Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» (УрФУ)

Институт радиоэлектроники и информационных технологий – РТФ

ОТЧЕТ

по дисциплине: «Основы веб-протоколов»

ФИО: Копытов Е. А.

Академическая группа: РИ-320910

Екатеринбург

2024

**Содержание**

[**ВВЕДЕНИЕ** 3](#_Toc187618687)

[**Основные файлы** 4](#_Toc187618688)

[**Парсер** 6](#_Toc187618689)

[**API** 8](#_Toc187618690)

[**1. Взять все данные всех продуктов** 8](#_Toc187618691)

[**2. Создать новый продукт** 9](#_Toc187618692)

[**3. Удалить все продукты, которые были сохранены** 11](#_Toc187618693)

[**4. Взять конкретный продукт по id** 11](#_Toc187618694)

[**5. Изменить конкретный продукт по id** 13](#_Toc187618695)

[**6. Удалить конкретный продукт по id** 14](#_Toc187618696)

[**Websockets** 17](#_Toc187618697)

[**Заключение** 18](#_Toc187618698)

# **ВВЕДЕНИЕ**

**Задание**: Необходимо создать API которое будет фоном парсить данные с сайта и сохранять их в БД. Доступ к данным нужно осуществить через RESTApi с возможность их редактирования и удаления. Все манипуляции с данными должны генерировать уведомления при помощи WebSocket.

Использовался python 3.12.Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание Код всего проекта можно разделить на два модуля: api, parser (рисунок 1).

Рисунок 1. Код проекта

# **Основные файлы**

Данный файл (рисунок 2) выполняет следующие функции:

* Создание таблиц в базе данных, если они ещё не существуют;
* Запуск парсера в другом процессе;
* Запуск api через uvicorn.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2. Основной файл

Данный файл запускается при запуске программы. Строки 12–14, 19–20 отвечают за создание таблиц в БД, если их еще нет. В 23–24 запускается парсер в другом процессе, чтобы не мешать основному процессу обрабатывать информацию из api. + чтобы можно было масштабировать api не затрагивая парсер. В 27 строке запускается api.

Ниже на рисунке 3 расположен код, отображающий модели в БД. Product – модель отображающая продукт с сайте для парсинга. Event – модель отражает события, которые регистрируются при различных событиях

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеРисунок 3. Модели проекта

Ниже на рисунке 4 расположен код для получения асинхронного движка для SQLAlchemy.

Изображение выглядит как текст, снимок экрана, Шрифт, белый

Автоматически созданное описаниеРисунок 4. Создание движка

# **Парсер**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеПарсер состоит из двух файлов. В первом хранится сам парсер, который мы делали в прошлой лабораторной работе, а также вспомогательные методы. Во втором схема данных продукта.

Рисунок 5. Запуск парсера

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описаниеЭтот вспомогательный метод нужен для того, чтобы при запуске парсера в другом процессе от корректно работал и завершался. 130 и 131 строчка отлавливают сигналы и приводят к тому, что процесс завершиться только если вызвать terminate(). 133 сточка нужна также для корректного завершения программы (Отлавливает KeyboardInterrupt от terminate).

Рисунок 6. Тело процесса парсера

Данный метод парсит сайт и записывает полученную информацию в бд. Причем из-за 119–121 строк в случаи, если на странице поменялась цена он перезапишет данные (При этом данные не дублируются).

Также парсер регистрирует событие «Парсинг каталога успешно завершен» в конце парсинга.



Рисунок 7. Создание ивента

# **API**

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описаниеНа рисунке 8 слева код моделей pydantic для валидации данных из api. На рисунке 8 справа код для создания асинхронных сессий для подключения к бд.

Рисунок 8. Код

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеВ api поддерживаются 6 методов работы с данными (рисунок 9).

Рисунок 9. Список всех методов

## **1. Взять все данные всех продуктов**

Код можно посмотреть ниже на рисунке 10.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеРисунок 10. Реализация метода

Пример вывода метода:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описаниеРисунок 11. Пример ответа

## **2. Создать новый продукт**

Код можно посмотреть ниже на рисунке 12.

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описаниеРисунок 12. Реализация метода

При вводе правильных данных:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описаниеРисунок 13. Пример вывода

Будет получен ответ со статусом 201:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описаниеРисунок 14. Пример вывода

## **3. Удалить все продукты, которые были сохранены**

Код можно посмотреть ниже на рисунке 15.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описаниеРисунок 15. Реализация метода

Пример вывода программы.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описаниеРисунок 16. Пример вывода

## **4. Взять конкретный продукт по id**

Код можно посмотреть ниже на рисунке 17.

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описаниеРисунок 17. Реализация метода

Если ввести существующий id:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описаниеРисунок 18. Пример вывода

То будет получен следующий ответ:

Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Автоматически созданное описаниеРисунок 19. Пример вывода

Если ввести несуществующий id:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описаниеРисунок 20. Пример вывода

## **5. Изменить конкретный продукт по id**

Код можно посмотреть ниже на рисунке 21.

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описаниеРисунок 21. Реализация метода

Если изменить продукт 136, выставив, например только цену, то он успешно обработается.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описаниеРисунок 22. Пример вывода

Результаты после изменеения можно увидеть тут:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описаниеРисунок 23. Пример вывода

## **6. Удалить конкретный продукт по id**

Код можно посмотреть ниже на рисунке 24.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеРисунок 24. Реализация метода

Если удалить существующий продукт:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описаниеРисунок 25. Пример вывода

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описаниеЕсли удалить продукт, которого нет в бд:

Рисунок 26. Пример вывода

# **Websockets**

Данный метод отправляет пользователю все ивенты, созданные методами обрабатывающие данные с сайта maxidom.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеРисунок 27. Реализация метода

# **Заключение**

Был создан парсер сайта maxidom, сохраняющий все данные в бд. Также был создана api с возможностью редактировать, получать и удалять эти данные. Также любые действия с данными генерируют уведомления и их можно получить через websocket.