

Шпаргалка Знакомство с Django

Как устроены проекты Django

Проект разделяется на несколько независимых кусочков-**приложений**. Нужно стремиться писать приложения так, что бы их можно было применять и в других проектах.

Структура проекта

Обычно проект устроен так:

```
├── anfisa4freinds # Имя папки == имя проекта. Здесь хранятся основные настройки
│   ├── settings.py # Глобальные настройки проекта
│   └── urls.py      # Список всех адресов, доступных в проекте
├── homepage       # Имя папки == имя приложения (hompage)
│   ├── views.py   # Место хранения view-функций приложения
│   └── urls.py     # Адреса, обрабатываемые приложением
├── iccream        # Имя папки == имя приложения (iccream)
│   ├── views.py
│   └── urls.py
└── manage.py      # Файл для взаимодействия с консолью
```

settings.py

Важные константы:

`INSTALLED_APPS = [...]` — список приложений, из которых состоит проект.

`ROOT_URLCONF = 'anfisa4friends.urls'` — файл, с которого начинается обработка URL, проекта.

`LANGUAGE_CODE = 'ru'` — настройка локализации проекта.

urlpatterns

URL, доступные в Django-проекте, перечисляют в файлах `urls.py` в списках `urlpatterns`. В них указывают относительные адреса, без указания протокола и домена.

`path()` связывает адрес и view-функцию, которая обрабатывает запрос к этому адресу. В приведённом примере в ответ на запрос к адресу `{URL}/aboutme/` вернётся результат работы функции `about()` из файла `views.py` в этой же папке.

```
urlpatterns = [
    ...
    path('aboutme/', views.about),
    ...
]
```

include

Если указывать все адреса в главном файле *urls.py* проекта, список адресов будет большим, работать с ним будет неудобно. Лучше разложить адреса в несколько файлов *urls.py*, по директориям приложений.

`include` подключает к файлу *urls.py* правила `path()` из других *urls.py*. Например:

```
urlpatterns = [
    # Правила обработки главной страницы в файле urls.py в директории /homepage
    path('', include('homepage.urls')),

    # Если запрос начинается с 'icecream/' —
    # то правила обработки таких запросов ищи в urls.py в директории /icecream
    path('icecream/', include('icecream.urls')),
]
```

view-функции

View-функции определяют, какой контент увидит пользователь при вызове определённого URL.

```
def about(request):
    return HttpResponse('Вот что я хочу тебе сказать на этой странице...')
```

import

Чтобы из одного файла можно было обратиться к коду другого — нужно **импортировать** файл. Импортировать можно файл целиком или какие-то объекты из него: функции или переменные.

При импорте отдельной функции к ней обращаются просто по имени: `path(...)`.

При импорте файла к его функциям обращаются через имя импортированного файла:

`views.icecream_list`.

Например, вот импорт необходимых функций в файл *urls.py*:

```
from django.urls import path # Импортируется функция path(), теперь её можно вызывать в этом файле.
from . import views # Импортирован файл views.py, теперь все его функции доступны в этом коде.

urlpatterns = [
    path('', views.icecream_list),
]
```