

Django Starter

Введение в Django

Django Starter

Introduction



Лазорык Михаил

Software developer, 3 года опыта

 mykhailo.lazoryk

 mykhailo-lazoryk



Введение в Django

Django Starter

План урока

1. Преимущества Django
2. История Django
3. Что такое MVC и его преимущества. Реализация в Django
4. Подготовка окружения и установка Django
5. Пример создания простого проекта и разбор его структуры
6. manage.py - зачем нужен и почему удобно

Django Starter

Преимущества Django

- Полноценный фреймворк всё-в-одном
- Обширное комьюнити
- Отличная документация
- Широкая востребованность Django при приеме на работу
- Легкость настройки и установки

Django Starter

История Django

Создан в 2003 г. Адрианом Головатым и Саймоном Уиллисоном



Опубликован в 2005 г. под лицензией BSD



Основание в 2008 г. Django Software Foundation, выпуск 1.0



2017 г., выпуск версии 1.11 LTS (до 08.2020). Python 2.7



07.2019 г., выпуск версии 2.2 LTS (до 08.2022). Python 3 only.



Джа́нго Ре́йнхардт

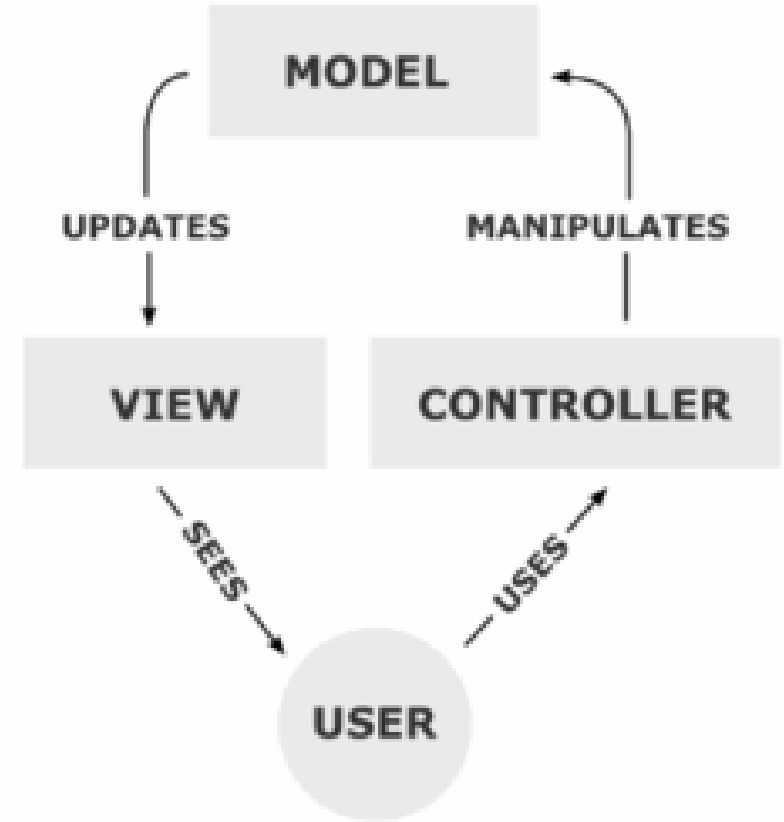
Django Starter

Что такое Model-View-Controller (MVC) и его преимущества

MVC - Model-View-Controller

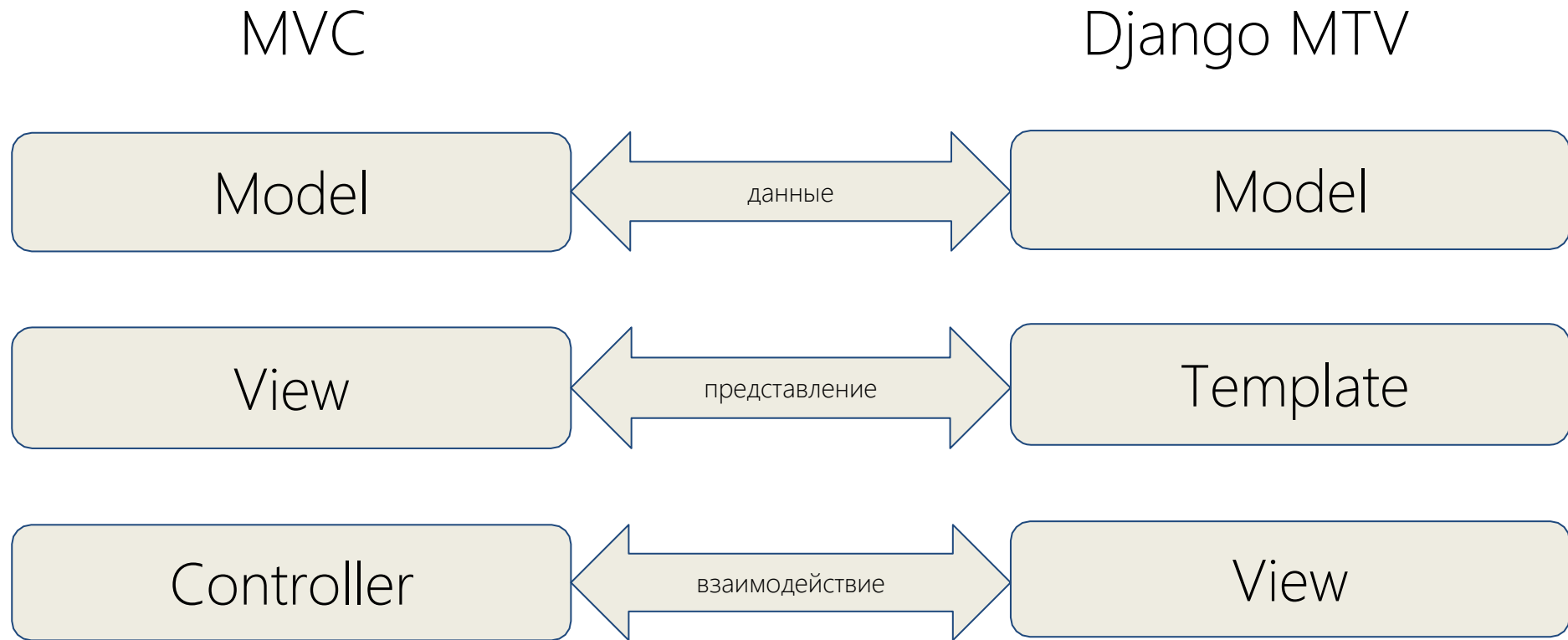
Это паттерн программирования, из трех
компонентов:

- **Модель** (Model)
- **Представление** (View)
- **Контроллер** (Controller)



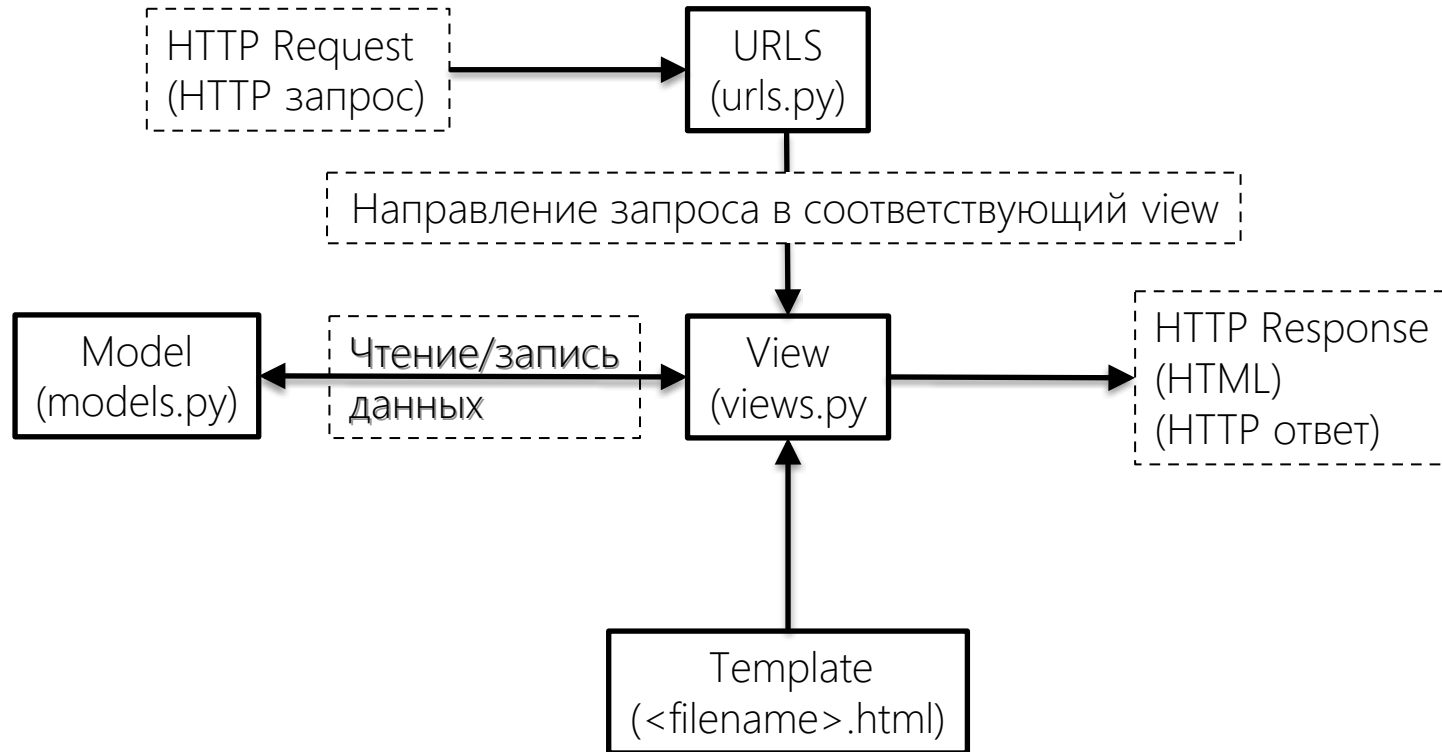
Django Starter

Реализация MVC в Django (MTV)



Django Starter

Структура файлов в реализации MVC в Django (MTV)



Django Starter

Подготовка виртуального окружения и установка Django

1. Установка Python с официального сайта (3.6 и выше)

<https://www.python.org/downloads/>

2. Создание виртуального окружения и его активация

```
pip install virtualenv
```

```
virtualenv ENV
```

(Windows)

```
ENV\Scripts\activate
```

(Linux)

```
./ENV/bin/activate
```

3. Установка Django:

```
pip install Django
```

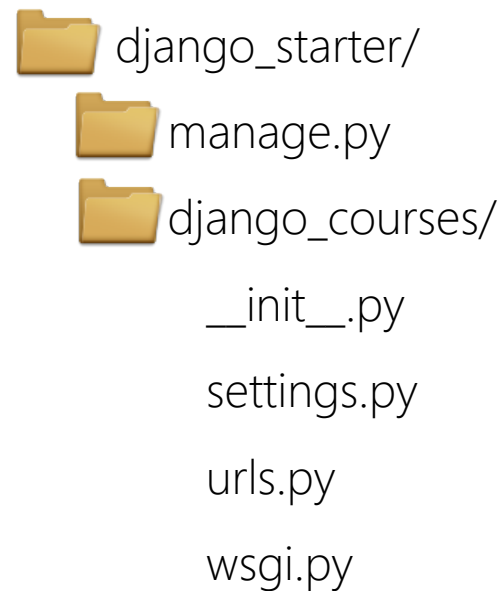
Django Starter

Создание Django проекта и анализ его структуры

Создадим Django проект:

```
django-admin startproject django_courses
```

Полученная структура проекта:



```
graph LR; A[django_starter/] --> B[manage.py]; A --> C[django_courses/]; C --> D[__init__.py]; C --> E[settings.py]; C --> F[urls.py]; C --> G[wsgi.py];
```

The diagram illustrates the file structure of a Django project created by the command. It shows a root directory 'django_starter/' which contains a 'manage.py' file and a subdirectory 'django_courses/'. The 'django_courses/' subdirectory contains four files: '__init__.py', 'settings.py', 'urls.py', and 'wsgi.py'.

Django Starter

Разница между проектами и приложениями в Django

Django project (проект) – это набор настроек и приложений для конкретного веб-сайта.

Django app (приложение) – это веб-приложение, которое делает что-то конкретное. Например приложение веб-блога, база данных публичных записей, приложение для опроса, видео-плеер и т.п. Т.е. что-то, реализующее конкретный функционал.

Один проект может содержать много приложений. Одно приложение может быть в разных проектах.

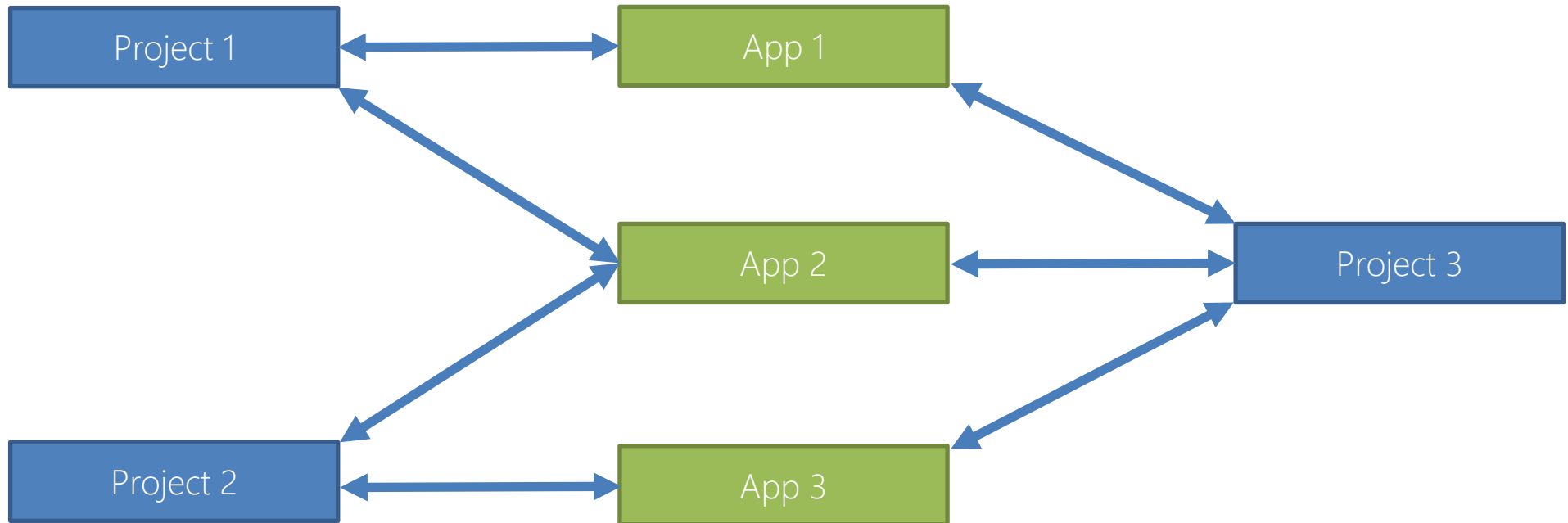
Папка приложения может находиться в любом месте в Python path, но обычно её помещают в папку проекта.

Django Starter

Связь между проектами и приложениями в Django

Один проект может содержать много приложений.

Одно приложение может быть в разных проектах.



Django Starter

manage.py - зачем нужен и почему удобно

1. Работает также как **django-admin**, но при этом также:
 1. Добавляет создаваемые проекты в **sys.path**
 2. Настраивает переменную окружения **DJANGO_SETTINGS_MODULE** для использования **settings.py** файла приложения

Django Starter

Запуск тестового сервера

Запустим наш сервер:

```
python manage.py runserver
```

<http://127.0.0.1:8000/> - ссылка в браузере на главную страницу приложения по умолчанию.

<http://localhost:8000/> - работает аналогично

Параметры запуска:

```
python manage.py runserver [IP]:[PORT]
```

```
python manage.py runserver 0:8000
```

0 – это сокращение для 0.0.0.0

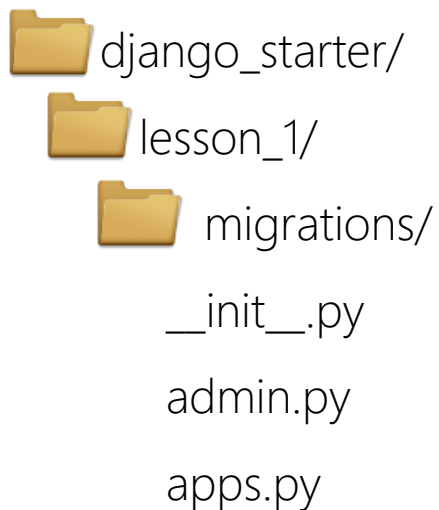
Django Starter

Создание Django приложения и анализ его структуры

Создадим Django приложение:

```
python manage.py startapp lesson_1
```

Полученная структура приложения:



Django Starter

Вывод простейшей HTML страницы

1. Редактирование `lesson_1/views.py`
2. Создание `lesson_1/urls.py`
3. Редактирование `django_courses/urls.py`
4. Открыть ссылку http://localhost:8000/lesson_1/

Django Starter

Редактирование lesson_1/views.py

```
from django.http import HttpResponse
```

```
def index(request):
```

```
    return HttpResponse("Hello, world! Вы на странице 1 урока.")
```

Django Starter

Создание lesson_1/urls.py

```
from django.urls import path
```

```
from . import views
```

```
urlpatterns = [  
    path("", views.index, name='index'),  
]
```

Django Starter

Редактирование django_courses/urls.py

```
from django.contrib import admin

from django.urls import include, path

urlpatterns = [
    path('lesson_1/', include('lesson_1.urls')),
    path('admin/', admin.site.urls),
]
```

Django Starter

Запуск тестового сайта

```
python manage.py runserver
```

http://localhost:8000/lesson_1/

Поздравляем!



Спасибо за урок!

Django Starter

Закрепление материала

1. Преимущества Django
2. История Django
3. Что такое MVC и его преимущества. Реализация в Django
4. Подготовка окружения и установка Django
5. Пример создания простого проекта и разбор его структуры
6. manage.py - зачем нужен и почему удобно

Проверка знаний

TestProvider.com



Проверьте как Вы усвоили данный материал на TestProvider.com

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

Django Starter

Спасибо за внимание! До новых встреч!



Лазорык Михаил
Software developer



Информационный видеосервис для разработчиков программного обеспечения

