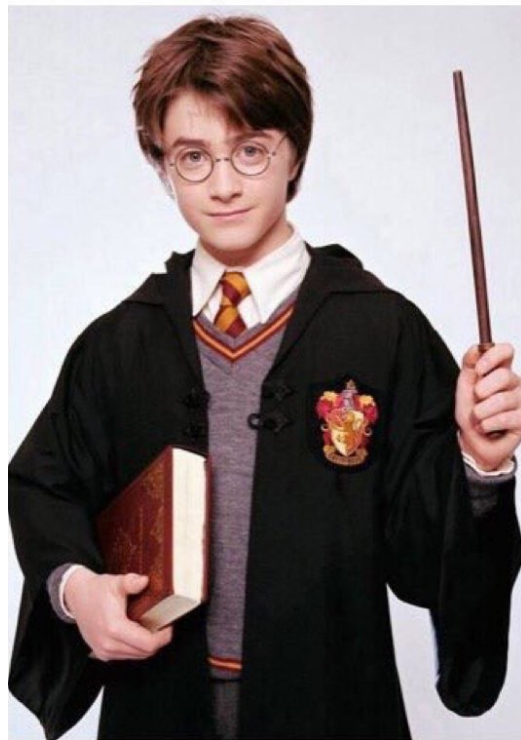


# Микросервисы

# План лекции

- Микросервисы: определение, преимущества и принципы;
- Сокеты, TCP, библиотеки `socket`, `beautiful soup`;
- Написание тривиального демона.

**Разработчики  
в начале проекта**



**Разработчики  
в конце проекта**



# Что это такое микросервисы?

**Микросервисы** - это небольшие, автономные, совместно работающие сервисы.

Небольшие и нацеленные на то, чтобы хорошо справляться только с одной работой.

# Основные преимущества

- Технологическая разнородность;
- Устойчивость;
- Масштабирование;
- Простота развёртывания;
- Компонуемость;
- Оптимизация с последующей замены.

# Немного критики

- Сетевые задержки;
- Форматы сообщений;
- Баланс нагрузки и отказоустойчивости;

# Принципы микросервисов



Написание  
демона



# Требования

- Во Flask-приложении из сообщения берём первый урл;
- Отправляем url демону, получаем json, содержащий поля url, title, img, desc;
- Отдаём фронтэнду информацию, как url можно отрисовать;

# Сокеты

**Сокеты** (англ. socket — разъём) — название программного интерфейса для обеспечения обмена данными между процессами.

Суть работы:

- на одном компьютере программа открывает сокет;
- слушает какой-то порт (в случае с TCP и UDP);
- другая программа, указав IP и этот самый порт, подключается к слушающей порт программе;
- дальше они обмениваются какими надо данными, после чего закрывают соединение.

# Сервер ТСП

Суть ТСП-соединения:

- одна программа устанавливает соединение с другой;
- происходит обмен данными;
- после завершения работы соединение должно быть закрыто.

Данные в ТСП — это поток байтов. Разделять его на отдельные сообщения придётся самой программе.

# Некоторые полезные функции модуля socket

- `bind` - связывает сокет с одним или всеми из имеющихся у компьютера хостов (IP-адресов) и каким-либо свободным портом;
- `listen` - прослушивание порта;
- `accept` - принимает соединение. Ждёт появление входящего соединения и возвращает связанный с ним сокет и адрес;
- `recv` - чтение данных. Тип возвращаемых данных — `bytes`;
- `send` - отправка данных в сокет.
- `close` - закрытие сокета

Практика

# Домашнее задание

- Написать собственный микросервис, добавляющий доп. функционал мессенджеру, например по урлу возвращает заголовок, описание и картинку (4 балл);
- Сделать вызов микросервиса асинхронным (3 баллов);
- Добавление сервиса в docker для проекта (3 балла);