

# Python Frameworks. Сущности бэкенда.

Что нужно знать о фреймворка, как их выбирать, а также  
какие сущности мы должны очень хорошо знать.

# Фреймворки

## Список фреймворков, о которых мы поговорим

- Flask
- Django (DRF)
- Fast API

**Flask**

**Удобство и гибкость**



**Flask**

# Немного о Flask

## Что из себя представляет и зачем нужен

- Flask - это микро-веб-фреймворк, написанный на Python. Micro web framework - это платформа веб-разработки с простой настройкой, которую можно использовать при разработке минималистичных веб-приложений.
- Flask поставляется с такими опциями, как шаблонизаторы, такие как ORM, кэширование и аутентификация.

# Плюсы Flask

## Что он может делать хорошо?

- Удобен при старте изучения бэра
- Разработка от начала и до конца проходит достаточно быстро
- Идеален для легковесных сервисов
- Очень гибкий и настраиваемый

# Минусы Flask

**В чем он будет не очень?**

- Не содержит все нужные функции под капотом, а значит гарантии безопасности системы сильно ниже
- Синхронный
- Не содержит много синтаксического сахара и инструментов для более удобного решения задач

# Django

Для чего-то по-  
настоящему большого

The Django logo, featuring the word "django" in a white, lowercase, sans-serif font, centered within a dark green rectangular background.

django

# Немного о Django

## Что он такое

- Django - это бесплатная платформа веб-разработки на Python с открытым исходным кодом, используемая при создании веб-сайтов.
- Он популярен благодаря своему надежному и прямолинейному характеру. Это один из самых популярных фреймворков во всем мире, который используется такими технологическими гигантами, как Instagram, Youtube и т.д.



# Плюсы Django

## Для чего лучше его использовать

- Структура приложений на джанго очень легко масштабируем
- DRF - очень мощная штука, которая поддерживает принципы Rest
- Безопасность

# Минусы Django

## Где могут быть проблемы?

- Менее гибкая система ORM (алхимия к примеру вроде как более гибкая???)
- Очень большая штука



# Fast API

Скорость и удобство





# Немного о Fast API

## Что надо о нем знать?

- Fast API - это современный, высокопроизводительный веб-фреймворк для создания API с Python 3.7+ на основе стандартных подсказок типа Python.
- Стоит он на плечах двух других фреймворков: работой с web занимается Starlette, а за валидацию отвечает Pydantic.

# Плюсы Fast API

## Почему он?

- Быстрый
- Асинхронный
- Легкий
- Интуитивный

# Минусы Fast API

## Почему его можно не выбрать?

- Все еще весьма минималистичный в сравнении с Django
- По нему сильно меньше хороших туториалов, так как довольно новый

Далее разговор по понятиям

# Model

## Что такое модель?

- Модель - это описание схемы табличных данных в виде класса в программном коде.



# View

## Вьюшка - что и зачем

- View - составная часть интерфейса, являющаяся элементом API, а с точки зрения кода реализующая точку входа в реализацию пользовательского действия.

# Controller

## Функциональная составляющая

- Controller - функция, решающая инкапсулированную задачу как черный ящик. Для нас они же будут utils.

# ORM

## Переезд жизни из таблиц и строчек в классы и объекты

- ORM - object relation mapper. Иначе говоря маппинг отношений данных в базе на отношения между объектами в ООП

# Test

## Что, зачем и как выглядит

- Test - сущность, описывающая поведение объекта и проверяющая соответствия ожиданий и действительности.
- AAA - Arrange, Act, Assert. Сначала настроим окружение, потом совершим действие, а затем проверим, что все прошло как надо.
- Тесты - лучшая документация.

# ADR

## Заметки архитектора

- ADR - документирование выпадающих вещей из общей концепции.

# Миграция

## Что и зачем

- Миграция - изменение представлений в таблицах и/или изменение самих данных. Есть миграция данных и схем. Для нас миграции - инструмент версионирования БД.

# Сериализация

## Про раззуливание форматов

- Сериализация - процесс преобразования данных.

# Валидация

## Всякие проверки

- Валидация - процесс проверки соответствия данных требованиям к форматам и контрактам.