**Министерство науки и высшего образования Российской Федерации  
ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»**

Школа бакалавриата

Оценка работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель от УрФУ Свинцов Д. В.

Тема задания на практику

**Основы Веб API и базовые Веб протоколы**

ОТЧЕТ

Вид практики: Учебная практика

Тип практики: Детальное изучение протокола http. Ознакомление с преимуществами и недостатками, а также собственноручное написание Rest API.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_ Гайдабура О. А.

**Подпись студента ФИО студента**

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Группа: РИ-200001

Екатеринбург 2022

Оглавление

[Введение 3](#_Toc109111611)

[1 Основы работы REST API на Python 4](#_Toc109111612)

[1.1 Основы Telnet и OpenSSL 4](#_Toc109111613)

[1.2 Написание парсера. Библиотеки bs4 и requests 6](#_Toc109111614)

[1.3 Библиотека SQLAlchemy 8](#_Toc109111615)

[2 Итоговая работа 9](#_Toc109111616)

[2.1 Написание парсера сайта 10](#_Toc109111617)

[2.2 Написание API 11](#_Toc109111618)

[Заключение 14](#_Toc109111619)

[Список использованных источников 15](#_Toc109111620)

[Задание на практику 16](#_Toc109111621)

[Форма отзыва руководителя 17](#_Toc109111622)

# Введение

REST API — это способ взаимодействия сайтов и веб-приложений с сервером. Его также называют RESTful.

Термин состоит из двух аббревиатур, которые расшифровываются следующим образом. API (Application Programming Interface) — это код, который позволяет двум приложениям обмениваться данными с сервера. На русском языке его принято называть программным интерфейсом приложения. REST (Representational State Transfer) — это способ создания API с помощью протокола HTTP. На русском его называют «передачей состояния представления».

Технологию REST API применяют везде, где пользователю сайта или веб-приложения нужно предоставить данные с сервера. Например, при нажатии иконки с видео на видеохостинге REST API проводит операции и запускает ролик с сервера в браузере. В настоящее время это самый распространенный способ организации API.

Целью учебной практики является:

* Изучение протокола HTTP + практика
* Изучение принципов REST API + практика
* Сравнение подхода HTTP/REST API c WebSocket + практика
* Итоговая работа REST API на Python

# 1 Основы работы REST API на Python

## 1.1 Основы Telnet и OpenSSL

TELNET - сетевой протокол для реализации текстового терминального интерфейса по сети.

Назначение протокола TELNET — в предоставлении достаточно общего двунаправленного восьмибитного байт-ориентированного средства связи. Его основная задача заключается в том, чтобы позволить терминальным устройствам и терминальным процессам взаимодействовать друг с другом. Предполагается, что этот протокол может быть использован для связи вида терминал-терминал («связывание») или для связи процесс-процесс («распределенные вычисления»).

В ходе практической работы TELNET использовался для подключения к сайту и проверки статуса текущего подключения.

OpenSSL — это криптографический инструментарий, реализующий сетевые протоколы Secure Sockets Layer (SSL v2/v3) и Transport Layer Security (TLS v1) и соответствующие им стандарты криптографии.

Программа OpenSSL — это инструмент командной строки для использования различных криптографических функций криптографической библиотеки OpenSSL в консоли.

Основныe возможности:

* Создание и управление закрытыми ключами, открытыми ключами и параметрами.
* Криптографические операции с открытым ключом
* Создание сертификатов X.509, CSR и CRL
* Расчёт дайджестов сообщений
* Шифрование и дешифрование с помощью шифров
* Клиентские и серверные тесты SSL/TLS
* Обработка подписанной или зашифрованной почты S/MIME
* Запросы отметок времени, генерация и проверка

На рисунке 1 можно увидеть успешный результат подключения к сайту [www.kaspersky.com](http://www.kaspersky.com)

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 1

## 1.2 Написание парсера. Библиотеки bs4 и requests

Парсер — это программа для сбора и систематизации информации, размещенной на различных сайтах. Незаменимыми помощниками в написании парсеров на python являются библиотеки bs4 и requests. Рассмотрим каждую их них поподробнее.

Beautiful Soup (bs4) — это библиотека Python для извлечения данных из файлов HTML и XML. Она работает с вашим парсером, чтобы дать вам естественные способы навигации, поиска и изменения дерева разбора. При проходе HTML-документа через Beautiful Soup нам возвращается объект BeautifulSoup, который представляет документ в виде вложенной структуры данных. Далее вы можете увидеть пример использования библиотеки bs4.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – изначальный HTML-документ

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – Результат работы библиотеки bs4

Помимо этого, данная библиотека имеет множество других полезных инструментов для парсинга различных сайтов.

Библиотека requests является стандартным инструментом для составления HTTP-запросов. Библиотека requests позволяет:

* Создавать запросы посредством наиболее популярных HTTP-методов
* Редактировать заголовки запросов и данных с помощью строки запроса, а также содержимого сообщения
* Анализировать данные запросов и откликов
* Создавать авторизированные запросы
* Настраивать запросы с целью предотвращения сбоев и замедлений в работе приложения

## 1.3 Библиотека SQLAlchemy

SQLAclhemy — это библиотека для работы с реляционными базами данных в Python. SQLAlchemy позволяет работать с базами данных MySQL, MS-SQL, PostgreSQL, Oracle, SQLite и другими.

Самая важная особенность SQLAlchemy — это ее ORM. ORM или Object Relational Mapper (объектно-реляционное отображение) позволяет работать с базой данных с помощью объектно-ориентированного кода, не используя SQL-запросы, что позволяет очень удобно использовать ее для написания Rest API.

# 2 Итоговая работа

Темой итоговой работы я выбрал парсинг сайта кормов для животных, так как очень их люблю. Целью моей работы является парсинг разных сайтов с кормами для дальнейшего составления рейтинга качества, цены и прочих характеристик. Однако из-за сжатых сроков удалось выполнить только минимальный объем итоговой работы:

* Написание парсера сайта
* Написание API для работы с базой данных, полученной в результате парсинга

Для создания API был выбран фреймворк FastAPI, позволяющий создать лаконичный и быстрый HTTP-API сервер со встроенной валидацией, сериализацией и асинхронностью.

## 2.1 Написание парсера сайта

Для написания парсера я использовал ранее описанную библиотеки bs4, requests, SQLAlchemy.

Алгоритм работы парсера состоит из следующих шагов:

1. Импорт необходимых библиотек
2. Парсинг всех ссылок на страницы с продуктами
3. Создание базы данных и добавление необходимых полей
4. Написание функции, записывающей запись в таблицу (рис. 4)
5. Заполнение базы данных в теле цикла

Таблица имеет следующие поля:

* Id (номер записи в БД)
* Name (наименование товара)
* Url (ссылка на товар)
* Datetime (время внесения записи, для дальнейшего сравнения и обновления цен в случае изменения)
* Price (цена товара)

Изображение выглядит как текст

Автоматически созданное описаниеНиже представлена функция добавления записи в таблицу.

Рисунок 4

С полным кодом парсера вы можете ознакомиться по ссылке ниже.

## 2.2 Написание API

Как было сказано ранее для разработки был выбран фреймворк FastAPI. Для удобства использования рекомендуется соблюдать следующую структуру проекта:

* \_\_init\_\_.py (используется для инициализации пакета)
* crud.py (содержит все необходимые функции)
* database.py (создает или инициализирует уже готовую БД)
* main.py (содержит запросы)
* models.py (отвечает за структуру данных и связи между таблицами)
* schemas.py (определяет параметры возвращаемых данных)

При написании API я создал 5 запросов, рассмотрим их подробнее.

1) Запрос на создание записи в таблице и проверка на то нет ли в таблице идентичной записи. (рисунок 5)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5

2) Запрос на просмотр первых 100 записей в БД. (рисунок 6)

Изображение выглядит как текст, внутренний, снимок экрана

Автоматически созданное описание

Рисунок 6

3) Запрос на просмотр записи по ее ID. (рисунок 7)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7

4) Запрос на удаление записи из БД. (рисунок 8)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8

5) Запрос на изменение записи по ID. (рисунок 9)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9

Фреймворк FastAPI позволяет удобно просматривать, а также исполнять запросы всех типов. На рисунке 10 представлены все описанные запросы и их метод.

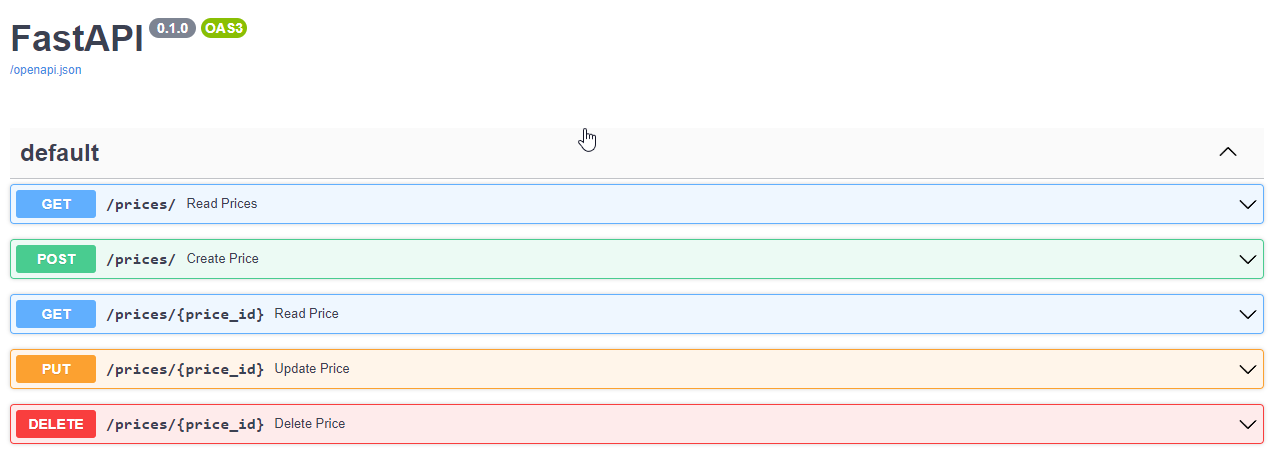


Рисунок 10 – интерфейс FastAPI

Для демонстрации работы запросов попробуем вывести запись по ее ID. Откроем щелчком мыши третий запрос, а затем введем ID продукта. (рисунок 11)

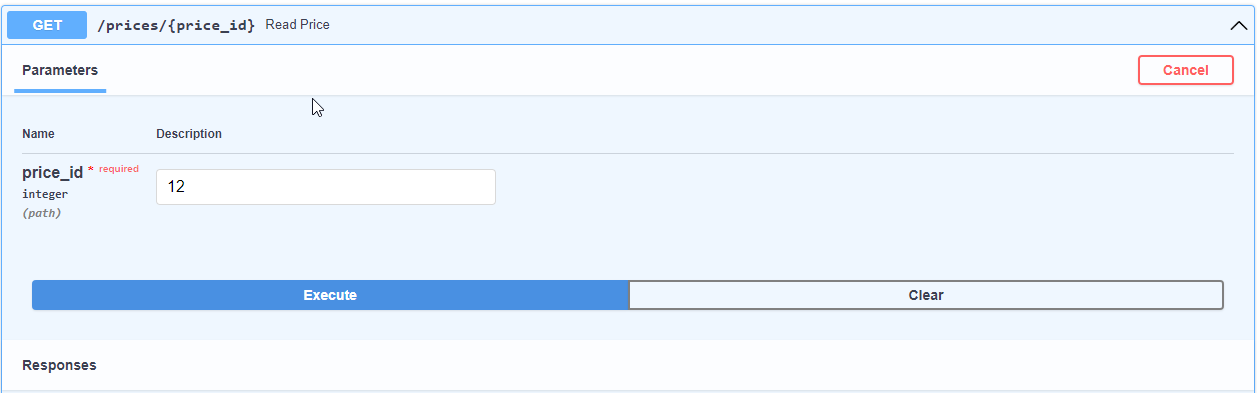


Рисунок 11

Далее нажимаем кнопку “Execute”, выполняем его и получаем результат. (рисунок 12)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12

Код – 200 означает что запрос выполнен успешно и запись имеющая ID=12 находится в теле ответа.

# Заключение

В ходе учебной практики были выполнены следующие задачи:

* Изучены протоколы HTTP
* Проведено ознакомление и работа с библиотеками Python, позволяющими производить парсинг веб-сайтов, а также работать с API и базами данных
* Выполнена итоговая работа
* Составлен отчет о ходе учебной практики

Вывод: Парсинг веб-сайтов является неотъемлемой частью работы со статистикой, так как позволяет быстро и удобно извлекать нужную информацию для дальнейшей работы с ней. Rest API в свою очередь, является очень гибким инструментом для выполнения запросов к серверу.

Облачный сервис deta с API: <https://w7nv5s.deta.dev/docs#/>

Полный код на github: https://github.com/OlegGaidabura/Practice2022

# Список использованных источников

1) <https://docs-python.ru>

2) <https://younglinux.info>

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

4) <https://web.deta.sh>

5) <https://habr.com>

# Задание на практику

**Министерство науки и высшего образования Российской Федерации**

**ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»**

|  |  |
| --- | --- |
|  | СОГЛАСОВАНО  УрФУ  «27» июня 2022 |
|  | Зав. кафедрой ИРИТ-РТФ  \_\_\_\_\_\_\_\_\_\_ И. Н. Обабков  Подпись расшифровка подписи |

Институт ИРИТ-РТФ Группа РИ-270012 Кафедра школа бакалавриата

Код, наименование направления 09.03.04-Программная инженерия

Наименование образовательной программы/Наименование магистерской программы 09.03.01 – Информатика и вычислительная техника

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

На учебную практику студента

Гайдабура Олег Андреевич

(фамилия, имя, отчество)

1. Тема задания на практику "Основы Веб API и базовые Веб протоколы"

2. Срок практики с 04.07.2022 г. по 17.07.2022 г. Срок сдачи студентом отчета c 01.09.2022 г. по 14.09.2022 г.

3. Место прохождения практики УрФУ, ИРИТ-РТФ, Школа бакалавриата

4. Вид практики: Учебная практика

5. Тип практики: Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности

6. Содержание отчета: титульный лист, оглавление, введение, основная часть, заключение, библиографический список

**Рабочий график (план)проведения практики**

|  |  |  |  |
| --- | --- | --- | --- |
| *Этапы практики* | *Наименование работ студента* | *Срок* | *Примечание* |
| организационный | Ознакомление с рабочей программой практики; изучение методических рекомендаций по практике; согласование индивидуального задания; усвоение правил техники безопасности и охраны труда | 04.07.2022-05.07.2022 |  |
| основной | Выполнение индивидуального задания, ежедневная работа по месту практики, мероприятия по сбору материала, заполнение отчета по практике | 06.07.2022-17.07.2022 |  |
| заключительный | Подведение итогов и составление отчета: систематизация, анализ, обработка собранного в ходе практики материала, предоставление отчета, публичная защита отчета | 01.09.2022-14.09.2022 |  |

Руководитель от УрФУ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Свинцов Дмитрий Владимирович

**Подпись расшифровка подписи**

Задание принял к исполнению (студент) \_\_\_\_\_\_\_\_\_\_\_\_\_ Гайдабура Олег Андреевич

**Подпись расшифровка подписи**

# Форма отзыва руководителя

ОТЗЫВ

ФИО: Гайдабура Олег Андреевич

Вид практики: Учебная практика

Тип практики: Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности

Организация: УрФУ, ИРИТ-РТФ, Школа бакалавриата

Сроки практики: с 04.07.2022 г. по 17.07.2022 г.

Студент(ка) Гайдабура Олег Андреевич группы РИ-200001 института ИРИТ-РТФ за время прохождения практики осуществил(а) следующие мероприятия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

В период практики студент(ка)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(краткая характеристика уровня подготовки и отношения практиканта к работе)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка за практику \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«18» июля 2022 г.

Руководитель практики от УрФУ Д. В. Свинцов