Уральский федеральный университет

имени первого Президента России Б. Н. Ельцина

Кафедра «Интеллектуальные информационные технологии»

**ПРАКТИЧЕСКАЯ РАБОТА 3**

ОТЧЁТ

Дисциплина «Веб-программирование»

Проверил:

преподаватель

Свинцов Д. В.

Выполнил:

студент гр.

РИ-380022

Кучерявый Д.А.

Екатеринбург 2020

**Работа с протоколом WebSocket**

**Задание 1.**

Написать WebSocket сервер на языке Python с использованием библиотеки aiohttp. Сервер должен уметь отдавать файлы по http запросу:

* <http://127.0.0.1:8080/myfile1.txt>
* <http://127.0.0.1:8080/>my\_server.py
* <http://127.0.0.1:8080/README.rst>

Создадим файл с расширением py. Внутри этого файла напишем код реализации Websocket сервера.



Рисунок 1 Создаем Read-методы для открытия файла по ссылке

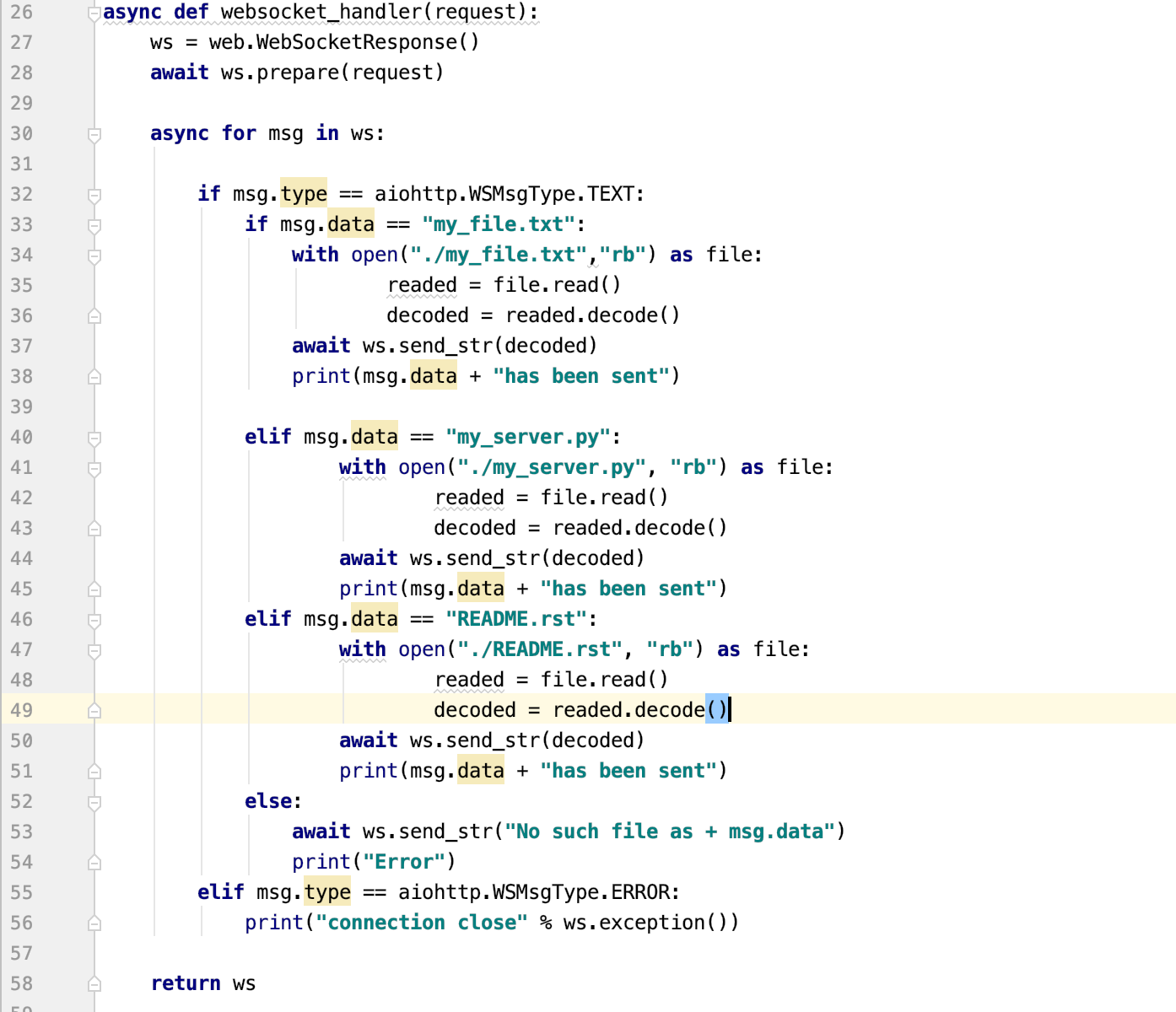


Рисунок 2 Создаем хэндлер

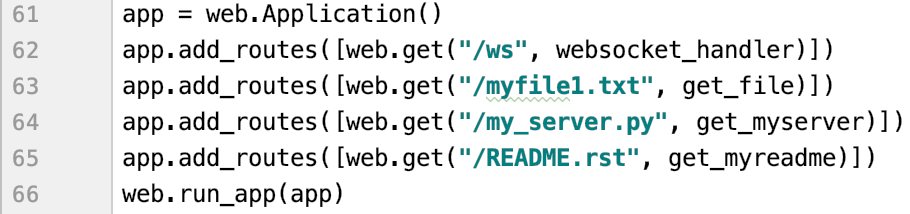


Рисунок 3 Запуск приложения

Теперь сохраним этот файл и запустим его. Попробуем выполнить запрос на получение файла с сервера.

В адресной строке введем предложенные запросы:

* <http://127.0.0.1:8080/myfile1.txt>
* <http://127.0.0.1:8080/>my\_server.py
* <http://127.0.0.1:8080/README.rst>

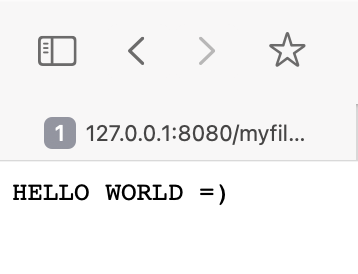


Рисунок 4 Вывод файла myfile1.txt

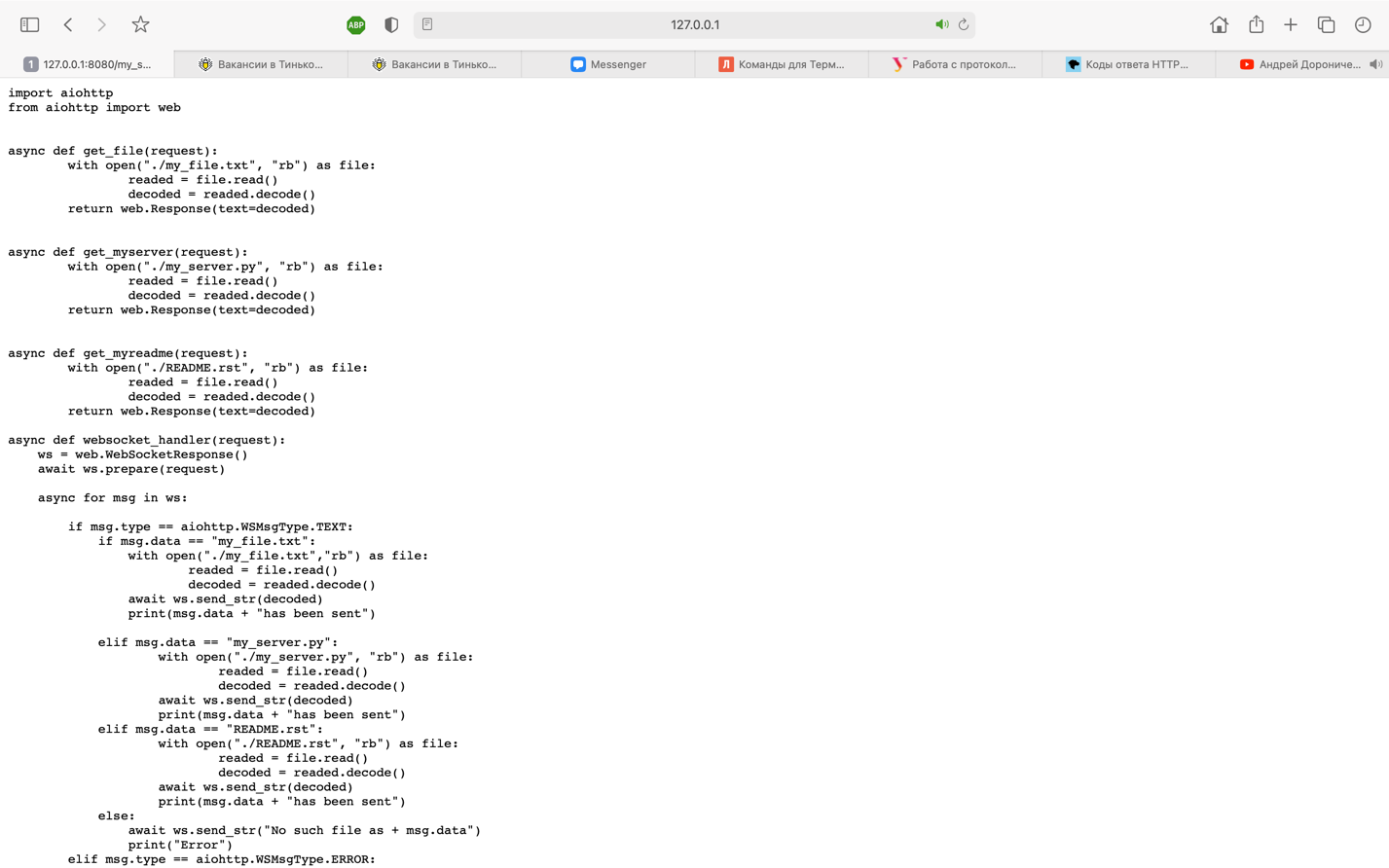


Рисунок 5 Вывод файла my\_server.py

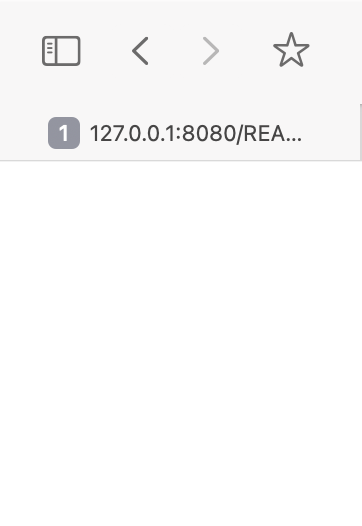


Рисунок 6 Вывод файла README. Он пустой.

**Задание 2.**

Создать проект со следующей структурой:

myproject/

index.html

Создадим файл index.html в папке myproject.

Внутри файла index.html поместим код на JavaScript, для получения содержимого файлов с сервера при помощи AJAX запросов через WebSocket.



Рисунок 7 Код index.html

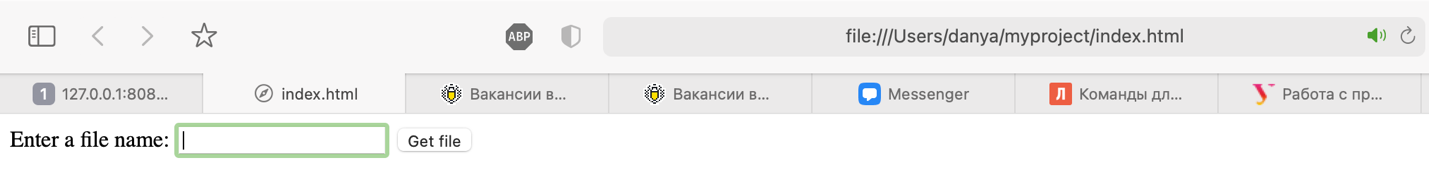


Рисунок 8 Результат в браузере

**Задание 3**

Добавить в index форму для отправки запросов на получения содержимого файлов.

Проверим работу скрипта, введем название файла:

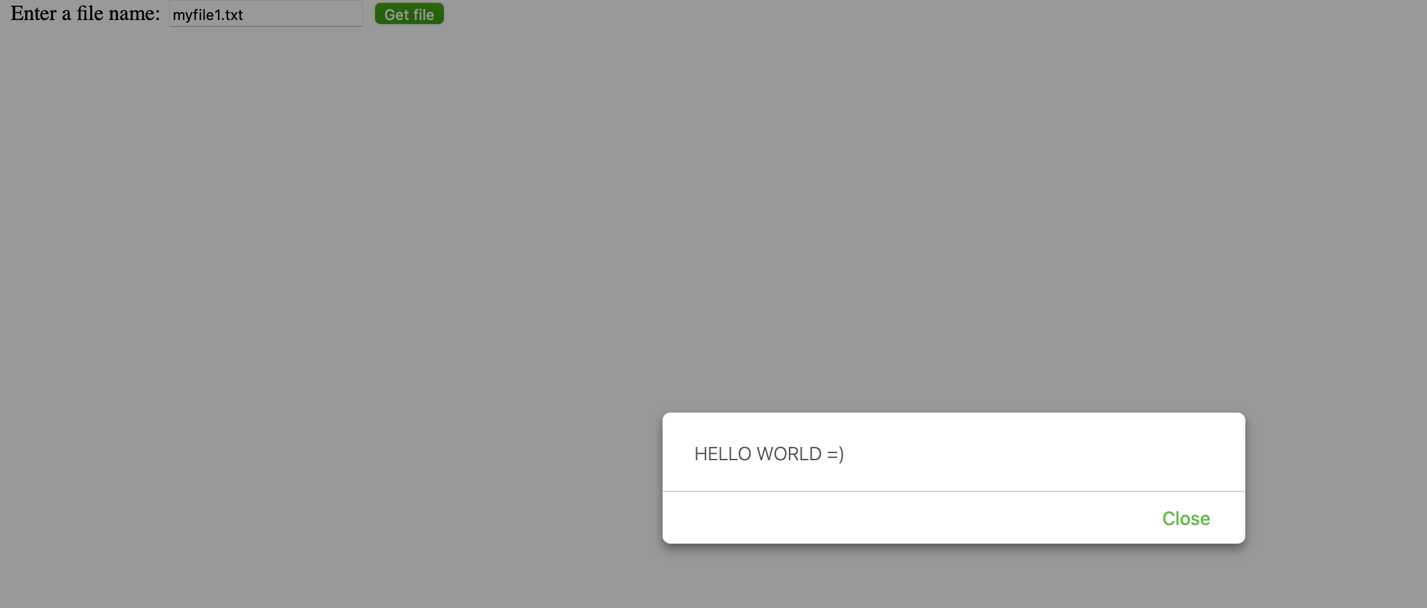


Рисунок 9 Результат выполнения