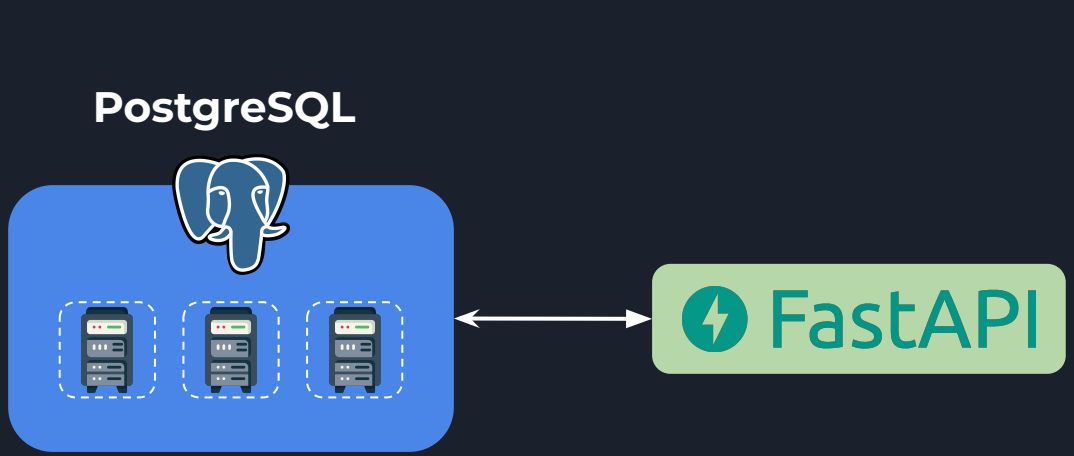
Back-End информационной системы оборудования установки SPD

Докладчик: **Двизов Борис Павлович**

Научные руководители: **Прокошин Ф.В.**, канд. физ.-мат. наук, старший научный сотрудник Лаборатории ядерных проблем им. В. П. Джелепова Объединенного института ядерных исследования
Тваури И.В., канд. физ.-мат. наук, доцент кафедры физики конденсированного состояния;



Device

PostgreSQL

FastAPI

Проектирование базы данных и разработка Rest API сервиса для хранения, получения и обновления информации о компонентах установки SPD. Число компонент может достигать нескольких миллионов, созданных на основе тысяч различных типов и организованных в десятки гетерогенных подсистем.

Цель работы

Стек технологий

PostgreSQl

1. Надежность: PostgreSQL обеспечивает высокую степень надежности и целостности данных.

2. Производительность: PostgreSQL имеет эффективный оптимизатор запросов и поддерживает параллельное выполнение запросов.

3. Гибкость: PostgreSQL поддерживает различные модели данных, включая реляционную модель, JSON, XML, геоданные и другие.

FastAPI

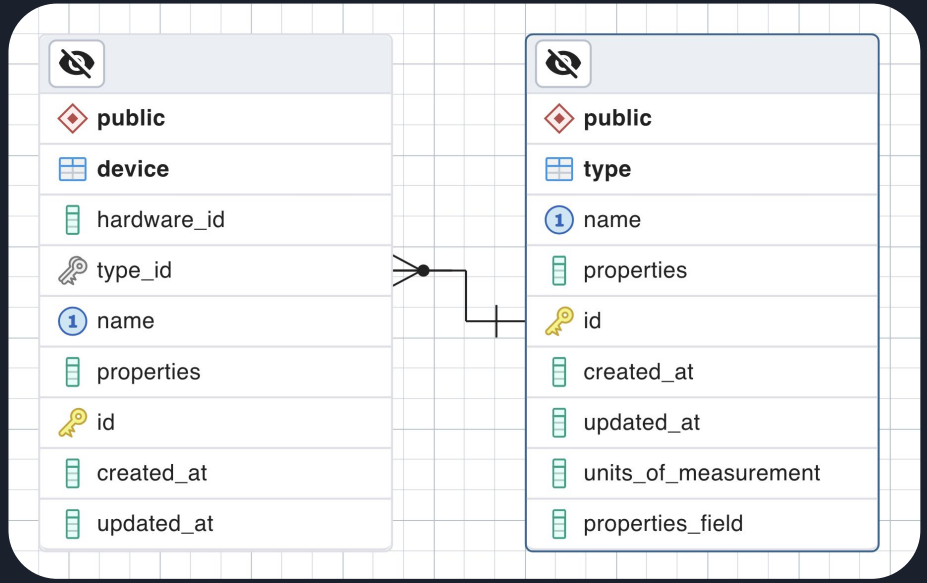
1. Высокая производительность: FastAPI обеспечивает высокую скорость обработки запросов и масштабируемость.

2. Ассинхронность: FastAPI поддерживает ассинхронные запросы

3. Автоматическая документация: FastAPI автоматически генерирует интерактивную документацию API на основе аннотаций кода.

- Alembic - библиотека для миграции таблиц в базу данных.
- SQLAlchemy - библиотека для моделирования таблиц и создания запросов к базе данных, а также служит прослойкой между клиентскими и машинными запросами к базе данных.
- Pydantic - библиотека валидации данных, написания схем.

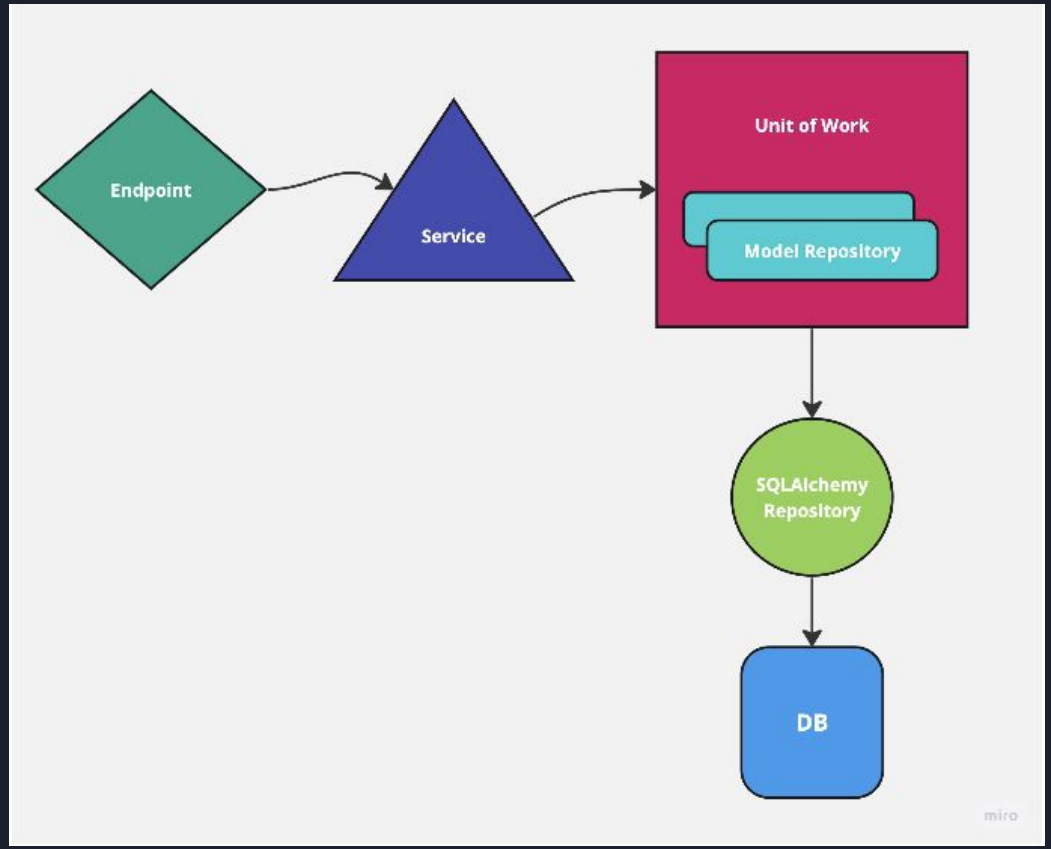
Библиотеки



Столбцы таблицы типов: (**id**) уникальный идентификатор типа, (**name**) имя типа, (**properties**) параметры типа - состоят из пары ключ, значение, где ключ - название параметра, а значение - тип данных параметра, (**properties_filed**) минимальные, максимальные и значения по умолчанию для параметров типа, (**units_of_measurement**) единицы измерения параметров, (**created_at**) время создания, (**updated_at**) время обновления

Столбцы таблицы устройств: (**id**) уникальный идентификатор устройства в базе данных, (**hardware_id**) уникальный идентификатор устройства, (**type_id**) id типа устройства, (**name**) имя устройства, (**properties**) параметры устройства состоят из пары ключ, значение, где ключ - унаследованное из таблицы типов имя параметра, а ключ - значение параметра, (**created_at**) время создания, (**updated_at**) время обновления

Таблицы


Endpoint
Service
Unit of Work
Model Repository
SQLAlchemy Repository
DB
miro

Unit Of Work

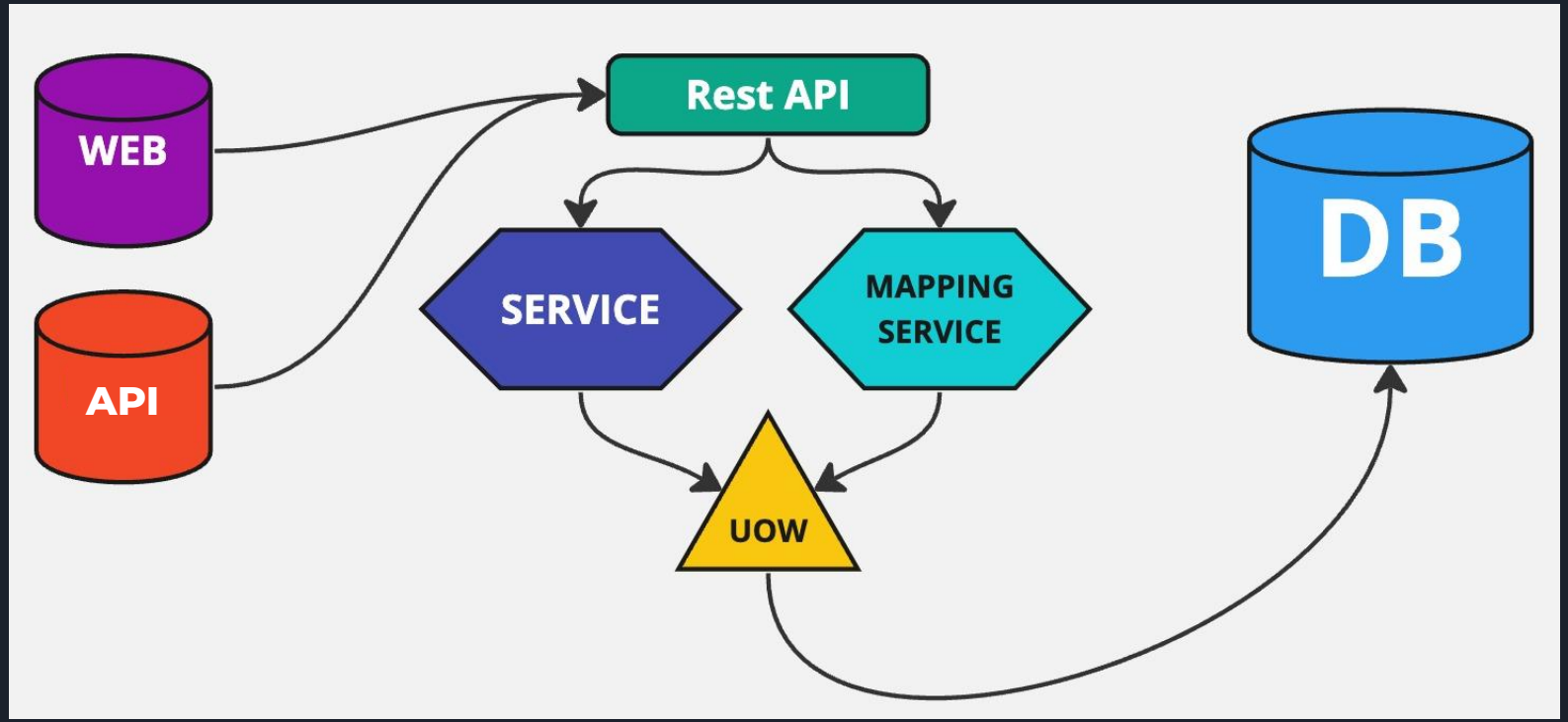
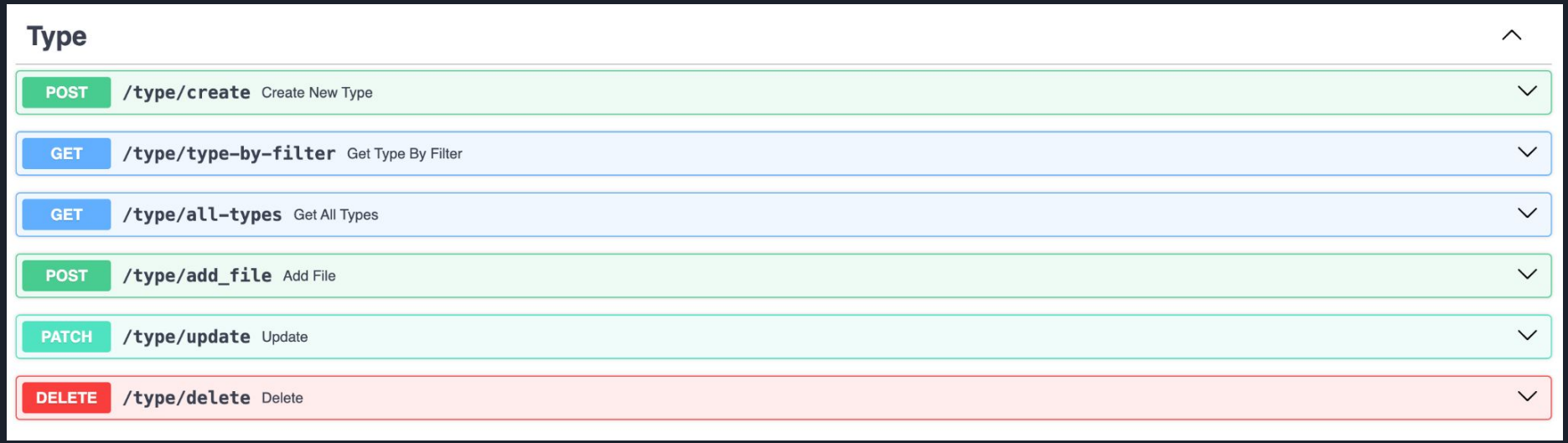


Схема работы сервиса



/create - создание типа
/type-by-filter - получение типа по имени или id
/all-types - получение всех типов
/add_file - создание нескольких типов
/update - обновление типа
/delete - удаление типа

Эндпоинты типов

/create - создание устройства
/get-by-datetime - получение устройства по времени создания
/get-by-type - получение устройства по типу
/get-by-hardwareId - получение устройства по hardware id
/all-devices - получение всех устройств
/add_file - создание нескольких устройств
/update - обновление устройства
/delete - удаление у

Эндпоинты устройств

- Спроектирована база данных на PostgreSQL
- Разработан Rest API сервис на FastAPI
- Подключена миграция таблиц в базу данных
- Интегрирован паттерн проектирования Unit Of Work
- Разработаны эндпоинты для работы с типами устройств
- Разработаны эндпоинты для работы с устройствами

Заключение

- Добавление эндпоинтов обеспечивающих развитый поиск
- Реализация наследования типов устройств
- Исходя из опыта использования системы и запросов пользователей - расширение функционала эндпоинтов
- Разработка клиентского API для использования из программ на Python, C++ и т.д
- Реализация аутентификации пользователя и прав пользователя

Развитие системы

Спасибо