

## Разработка программного обеспечения на языке Python

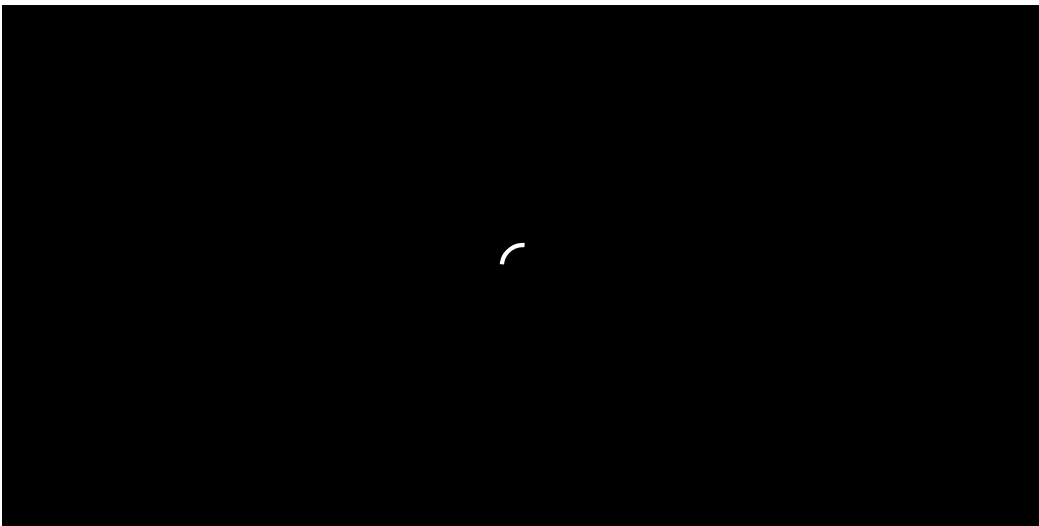
[Обзорная панель](#) ▶ [Мои курсы](#) ▶ [Разработка ПО на языке Python](#) ▶ [Веб-программирование на Python](#) ▶

[Лекция 2. Библиотека Django](#)

### Лекция 2. Библиотека Django

Посмотрите видеоуроки и ответьте на контрольные вопросы после лекции

### Маршрутизация проекта Django



Рассмотрим инструмент для добавления навигации в приложение django.

Центральным моментом любого веб-приложения является обработка запроса, который отправляет пользователь. В Django за обработку запроса отвечают представления или **views**.

По сути представления – это функции обработки, которые принимают данные запроса в виде объекта **request** и генерируют некоторый результат, который затем отправляется пользователю. Ответ может быть сгенерирован в различном виде, например, HTML-страница, перенаправление на другую страницу, код ошибки, XML-документ, или изображение.

По умолчанию представления размещаются в приложении в файле **views.py**. При создании нового проекта файл *views.py* пока никак не обрабатывает запросы, он только импортирует функцию **render()**, которая может использоваться для обработки.

Так, изменим файл *views.py* следующим образом:

```
views.py
1 from django.shortcuts import render
2 from django.http import HttpResponse
3
4
5 def index(request):
6     return HttpResponse("<h2>Главная</h2>")
7
8
9 def about(request):
10    return HttpResponse("<h2>О сайте</h2>")
11
12
13 def contact(request):
14    return HttpResponse("<h2>Контакты</h2>")
```

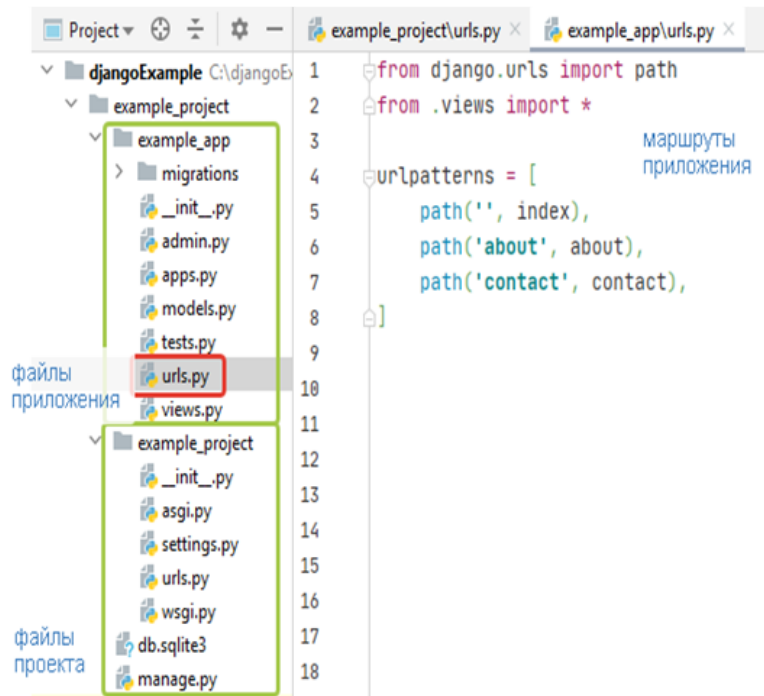
создадим три функции, которые будут обрабатывать запросы. Каждая функция принимает в качестве параметра объект `request`. Генерировать результат можно различными способами. Один из них представляет использование класса `HttpResponse`, который позволяет отправить текстовое содержимое. Для генерации ответа в качестве параметра `HttpResponse` передается некоторая строка. Это может быть в том числе и код html в виде строки.

Чтобы эти функции сопоставлялись с запросами, надо определить для них маршруты в проекте. Маршруты проекта находятся в файле `urls.py`. По умолчанию в проект добавлены маршруты, относящиеся к приложению администратора, подключаемому по умолчанию.

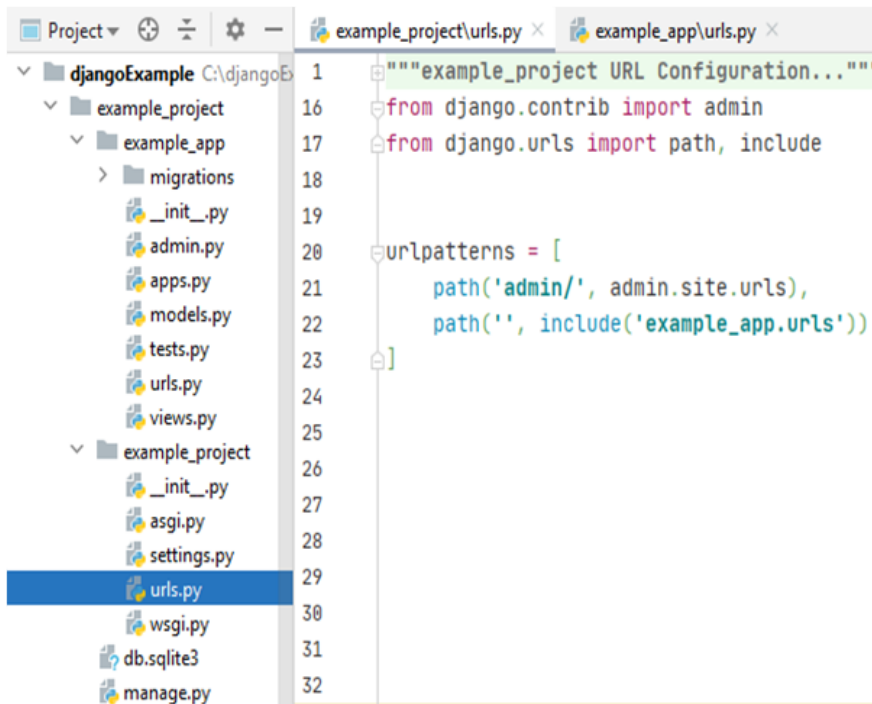
```
urls.py
1 """example_project URL Configuration..."""
16 from django.contrib import admin
17 from django.urls import path
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21 ]
```

Маршруты нашего сайта можно добавить в этот список (как было сделано в примере в предыдущем видео). Однако часто проект может состоять из нескольких приложений, и чтобы было удобно ориентироваться в том, к какому приложению относится определенный маршрут, можно в каждом приложении создавать свой файл со списком маршрутов и подключать их к маршрутам проекта.

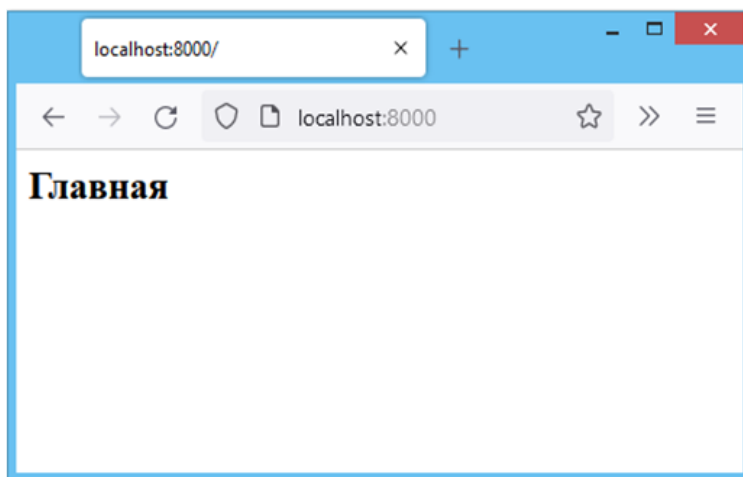
Таким образом, создадим в папке приложения файл `urls.py` и добавим в него переменную `urlpatterns`. Она определяет набор сопоставлений функций обработки с определенными строками запроса. Функция `path()` для задания маршрута принимает два параметра: запрошенный адрес URL и функцию, которая обрабатывает запрос по этому адресу. Например, запрос к корню веб-сайта будет обрабатываться функцией `index`, запрос по адресу "about" будет обрабатываться функцией `about`, а запрос "contact" - функцией `contact`.



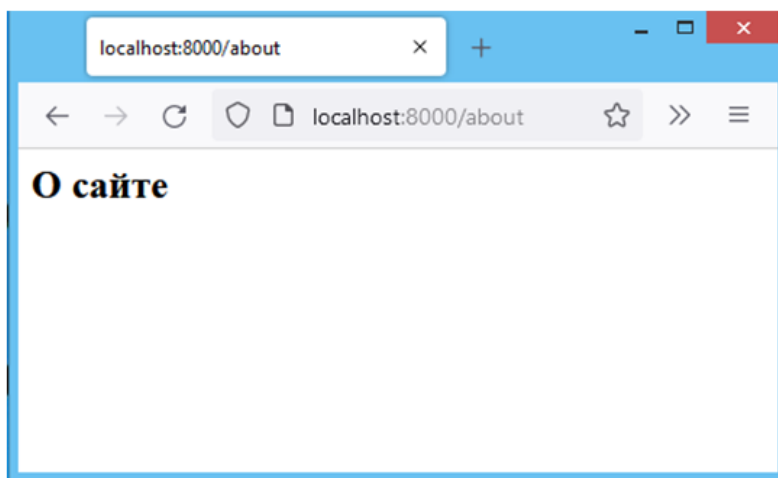
Далее подключим маршруты приложения к проекту. Для этого импортируем модуль `include` и с помощью него добавим в список маршруты нашего приложения.



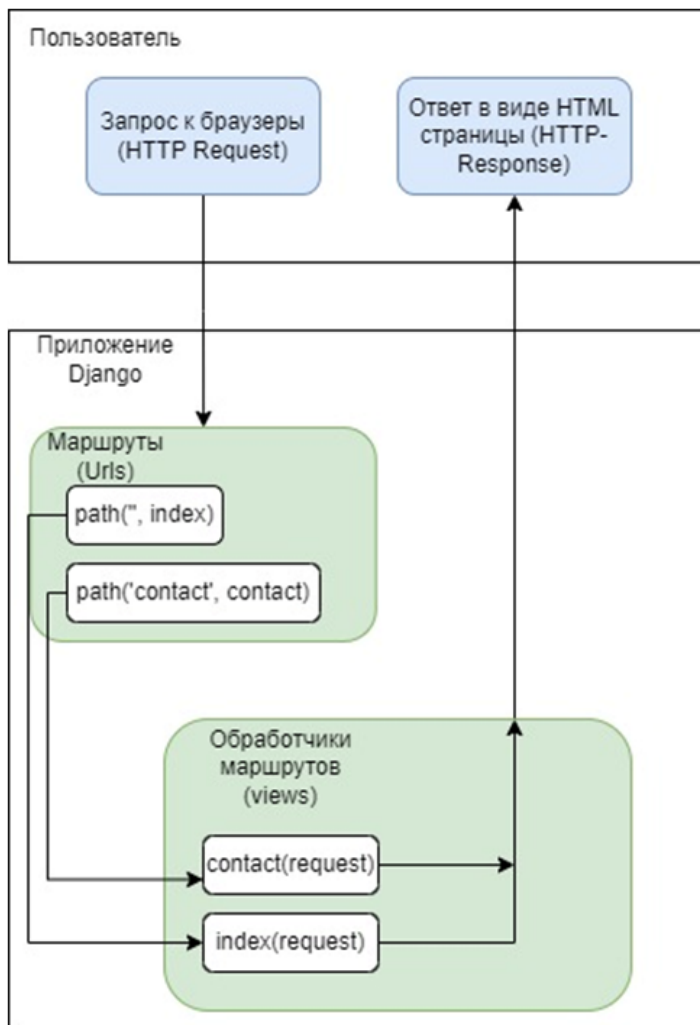
Запустим проект и обратимся по некоторым из этих адресов. Перейдем в браузере по адресу `http://127.0.0.1:8000/`. Браузер нам отобразит главную страницу сайта.



Если в адресной строке браузера изменить адрес на: `http://127.0.0.1:8000/about`, то мы попадем на страницу 0 сайта.



Схематично маршрутизация запросов пользователя и формирование ответов от веб-приложения в рассмотренном примере представлена далее.



Подведем итоги, были рассмотрены некоторые ключевые элементы навигации в проекте Django: представления и маршрутизация запросов пользователей.

Параметры представлений в Django

ПРЕДЫДУЩИЙ ЭЛЕМЕНТ КУРСА

◀ [Задание 2. Создание flask приложения](#)

Перейти на...

СЛЕДУЮЩИЙ ЭЛЕМЕНТ КУРСА

[Задание 3. Создание своего первого сайта на Django](#) ►

[Политика допустимого использования](#)

**Контакты** +7(391) 206-27-05  
[info-ms@sfu-kras.ru](mailto:info-ms@sfu-kras.ru)

[Скачать мобильное приложение](#)

[Инструкции по работе в системе](#)