

BACKEND - РАЗРАБОТКА НА ФРЕЙМВОРКЕ DJANGO

РЭУ ИМ. Г.В. ПЛЕХАНОВА МПТ



<https://ucmpt.ru/>

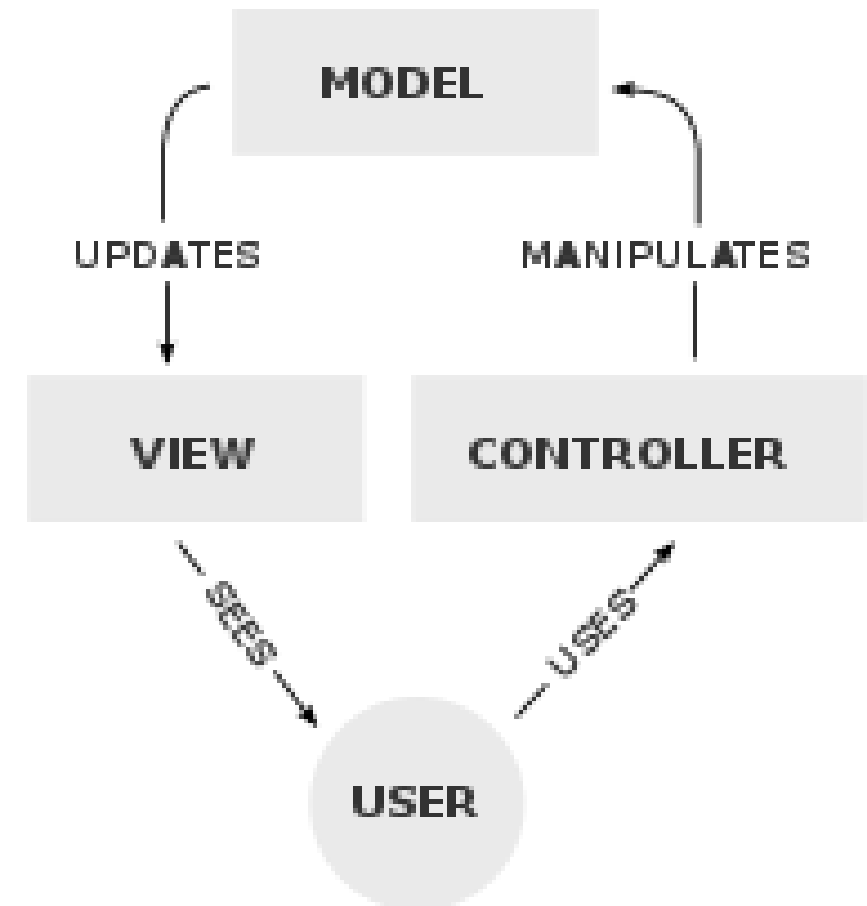
<https://do.rea.ru/>

DJANGO

- **Django - это высокоуровневый веб-фреймворк на Python, который поощряет быструю разработку и чистый, прагматичный дизайн. Созданный опытными разработчиками, он берет на себя большую часть хлопот по веб-разработке, так что вы можете сосредоточиться на написании своего приложения без необходимости изобретать велосипед. Это бесплатно и с открытым исходным кодом.**

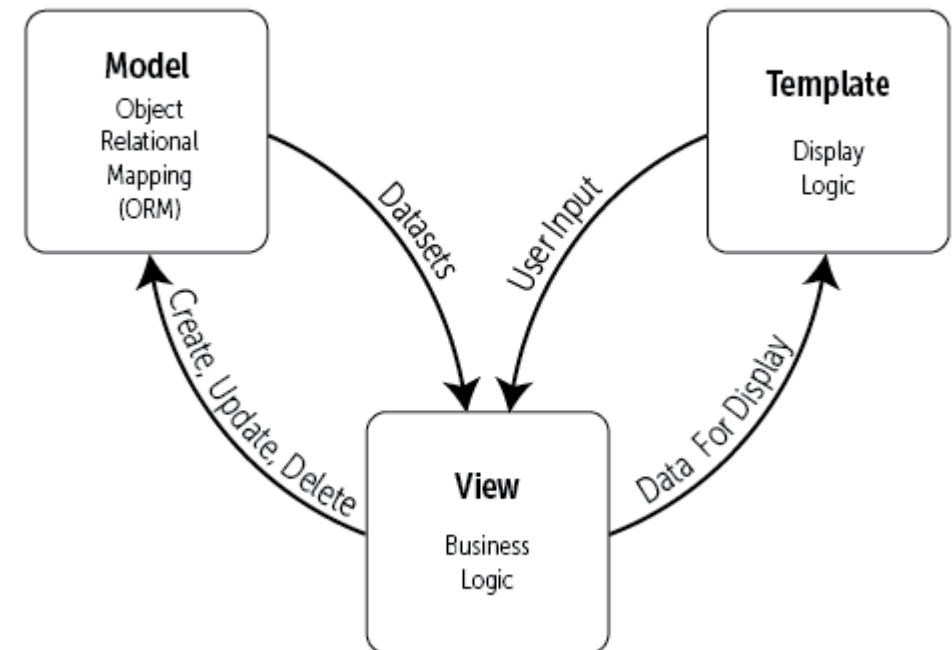
MVC - ПАТТЕРН

- **MVC - Model-View-Controller** – схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер – таким образом, что модификация каждого компонента может осуществляться независимо.
- Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.
- Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.
- Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.



MTV - ПАТТЕРН

- Фреймворк Django реализует архитектурный паттерн Model-Template-View или сокращенно MTV, который является модификацией паттерна MVC (Model-View-Controller).
- **Model:** описывает данные, используемые в приложении. Отдельные классы, как правило, соответствуют таблицам в базе данных.
- **View (Controller):** получает запрос, обрабатывает его и отправляет в ответ пользователю некоторый ответ. Если для обработки запроса необходимо обращение к модели и базе данных, то View взаимодействует с ними. Для создания ответа может применять Template или шаблоны.
- **Template (View):** представляет логику представления в виде сгенерированной разметки html.

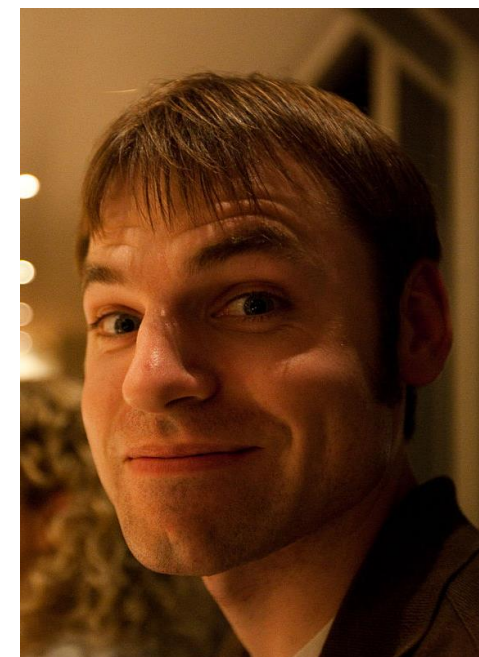


ИСТОРИЯ РАЗВИТИЯ DJANGO

- Django был создан осенью 2003 года, когда веб-программисты из газеты Lawrence Journal-World Адриан Головатый и Саймон Уиллисон начали использовать Python для создания приложений. На ранней стадии разработки был нанят Джейкоб Каплан-Мосс незадолго до окончания стажировки Саймона Уиллисона. Django был выпущен публично под лицензией BSD в июле 2005 года. Фреймворк был назван в честь гитариста Джанго Рейнхардта. Адриан Головатый - цыганский джазовый гитарист и большой поклонник Джанго Рейнхардта.
- В июне 2008 года было объявлено, что созданная организация Django Software Foundation (DSF) будет поддерживать Django в будущем.



Саймон Уиллисон



Адриан Головатый

ИСТОРИЯ РАЗВИТИЯ DJANGO

2.0	2 дек 2017	Первая версия с поддержкой только Python 3, упрощенный синтаксис маршрутизации URL, «админка» с дружественным мобильным интерфейсом.
2.1	1 авг 2018	Разрешение view для моделей администратора.
2.2 LTS	апр 2019	LTS с поддержкой до апреля 2022 года
3.0	2 дек 2019	Поддержка MariaDB, ASGI
3.1	4 авг 2020	Асинхронные view и middleware, JSONField для всех поддерживаемых БД
3.2 LTS	апр 2021	LTS с поддержкой до апреля 2024 года
4.0	дек 2021	Расширенная поддержка как минимум до апреля 2023 года
4.1	3 авг 2022	Асинхронный интерфейс к ORM

ПРЕИМУЩЕСТВА DJANGO

- **Высокая скорость разработки**
- **Безопасность**
- **Масштабируемость**
- **Универсальность**
- **Переносимость**
- **Открытость**

ФИЛОСОФИЯ DJANGO

- **Свободное соединение**
- **Меньше кода**
- **Быстрое развитие**
- **Не повторяйтесь (DRY)**
- **Явное лучше неявного**
- **Последовательность**

ГДЕ ИСПОЛЬЗУЕТСЯ DJANGO?



**NATIONAL
GEOGRAPHIC**



Bitbucket

УСТАНОВКА ПРИЛОЖЕНИЕ DJANGO ИЗ PIP

- Для начала необходимо убедиться в наличии и работоспособности pip-менеджера. Для этого в командной строке пропишем

pip - V

- Если pip установлен, то выведется версия pip и путь где он находится.
- Если pip не установлен, то выведется ошибка.
- Для того, чтобы скачать и установить pip, следует прописать следующую команду

python -m ensurepip --upgrade

- Также можно обновить pip

pip install --upgrade pip

УСТАНОВКА ПРИЛОЖЕНИЕ DJANGO ИЗ PIP

- Создадим папку в любом удобном месте, где будет лежать наш проект

`md django-pip`

- Далее надо создать виртуальную среду

`python -m venv .venv`

- После инициализации, для того чтобы зайти в виртуальную среду нужно вызвать bat-файл

`.venv\Scripts\activate.bat`

- Находясь в виртуальной среде скачиваем модуль django

`python -m pip install Django`

- Теперь можно создать проект в котором будем работать и разрабатывать веб-приложение

`django-admin startproject Magazine`

УСТАНОВКА ПРИЛОЖЕНИЕ DJANGO ИЗ PIP

- После переходим в папку проекта

`cd Magazine`

- И запустим наше приложение

`python manage.py runserver`

- После этого можно перейти в браузер и по URL-адресу зайти на наше приложение

(По умолчанию) `localhost:8000/`

- Для остановки работы приложения следует нажать комбинацию клавиш **CTRL+BREAK**

- Для выхода из виртуального окружения достаточно прописать `deactivate`

СТРУКТУРА ПРОЕКТА DJANGO

- **manage.py:** выполняет различные команды проекта, например, создает и запускает приложение
- **django-pip** - собственно папка проекта Django, которая содержит следующие файлы:
 - **__init__.py:** данный файл указывает, что папка, в которой он находится, будет рассматриваться как модуль. Это стандартный файл для программы на языке Python;
 - **settings.py:** содержит настройки конфигурации проекта;
 - **urls.py:** содержит шаблоны URL-адресов, по сути определяет систему маршрутизации проекта;
 - **wsgi.py:** содержит свойства конфигурации WSGI;
 - **asgi.py:** содержит свойства конфигурации ASGI.

WSGI И ASGI

- **WSGI (Web Server Gateway Interface)** - Это спецификация (PEP 3333), которая описывает, как веб-сервер взаимодействует с веб-приложениями и как веб-приложения могут быть объединены в цепочку для обработки одного запроса.
- Он используется при развертывании проекта.
- **ASGI (Asynchronous Server Gateway Interface)** - Это спецификация, предназначенная для обеспечения стандартного интерфейса между веб-серверами, фреймворками и приложениями Python с поддержкой `async`.
- Там, где WSGI предоставил стандарт для синхронных приложений Python, ASGI предоставляет стандарт как для асинхронных, так и для синхронных приложений, с реализацией обратной совместимости WSGI и несколькими серверами и фреймворками приложений.

ПРИЛОЖЕНИЯ В DJANGO

- Приложение в Django – это отдельные папки с файлами для реализации веб-приложений. Django состоит из приложений, т.е. частей сайта. Философия Django заключается в том, чтобы приложения были как можно менее связанные друг с другом. В отличном случае приложения не связаны вовсе.

СТРУКТУРА ПРИЛОЖЕНИЙ

- **migrations** – Папка с миграциями;
- **Templates** – Папка с представлениями; (Создается самостоятельно)
- **__init__.py** - данный файл указывает, что папка, в которой он находится, будет рассматриваться как модуль. Это стандартный файл для программы на языке Python;
- **admin.py** – Файл для настройки панели администратора;
- **apps.py** - Этот файл создан для того, чтобы помочь пользователю включить любую конфигурацию для приложения. Можно настроить некоторые атрибуты приложения;
- **models.py** – Содержит информацию и архитектуру таблиц базы данных;
- **tests.py** – Файл для написания тестов своего приложения;
- **urls.py** – содержит шаблоны URL-адресов, по сути определяет систему маршрутизации проекта; (Создается самостоятельно);
- **views.py** – Контроллер, в нем хранятся методы для обработки запросов.

МИГРАЦИИ

- Миграция – это файлы с последовательным применением изменений к базе данных, с целью синхронизации между приложением и базой данных.
- В реальных проектах модели данных изменяются по мере реализации функций. При добавлении или изменении новых сущностей или свойств схемы базы данных должны быть соответствующим образом изменены для синхронизации с приложением. Функция миграции позволяет последовательно применять изменения схемы к базе данных, чтобы синхронизировать ее с моделью данных в приложении без потери существующих данных.