



Пишем API приложение на Python с помощью FastApi и Docker

Пишем API приложение на Python с помощью FastApi и Docker

Introduction



Антон Козаченко
Python Back-end Developer

 anton_kazak

 kazak600



Пишем API приложение на Python с помощью FastApi и Docker

План

1. Ознакомление с фреймворком **FastApi**
2. Написание несколько endpoint-ов API
3. Написание тестов для этих endpoint-ов
4. Написание конфигурации для **Docker**
5. Запуск приложения в **Docker** containers
6. Запуск тестов



Видео курсы ITVDN для Python разработчиков



Python 3.8.6 Стартовый

Автор: Бондаренко Кирилл
Продолжительность: 4 ч 43 м
10 уроков



Python 3.8.6 Базовый

Автор: Бондаренко Кирилл
Продолжительность: 5 ч 52 м
9 уроков



Python Углубленный

Автор: Евгений Хацко
Продолжительность: 13 ч 51 м
8 уроков



Flask

Автор: Романюк Влад
Продолжительность: 6 ч 28 м
10 уроков



Django Стартовый

Автор: Михаил Лазорык
Продолжительность: 18 ч 27 м
14 уроков



Django Базовый

Автор: Михаил Лазорык
Продолжительность: 13 ч 10 м
11 уроков



Создание проекта на Python и Flask

Автор: Ксения Крементарь
Продолжительность: 11 ч 12 м
14 уроков



Основы работы с Git

Автор: Анна Маргина
Продолжительность: 2 ч 13 м
6 уроков

Пишем API приложение на Python с помощью FastApi и Docker

FastApi framework

FastAPI — веб-фреймворк, написанный на языке программирования Python для создания REST API, основанный на новых возможностях Python 3.6+, таких как: подсказки типов (type-hints), нативная асинхронность (asyncio). Помимо всего прочего, FastAPI плотно интегрируется с OpenAPI-schema и автоматически генерирует документацию для вашего API посредством Swagger и ReDoc.

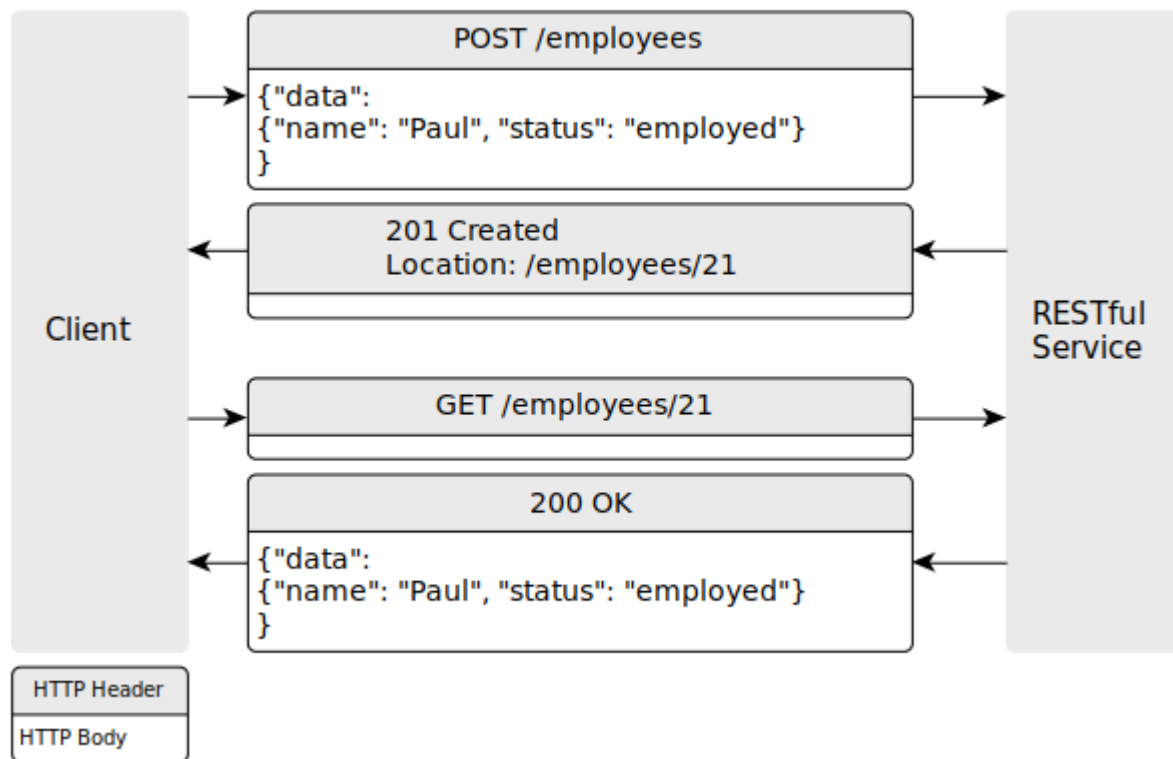
FastAPI построен на базе **Starlette** и **Pydantic**.

Starlette — ASGI микро-фреймворк для написания веб-приложений.

Pydantic — библиотека для парсинга и валидации данных, основанная на Python type-hints.

Пишем API приложение на Python с помощью FastApi и Docker

REST API



Пишем API приложение на Python с помощью FastApi и Docker

Пример API хендлеров GET и POST запроса

GET запрос на получение юзера

```
@router.get('/user', name='user:get')
def get_user(token: AuthToken = Depends(check_auth_token), database=Depends(connect_db)):
    user = database.query(User).filter(User.id == token.user_id).one_or_none()
    return {'user': user.get_filtered_data()}
```

POST запрос для логина

```
@router.post('/login', name='user:login')
def login(user_form: UserLoginForm = Body(..., embed=True), database=Depends(connect_db)):
    user = database.query(User).filter(User.email == user_form.email).one_or_none()
    if not user or get_password_hash(user_form.password) != user.password:
        return {'error': 'Email/password doesnt match'}

    auth_model = AuthToken(token=str(uuid.uuid4()), user_id=user.id, created_at=datetime.now())
    database.add(auth_model)
    database.commit()
    return {'auth_token': auth_model.token}
```

Пишем API приложение на Python с помощью FastApi и Docker

Структура API нашего приложения

default



POST /api/login User:Login



GET /api/user User:Get



POST /api/user User:Create



GET /api/stream Stream:Get



PUT /api/stream Stream:Update



POST /api/stream Stream:Create



Тестирование API

- **Pytest** framework используем для запуска тестов
- **Requests** библиотеку для http запросов



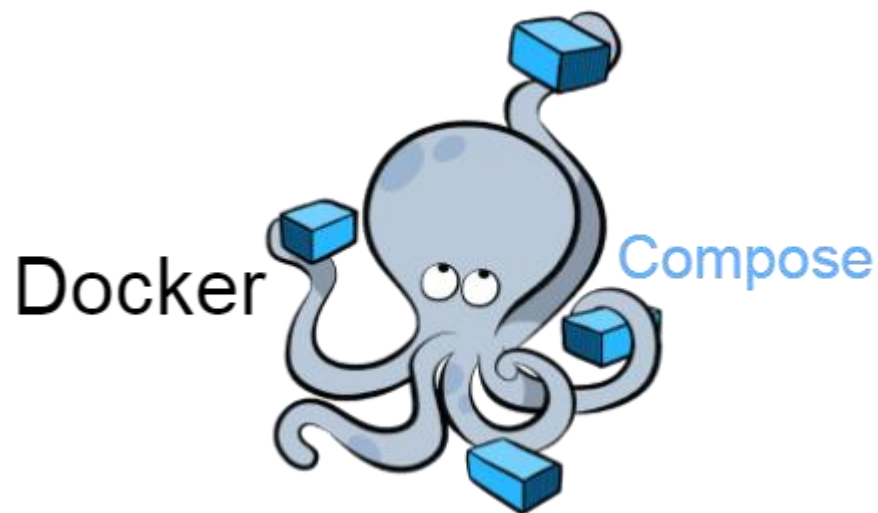
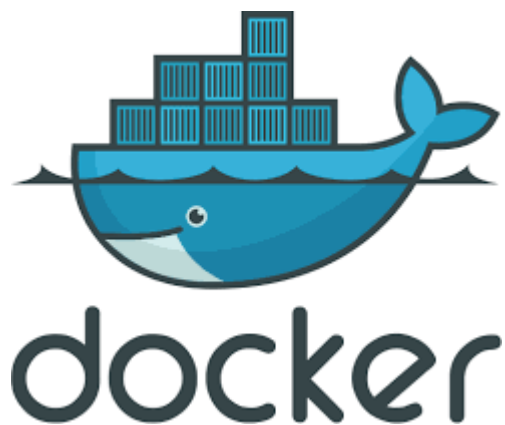
Пишем API приложение на Python с помощью FastApi и Docker

Docker и Docker-compose

Docker — это платформа для разработки, доставки и запуска контейнерных приложений. Docker позволяет создавать контейнеры, автоматизировать их запуск и развертывание, управляет жизненным циклом. Он позволяет запускать множество контейнеров на одной хост-машине.

Docker version 20.10.10

1. Docker-compose version 1.24.1



Q&A

Обучение онлайн с тренером по специальности Python Developer



LIVE ONLINE

PYTHON DEVELOPER

- Старт курса: с 29 ноября
- График: 3 раза в неделю

- Уровень: с нуля
- 41 занятий по 2 часа

БЕСПЛАТНЫЙ ОЗНАКОМИТЕЛЬНЫЙ УРОК

Смотрите видео курсы ITVDN

ВИДЕО КУРСЫ ПО СПЕЦИАЛЬНОСТИ PYTHON DEVELOPER

● Python 3.8.6 Стартовый	🕒 4 ч 43 м
● Python 3.8.6 Базовый	🕒 5 ч 52 м
● Python Углубленный	🕒 13 ч 51 м
● Flask	🕒 6 ч 28 м
● Создание проекта на Python и Flask	🕒 11 ч 12 м
● Основы MongoDB	🕒 2 ч 57 м
● PostgreSQL	🕒 3 ч 5 м
● MySQL Базовый	🕒 7 ч 58 м
● HTML5 и CSS3 Стартовый	🕒 8 ч 0 м
● JavaScript Стартовый	🕒 8 ч 41 м
● Django Стартовый	🕒 18 ч 27 м
● Django Базовый	🕒 13 ч 10 м
● Основы работы с Git	🕒 2 ч 13 м
● Методологии управления проектами. Вступление в SCRUM	🕒 1 ч 53 м
● Подготовка к собеседованию в IT компании. Вопросы и ответы. Хитрости. Трюки.	🕒 1 ч 50 м

ЗАПИСИ БЕСПЛАТНЫХ ВЕБИНАРОВ

● Как стать Python разработчиком в 2021 году?	1 ч 53 м
● Создание простой казуальной игры на Python	1 ч 51 м
● Создание чат-бота "прогноз погоды" на Python	2 ч 54 м
● Объектно-ориентированное программирование в Python	2 ч 2 м
● Автоматизация парсинга сайтов на Python	2 ч 4 м
● Создаем нейронную сеть на Keras с нуля	2 ч 14 м
● Структуры данных в Python. Уровень Advanced	2 ч 3 м
● Знакомство с Python	2 ч 13 м
● Как стать Python разработчиком?	1 ч 31 м
● Создание первого веб-сайта при помощи Python и Django	1 ч 28 м
● Создание социальной сети с помощью Python	1 ч 46 м
● Размещение веб-приложений на Python в облаке Microsoft Azure	2 ч 4 м
● Архитектура Flask приложений	2 ч 4 м
● SQLAlchemy ORM: удобная работа с базами данных на Python	1 ч 36 м
● PySharp - практическое использование Python в .NET приложениях	2 ч 40 м
● Что нового в Python? Спецификация PEP 8	1 ч 46 м
● Разработка сервиса обмена фотографиями на Django	1 ч 40 м

● Обязательные курсы ● Дополнительные курсы ● Бесплатные вебинары

Проверка знаний

TestProvider.com



Проверьте Ваши знания на
[TestProvider.com](https://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 тестирования позволит Вам получить соответствующий Сертификат.

Информационный видеосервис для разработчиков программного обеспечения

