Сырятов Максим Алексеевич РИ-320932 АТ-01

Итоговый проект по WebApi.

**Введение**

В рамках данного проекта была поставлена задача разработать инструмент для автоматизированного сбора информации о товарах и ценах с веб-сайтов (парсер) и работу с ними через Rest API для удобного управления собранными данными. После чего подключенный клиент должен получать уведомления о каждом действии через websocket. Основные цели задания заключались в разработке функционального парсера, создании базы данных для хранения информации о товарах, реализации API, через который пользователи могли бы запрашивать данные и управлять ими, а также отправка уведомлений через websocket подключенному клиенту.

**Основная часть**

Моё решение состоит из нескольких частей:

1. Парсер. Было решено использовать библиотеки Beautiful Soup и Requests для загрузки и парсинга HTML-кода. Основная функция get\_products\_from\_page(url) отвечает за загрузку страницы и извлечение информации о товарах. Она написана для сайта <https://at-store.ru/internet-shop/kupit_apple_iphone/>. Но при небольших изменениях в тегах подойдет для других сайтов. Разработанная функция parse\_products(base\_url) позволяет извлекать данные о товарах с нескольких страниц, учитывая пагинацию, что значительно увеличивает объем собираемой информации.
2. Модель базы данных. Для хранения данных была использована SQLite, так как она наиболее простая и быстрая в освоении. С помощью SQLAlchemy была разработана модель Product, которая описывает структуру таблицы с полями для идентификатора, названия товара и его цены.
3. Основной раздел, в котором описаны базовые методы работы с базой данных и парсером, такие как парсинг сайта с добавлением товаров в базу данных, получение списка товаров, добавление нового товара, обновление данных и удаление товаров. Реализована обработка различных типов запросов, а также обработка возможных ошибок, таких как несуществующие товары или некорректные данные. Попутно каждый метод возвращает уведомление клиенту через websocket.

Ниже представлены примеры работы с моим решением.

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

Автоматически созданное описание

Рисунок 1 – запрос к сайту по введенному url.

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

Автоматически созданное описание

Рисунок 2 – результат работы функции GetProducts через запрос Get.

![Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание]()

Рисунок 3 – добавление элемента в базу данных.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 4 – результат добавления элемента корректен.

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

Автоматически созданное описание

Рисунок 5 – пробуем удалить элемент.

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

Автоматически созданное описание

Рисунок 6 – элемент удален корректно.

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

Автоматически созданное описание

Рисунок 7 – изменение товара с id 4.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 8 – изменение прошло успешно.

Изображение выглядит как текст, программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 9 – пример уведомлений, получаемых при помощи websocket.

**Вывод:**

Результат был достигнут, имеется WebApi, способное парсить товары с сайта, добавлять их в базу данных, получать список товаров, изменять товары, удалять товары, а также отправлять уведомления через websocket. При небольших доработках Api способно работать с другими сайтами. Проект удобен для расширения и добавления новых функций для работы с базой данных и парсером.