

Django Starter

Вступ до Django

Django Starter

Introduction



Лазорик Михайло

Software developer, 3 роки
досвіду



mykhailo.lazoryk



mykhailo-lazoryk



Вступ до Django

Django Starter

План уроку

1. Переваги Django.
2. Історія Django.
3. Що таке MVC та його переваги. Реалізація у Django.
4. Підготовка оточення та встановлення Django.
5. Приклад створення простого проекту та розбір його структури.
6. manage.py: навіщо потрібний і чому зручно.

Django Starter

Переваги Django

- Повноцінний фреймворк «все в одному».
- Велике ком'юніті.
- Чудова документація.
- Широкий попит Django під час наймання на роботу.
- Легкість налаштування та встановлення

Django Starter

Історія Django

Створений у 2003 році Адріаном Головатим і Саймоном Віллісоном.



Опублікований у 2005 році під ліцензією BSD.



Заснування у 2008 році Django Software Foundation, випуск 1.0.



2017 рік, випуск версії 1.11 LTS (до 08.2020). Python 2.7/



07.2019 року, випуск версії 2.2 LTS (до 08.2022). Python 3 only/



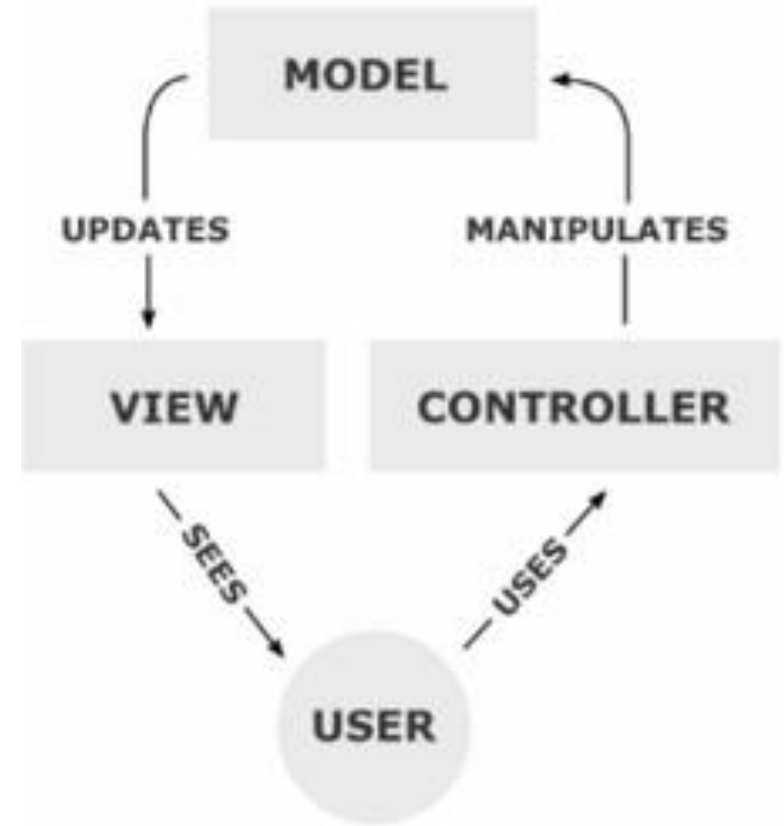
Дж́анго Р́ейнхардт

Django Starter

Що таке Model-View-Controller (MVC) та його переваги

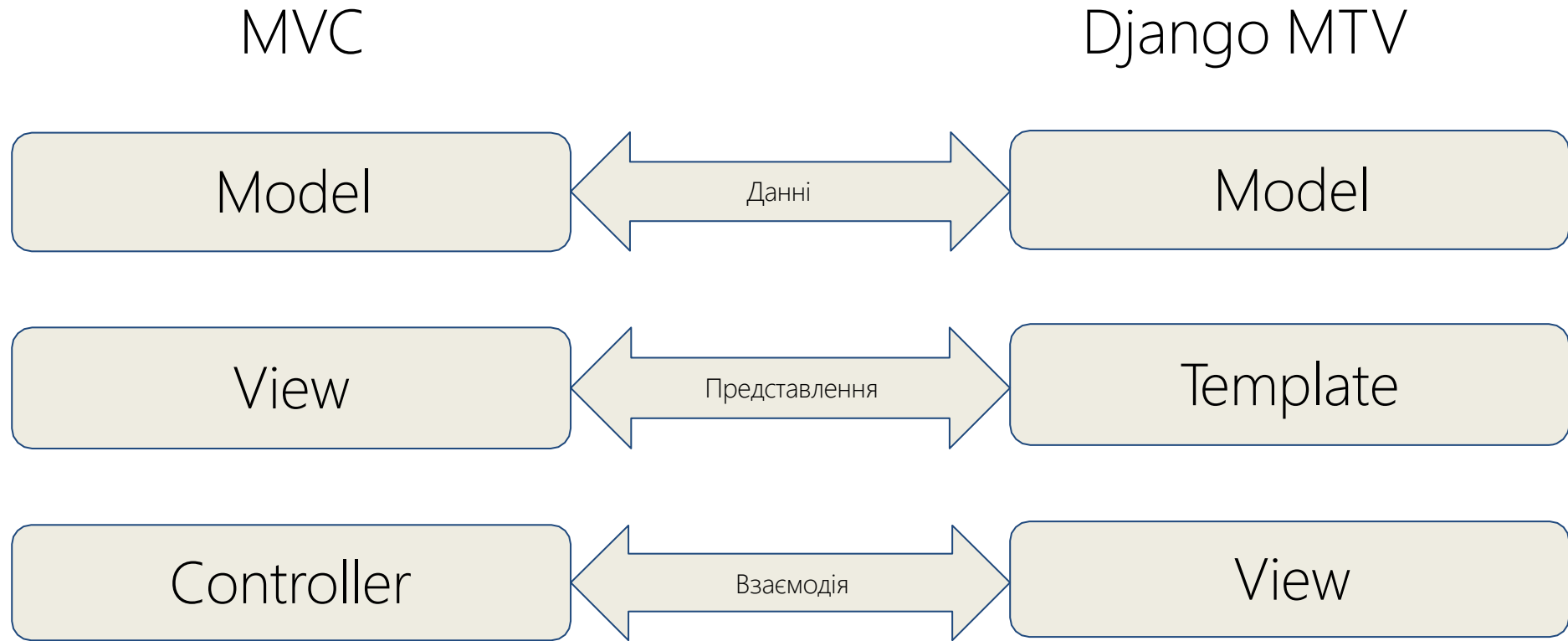
MVC (*Model-View-Controller*) – це паттерн програмування, який складається з трьох компонентів:

- Модель (Model).
- Представлення (View).
- Контролер (Controller).



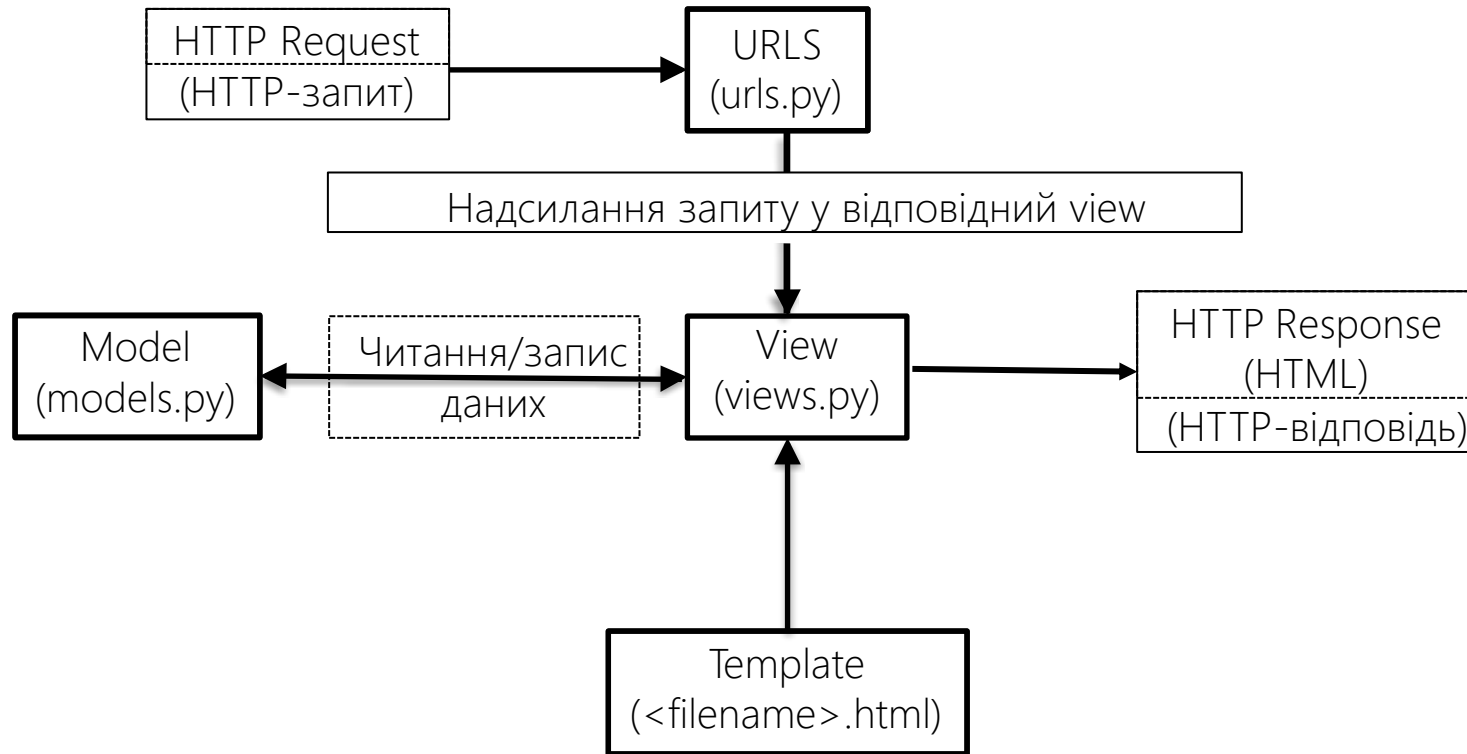
Django Starter

Реалізація MVC у Django (MTV)



Django Starter

Структура файлів і реалізація MVC у Django (MTV)



Django Starter

Підготовка віртуального оточення та встановлення Django

1. Встановлення Python з офіційного сайту (3.6 та вище)

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

2. Створення віртуального оточення та його активація:

```
pip install virtualenv virtualenv ENV
```

(Windows)	(Linux)
ENV\Scripts\activate	./ENV/bin/activate

3. Встановлення Django:

```
pip install Django
```

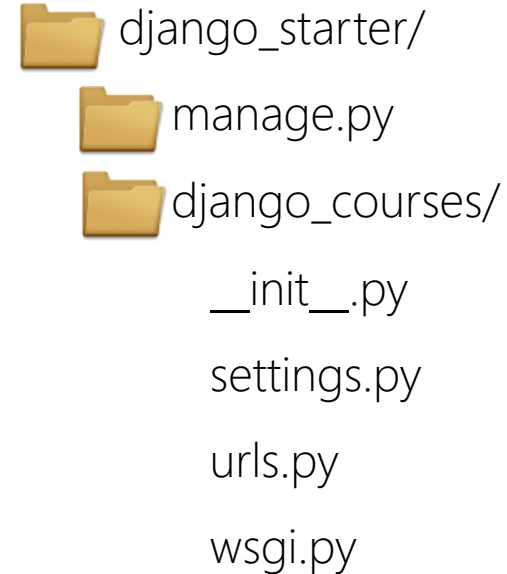
Django Starter

Створення проекту Django та аналіз його структури

Створимо Django-проект:

```
django-admin startproject django_courses
```

Отримана структура проекту :



```
django_starter/  
├── manage.py  
└── django_courses/  
    ├── __init__.py  
    ├── settings.py  
    ├── urls.py  
    └── wsgi.py
```

Django Starter

Різниця між проектами та застосунками в Django

Django project (проект) – це набір налаштувань і застосунків для конкретного веб-сайту.

Django app (застосунок) – це веб-застосунок, який робить щось конкретне. Наприклад, застосунок веб-блогу, база даних публічних записів, застосунок для опитування, відеоплеєр тощо. Тобто щось, що реалізує конкретний функціонал.

Один проект може містити багато застосунків. Один застосунок може бути у різних проектах.

Тека застосунку може бути розташована у будь-якому місці в Python path, але зазвичай її поміщають у теку проекту.

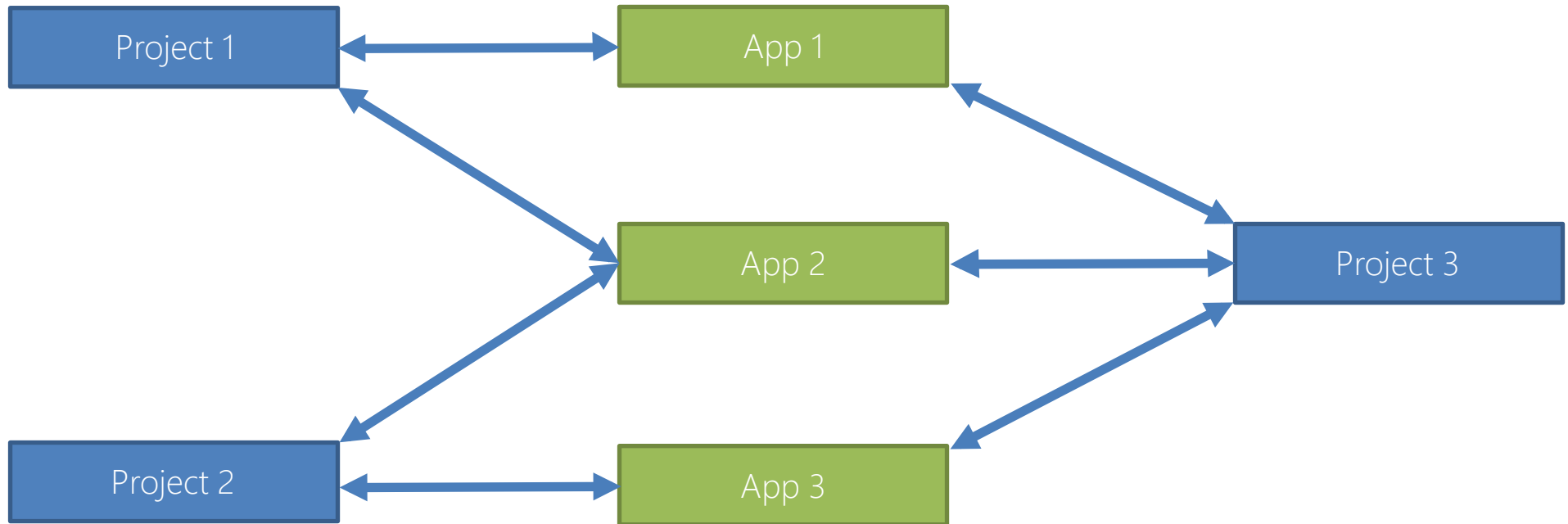


Django Starter

Зв'язок між проектами та застосунками в Django

Один проект може містити багато застосунків.

Один застосунок може бути у різних проектах.



Django Starter

`manage.py`: навіщо потрібен і чому зручно

Працює так само як `django-admin`, але також:

- 1) додає проекти, що створюються, у `sys.path`;
- 2) налаштовує змінну оточення `DJANGO_SETTINGS_MODULE` для використання `settings.py` файлу застосунку.

Django Starter

Запуск тестового сервера

Запустимо наш сервер:

```
python manage.py runserver
```

<http://127.0.0.1:8000/> – посилання в браузері на головну сторінку застосунку за замовчуванням.

. <http://localhost:8000/> – працює аналогічно.

Приклади запуску:

```
python manage.py runserver [IP]:[PORT]  
python manage.py runserver 0:8000
```

0 – це скорочення для 0.0.0.0

Django Starter

Створення Django-застосунку й аналіз його структури

Створимо Django-застосунок: `python manage.py startapp lesson_1`

Отримана структура застосунку:



models.py

tests.py

views.py

Django Starter

Виведення найпростішої HTML-сторінки

1. Редагування lesson_1/views.py.
2. Створення lesson_1/urls.py.
3. Редагування django_courses/urls.py.
4. Відкрити посилання http://localhost:8000/lesson_1/.

Django Starter

Редагування lesson_1/views.py

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

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

```
    return HttpResponse("Hello, world! Ви на сторінці 1 уроку.")
```

Django Starter

Створення lesson_1/urls.py

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

```
from . import views
```

```
urlpatterns = [
```

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

```
]
```

Django Starter

Редагування 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 сайту

```
python manage.py runserver
```

http://localhost:8000/lesson_1/

Вітаємо!



Дякую за урок!

Django Starter

Закріплення матеріалу

1. Переваги Django.
2. Історія Django.
3. Що таке MVC та його переваги. Реалізація у Django.
4. Підготовка оточення та встановлення Django.
5. Приклад створення простого проекту та розбір його структури.
6. manage.py: навіщо потрібен і чому зручно.

Перевірка знань

TestProvider.com



Перевірте, як Ви засвоїли цей матеріал на TestProvider.com

TestProvider – це online-сервіс перевірки знань з інформаційних технологій. Завдяки йому Ви можете оцінити Ваш рівень і виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT-фахівця.

Успішне проходження фінального тестування дасть Вам змогу отримати відповідний Сертифікат.

Django Starter

Дякую за увагу! До нових зустрічей!



Лазорик Михайло
Software developer



Інформаційний відеосервіс для розробників програмного забезпечення

