительные пути** для связывания ваших статичных файлов друг с другом, так как вы можете изменить параметр конфигурации `STATIC_URL` (используемый шаблонным тегом `static` для генерации своих URL) без необходимости менять кучу путей в ваших статичных файлах.

Это были **основы**.

Ознакомившись с работой со статическими файлами, вы можете приступить к следующей главе: «Создаем свое первое приложение с Django, часть 7», чтобы узнать, как настроить интерфейс администратора Django.