# Создание CRUD-приложения с помощью FastAPI

*Учебный материал необходимо дополнить информацией. Данные которые нужно добавить в документе указывается тремя точками …. Не забудьте указать источники, которые Вы используете при дополнении материала.*

FastAPI — это фреймворк для создания лаконичных и довольно быстрых HTTP API-серверов со встроенными валидацией, сериализацией и асинхронностью.

Важные причины выбрать FastAPI (найдите определение, что это?):

* **Асинхронность**

Асинхронность означает, что в языке есть возможность сообщить машине / программе, что в определённой точке кода ей нужно будет ожидать завершения выполнения чего-то ещё в другом месте. Допустим это что-то ещё называется "медленный файл".

* **Типизированность**

У переменных есть свой тип данных, за счет чего мы всегда знаем какие есть методы, её функции и.т.д у данной переменной. Пародия на Java. Приятный бонус.

* **Встроенная документация (Swagger)**

FastApi автоматически генерирует документацию в формате OpenAPI и отдаёт её в формате Swagger/ReDoc в зависимости от где вы смотрите, ваш\_сайт/docs или ваш\_сайт/redoc соответственно.

* **Применение websockets**

Протокол WebSocket («веб-сокет»), описанный в спецификации [RFC 6455](http://tools.ietf.org/html/rfc6455), обеспечивает возможность обмена данными между браузером и сервером через постоянное соединение. Данные передаются по нему в обоих направлениях в виде «пакетов», без разрыва соединения и дополнительных HTTP-запросов. WebSocket особенно хорош для сервисов, которые нуждаются в постоянном обмене данными, например онлайн игры, торговые площадки, работающие в реальном времени, и т.д.

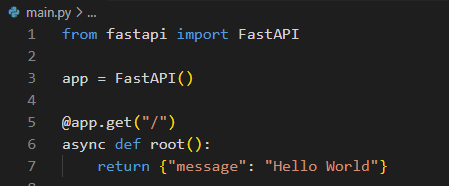
Документация <https://fastapi.tiangolo.com/>

Как установить FastAPI

pip install fastapi

pip install uvicorn

**uvicorn main:app –reload** //запустить сервер



Чтобы запустить приложение нужно набрать в терминале:

uvicorn main:app --reload

* **main** это имя файла (точка входа)
* **app**  the object created inside of main.py with the line app = FastAPI().
* **--reload** make the server restart after code changes. Only do this for development.

Для автоматических интерактивных документов API необходимо использовать следующий адрес:

* **http://127.0.0.1.8000/docs -> SwaggerUI**

Для визуализации и взаимодействия с ресурсами API без какой-либо логики реализации. Он автоматически создается из вашей спецификации OpenAPI (ранее известной как Swagger) с визуальной документацией, упрощающей реализацию на стороне клиента и использование на стороне клиента.

* **http://127.0.0.1.8000/redoc -> Redoc**

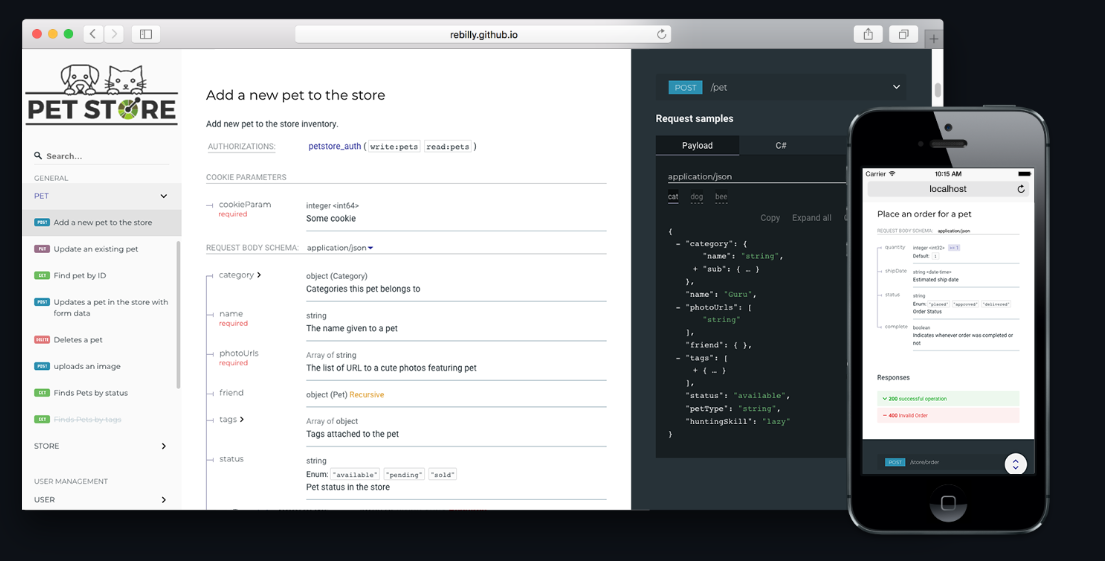
это инструмент с открытым исходным кодом для создания документации из определений OpenAPI (fka Swagger).

По умолчанию Redoc предлагает адаптивный макет с тремя панелями:

Левая панель содержит строку поиска и меню навигации.

Центральная панель содержит документацию.

Правая панель содержит примеры запросов и ответов.

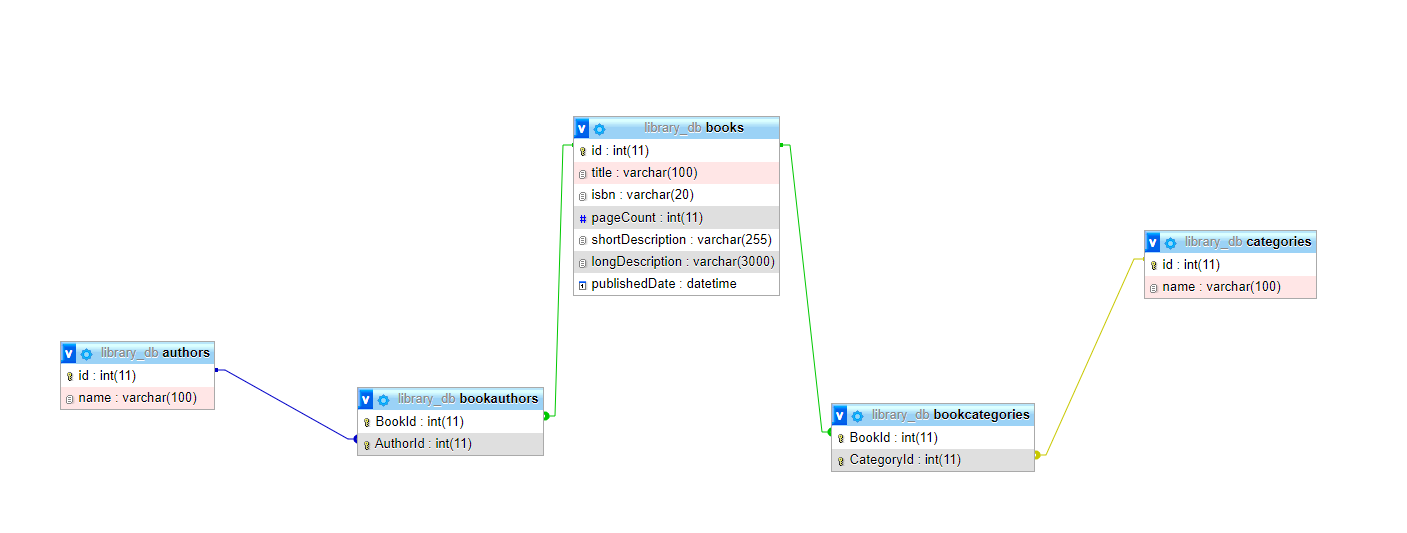


**OpenAPI (swagger)**

OpenAPI — это открытая спецификация, управляемая сообществом в рамках инициативы OpenAPI, совместного проекта Linux Foundation.

Спецификация OpenAPI (OAS) определяет стандартное описание интерфейса HTTP API, не зависящее от языка программирования, которое позволяет как людям, так и компьютерам обнаруживать и понимать возможности службы, не требуя доступа к исходному коду, дополнительной документации или проверки сетевого трафика. . При правильном определении через OpenAPI потребитель может понимать и взаимодействовать с удаленной службой с минимальным количеством логики реализации.

Для примера используется база данных MySQL про книги. См файл library\_db.sql



**Опишите ERD диаграмму. Какие таблицы, поля, ключи содержит база данных.**

Есть таблицы с определенными полями. От главной таблицы books отходят две ветки по две таблицы в каждой. Первые таблицы каждой ветки связываются с главной по id книги. Т.е. можно перейти в первые таблицы веток(foreign) по id книги(primary). 1 к 1. От первых таблиц каждой ветки отходит по 1 таблице в каждой, которая связана точно таким же способом. 1 к 1.

Для работы с базами данных можно использовать следующие библиотеки, которые нужно установить:

SQLAlchemy - <https://www.sqlalchemy.org/docs/> - pip install SQLAlchemy

pip install mysql-connector-python

В проекте следующие файлы

- crud.py : методы CRUD (чтение и изменение данных)

- database.py : для соединения с базой данных MySQL

- main.py: веб-сервисы FastAPI

- models.py: определение моделей таблиц

- schemas.py: определение схем для обработки request/response

Прокомментируйте коды данных скриптов.

Добавьте еще методы для просмотра, добавления, удаления данных.

# Используемые источники

1. Почему вам стоит выбрать фреймворк FastAPI для своего следующего проекта <https://tproger.ru/articles/pochemu-vam-stoit-vybrat-frejmvork-fastapi-dlja-svoego-sledujushhego-proekta/>

2. <https://tproger.ru/articles/pochemu-vam-stoit-vybrat-frejmvork-fastapi-dlja-svoego-sledujushhego-proekta/>

3. <https://fastapi.tiangolo.com/ru/>

4. <https://learn.javascript.ru/websocket>

5.