

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

ОТЧЕТ О ПРОХОЖДЕНИИ

|  |
| --- |
| производственной практики (проектно-технологической) |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ОБУЧАЮЩЕГОСЯ | 3 КУРСА | БАКАЛАВРИАТА | ГРУППЫ | ИДБ-21-06 |
|  |  |  |  |  |

|  |
| --- |
| **Музафарова Карима Ринатовича** |
|  |

|  |  |
| --- | --- |
| Кафедра: | информационных систем |
| Направление подготовки: | 09.03.02 «Информационные системы и технологии» |

|  |  |
| --- | --- |
| Место прохождения практики: | АО НТЦ «НЕПТУН» |
| сРОКИ прохождения ПРАКТИКИ: | 28.06.2024 – 25.07.2024 |

Руководитель практики:

|  |  |  |
| --- | --- | --- |
| ОТ КАФЕДРЫ | – | Иванова Татьяна Васильевна,  старший преподаватель |
|  |  |  |
| ОТ ОРГАНИЗАЦИИ | – | Зарубин Сергей Вадимович,  генеральный директор |

Оглавление

[ВВЕДЕНИЕ 3](#_Toc172295276)

[глава 1. Востребованность парсинга сайтов 4](#_Toc172295277)

[глава 2. Общее описание программы 5](#_Toc172295278)

[глава 3. Автоматическая регистрация 7](#_Toc172295279)

[глава 4. Получение необходимых данных 9](#_Toc172295280)

[глава 5. Сохранение полученных данных 12](#_Toc172295281)

[глава 6. Обеспечение равертывания приложения по методологии DEvops 14](#_Toc172295282)

[Заключение 17](#_Toc172295283)

[список литературы 18](#_Toc172295284)

ВВЕДЕНИЕ

Местом прохождения производственной практики является АО НТЦ «НЕПТУН».

Основные задачи на практику были получены от генерального директора Зарубина Сергея Вадимовича.

Компания «Акционерное общество Научно-Технический Центр "Нептун"» зарегистрирована 26.12.2019 г. по адресу г. Москва пер. Нижний Сусальный д. 5 стр. 18 помещ. I, комн. 11, этаж 3. Краткое наименование: Научно-Технический Центр Нептун. При регистрации организации присвоен ОГРН 1197746752609, ИНН 9709058173 и КПП 770901001. Юридический адрес: г. Москва пер. Нижний Сусальный д. 5 стр. 18 помещ. 1, комн. 11, этаж 3.

Основным видом деятельности является Разработка компьютерного программного обеспечения, зарегистрировано 19 дополнительных видов деятельности.

В рамках прохождения практики были поставлены задачи:

* Проанализировать сайт федеральной службы по аккредитации.
* Выделить поля в детальной информации по сертификатам, которые нужны организации.
* Спроектировать и написать программу, которая осуществляет сбор данных с выделенных полей.
* Проанализировать нужды сотрудников организации для нахождения наиболее удобного для них формата получения данных.
* Реализовать выдачу в данном формате.

глава 1. Востребованность парсинга сайтов

Парсинг (англ. parsing) — автоматизированный сбор и систематизация информации из открытых источников с помощью программного обеспечения. Альтернативное название этого процесса — вебскрейпинг (web scraping). ПО, которое собирает и систематизирует информацию, называется парсером.

За последнее десятилетие данные стали ресурсом для развития бизнеса, а сеть Интернет — их основным источником благодаря уже более чем пяти миллиардам пользователей [1]. Анализ данных Всемирной паутины помогает компаниям выявлять скрытые закономерности, позволяющие им добиваться выполнения своих целей. Парсинг веб-сайтов позволяет компаниям автоматизировать процессы сбора данных во Всемирной паутине. В контексте прохождения производственной практики был написан парсер, который собирает нужную для аналитиков данных информацию с сайта федеральной службы по аккредитации(https://pub.fsa.gov.ru) по определенным запросам и помещает в таблицу в формате xlsx. В данном отчете мы рассмотрим поэтапно написание парсера. Выделим следующие этапы парсинга сайтов:

1. Автоматическая регистрация на веб-портале.
2. Получение необходимых данных.
3. Сохранение полученных данных.

глава 2. Общее описание программы

В ходе практики было разработано программное обеспечение, которое по средствам HTTP запросов получает данные с сайта федеральной службы по аккредитации(https://pub.fsa.gov.ru).

Любое действие пользователя, работающего в веб-браузере, инициирует отправку пакетов HTTP-данных на удаленный сервер. Суть подхода заключается в том, что необходимо чтобы ПО отправляло такие же данные на сервер, какие бы отправил браузер. В этом случае ПО выполняет роль веб-браузера, а удаленный сервер не знает кто отправил эти данные, ведь данные идентичны.

Большинство современных языков программирования (C# [2], Java [3], Python [4]) имеют инструменты для отправки HTTP-пакетов и остается только узнать какие данные записать в отправляемый HTTP-пакет. Для этого существует множество инструментов таких как снифферы HTTP-пакетов (например: Fiddler [5], Charles Proxy [6]). Суть работы со сниффером – это запустить сниффер, в веб-браузере нажать кнопку авторизации, проверить в сниффере отправленный HTTP-пакет(см рис.1).

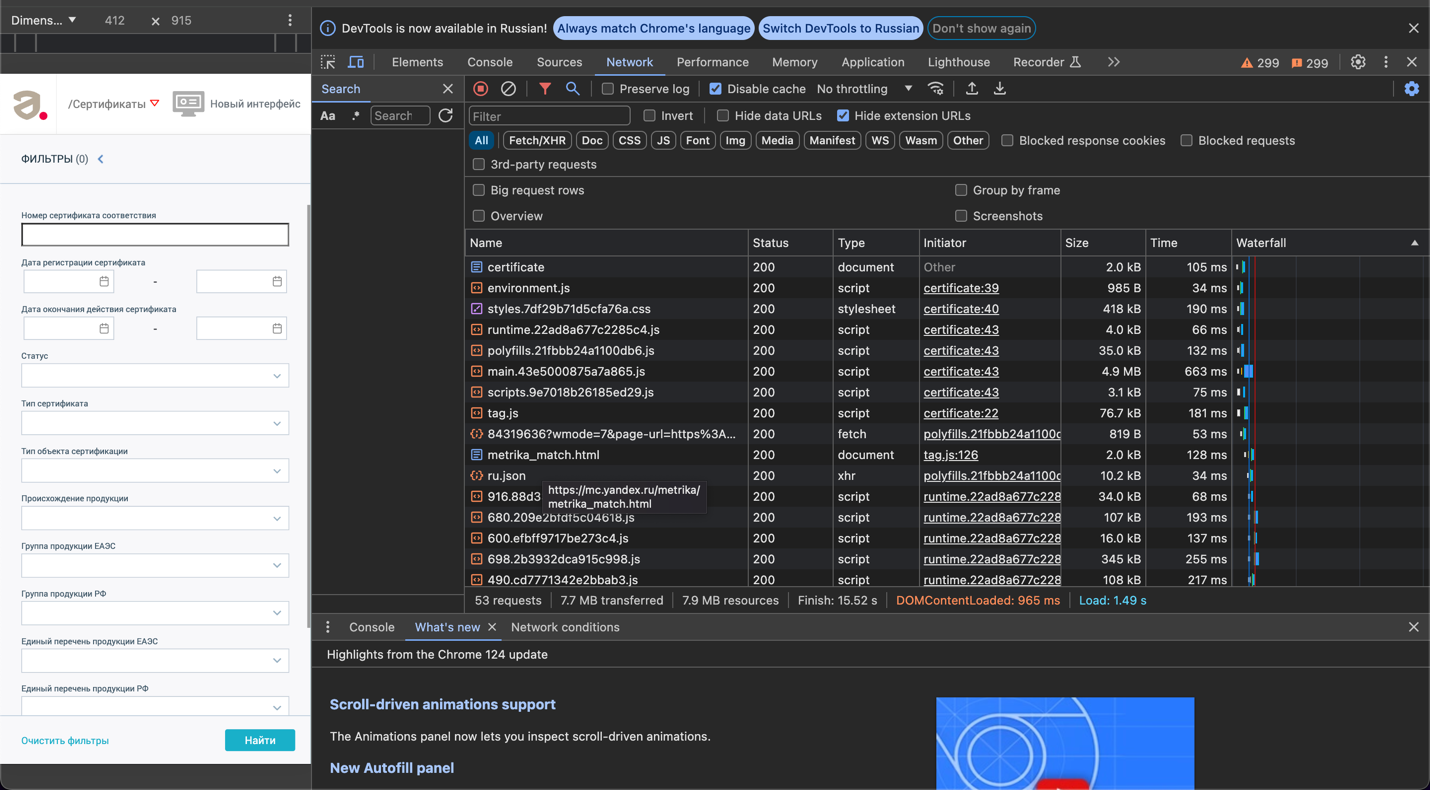


Рис.1. Получение всех HTTP пакетов веб браузером.

При парсинге сайтов в основном применяются данные полученные браузером по средствам XMLHttpRequest. XMLHttpRequest (XMLHTTP, XHR) — API, доступный в скриптовых языках браузеров, таких как JavaScript. Использует запросы HTTP или HTTPS напрямую к веб-серверу и загружает данные ответа сервера напрямую в вызывающий скрипт.[1] Информация может передаваться в любом текстовом формате, например, в XML, HTML или JSON. Позволяет осуществлять HTTP-запросы к серверу без перезагрузки страницы[7].

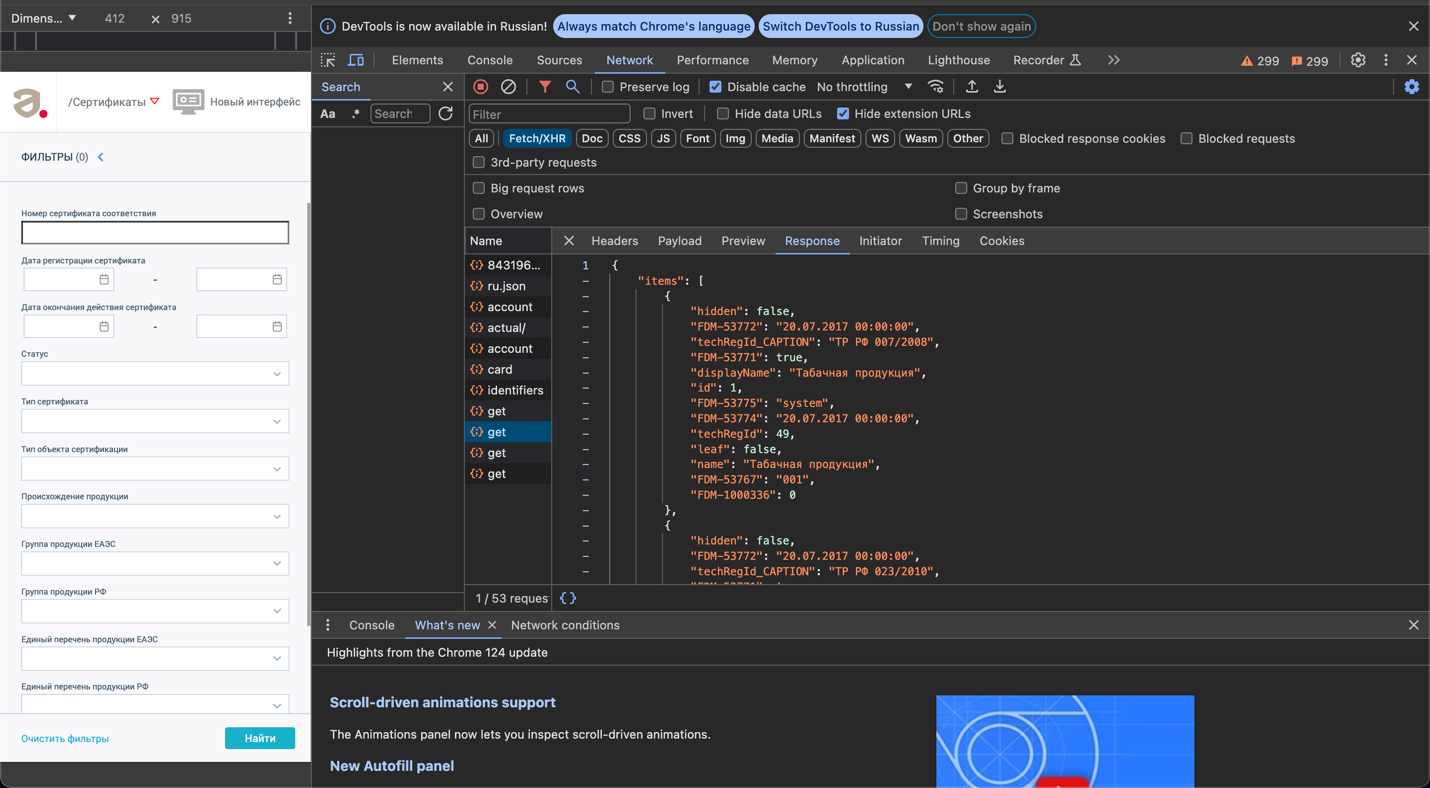


Рис.2. Вкладка с XMLHttpRequest.

глава 3. Автоматическая регистрация

«Автоматически» означает, что получением данных занимается не пользователь, а некоторое ПО. При обращении к веб-порталу (будь то пользователь или ПО) вы столкнетесь с проблемой выполнения авторизации на веб-портале. Все дело в том, что данные в сети Интернет бывают открытыми (авторизация не требуется) и закрытыми (требуется авторизация, обычно логин и пароль).

На сайте федеральной службы по аккредитации(https://pub.fsa.gov.ru) данные хранятся в открытом доступе, что делает парсинг данный с этого сайта законным, но при этом все равно проходит регистрация анонимного пользователя для создания сессии. Об этом свидетельствует заголовок авторизации внутри запросов на сайт(см рис.3).

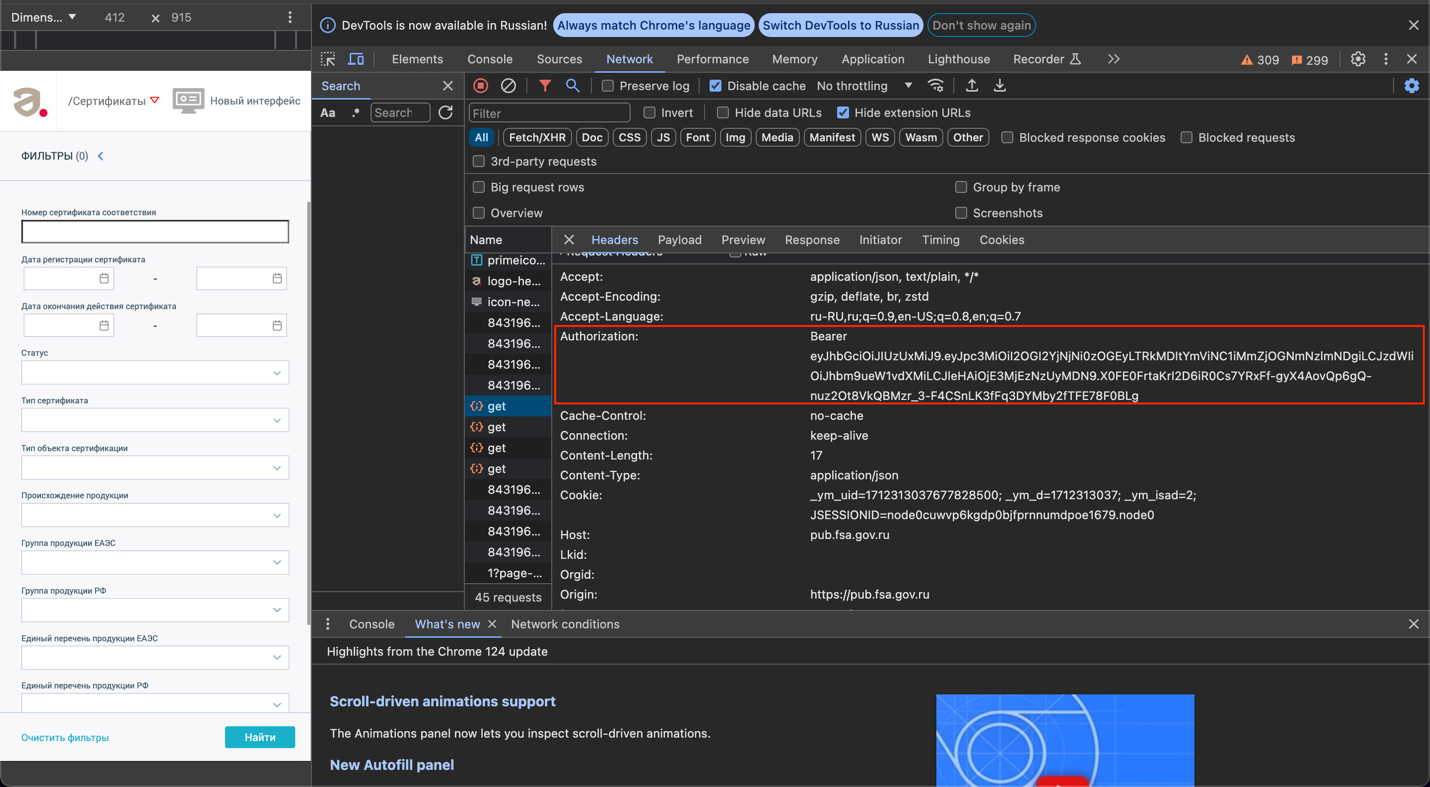


Рис.3. Заголовки запроса на сайт.

Проанализировав обмен данными между сервером и браузером в момент загрузки страницы был получен запрос, с помощью которого клиент(браузер) получает данный заголовок(см рис.4). Он приходит в заголовках ответа на post запрос «/login».



Рис.4. Получение заголовка авторизации.

Программа, при запросах на сайт запрашивает заголовок авторизации и добавляет к остальным заголовкам прописанным заранее (см рис.5).

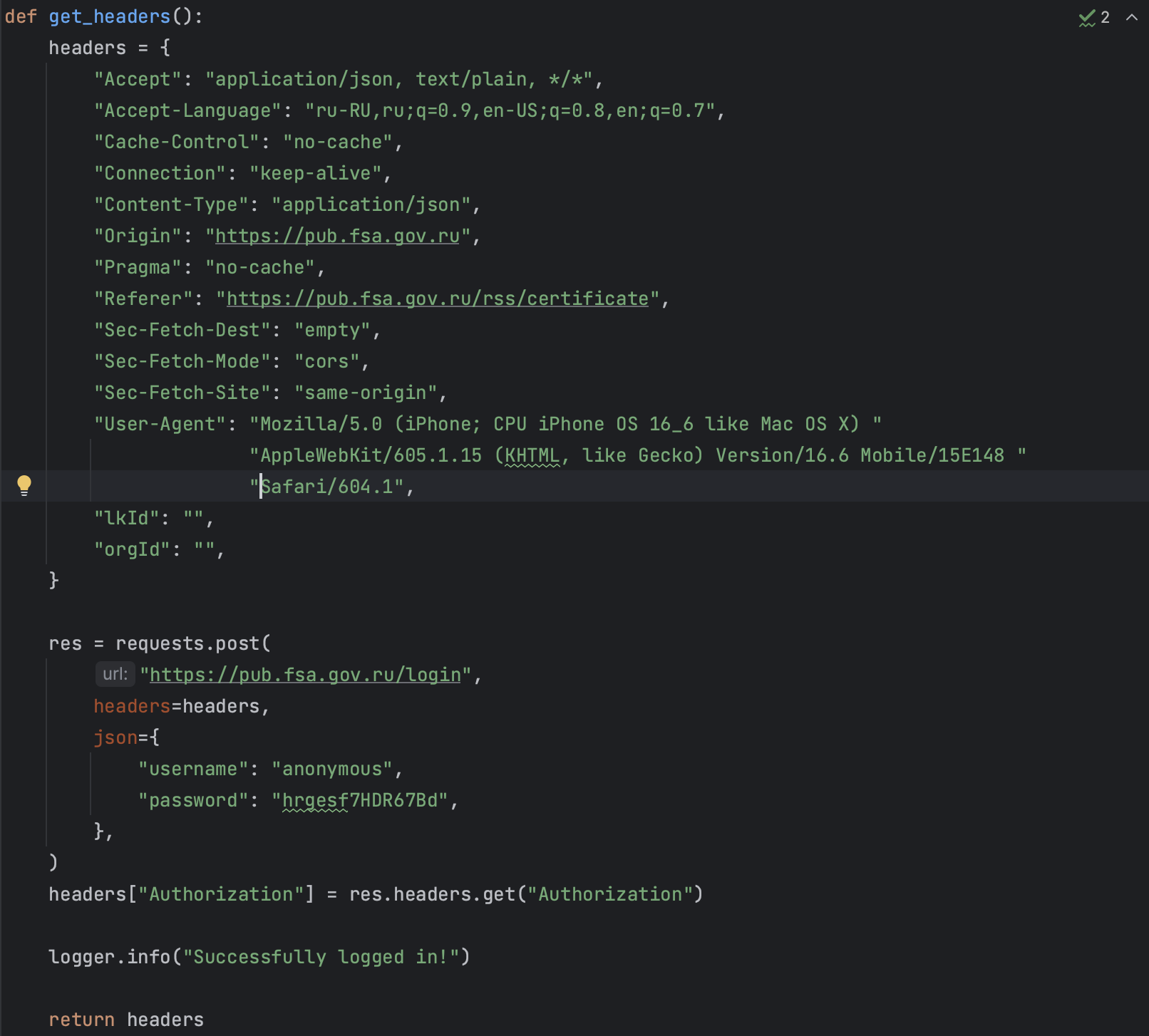


Рис.5. Получение заголовков для стандартного запроса.

глава 4. Получение необходимых данных

Получение данных идет при помощи XMLHttpRequest, в ответе с сайта федеральной службы по аккредитации(https://pub.fsa.gov.ru) приходит ответ в формате JSON

JSON (англ. JavaScript Object Notation)— текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми. Пример на (см рис.2).

При помощи прочтения данных ответов на запросы из браузера были найдены нужные аналитикам поля(см рис.6), и написан код для совершения совершения аналогичных запросов и получения данных из ответов.

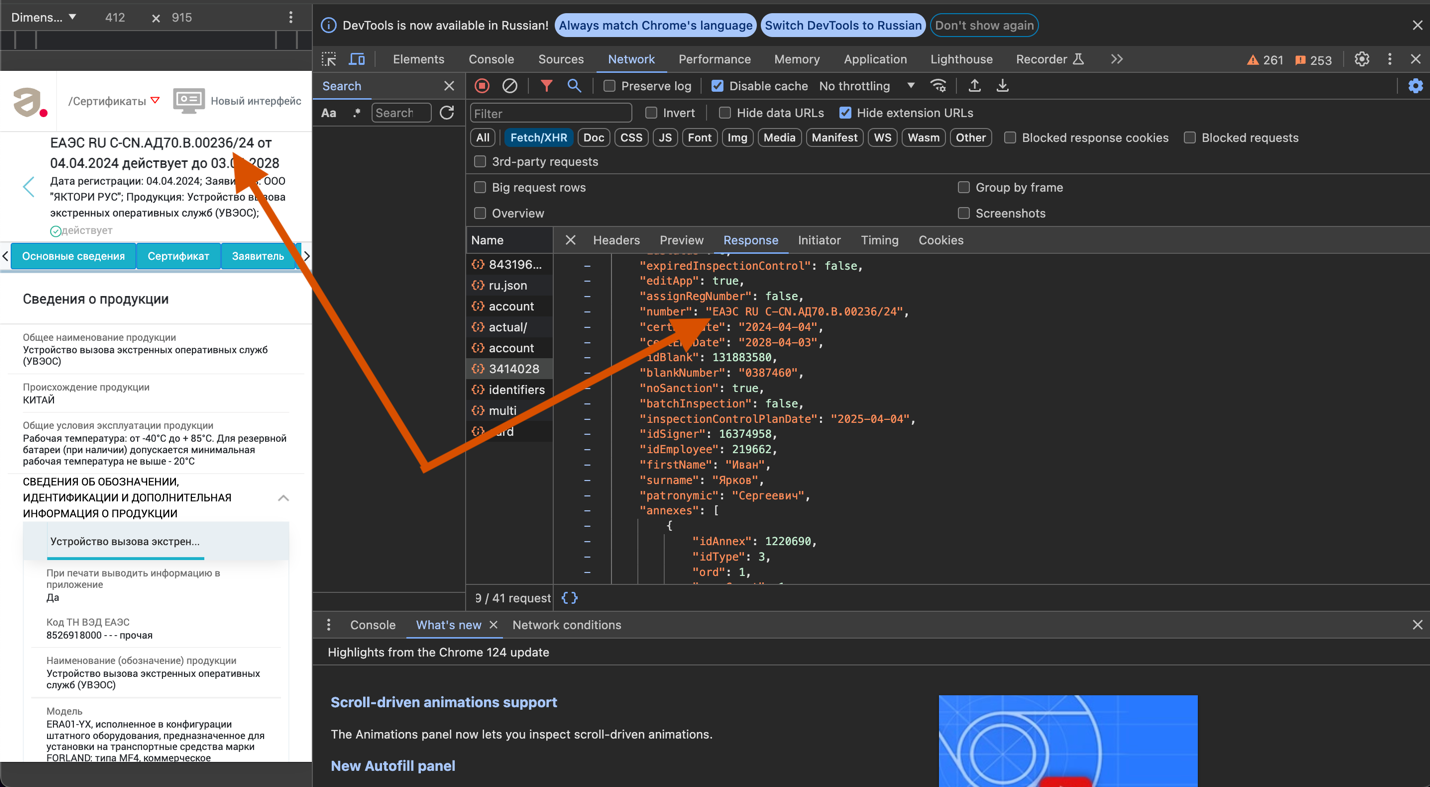


Рис.6. Общие поля на странице и в JSON файле.

Для отправки 5 видов запросов был написан специальный класс и 7 функций, распределенные по 4 файлам (см рис.8). В данном отчете будет рассмотрен запрос на получение сертификатов из поисковика по категориям(см рис 9-).

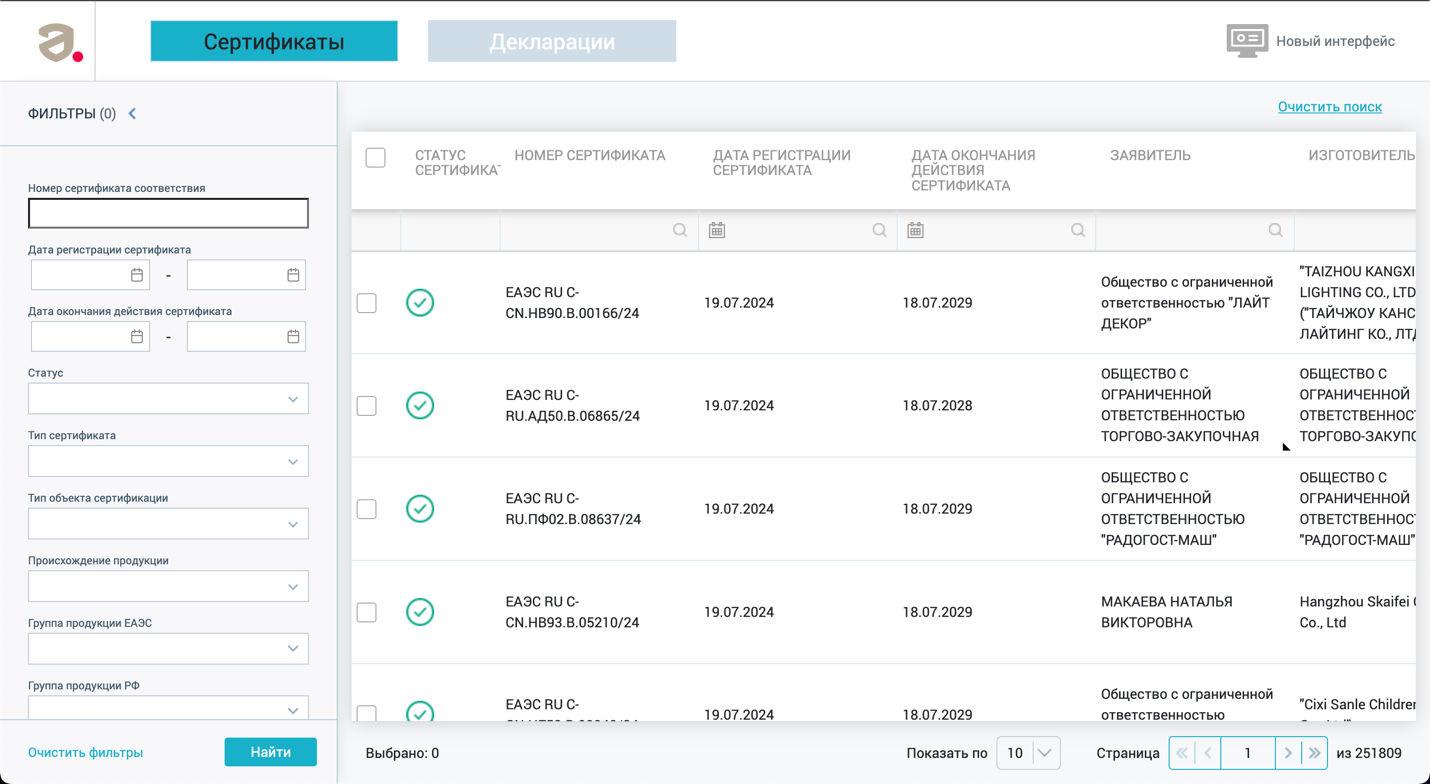


Рис.7. Результат выдачи по поисковому запросу.

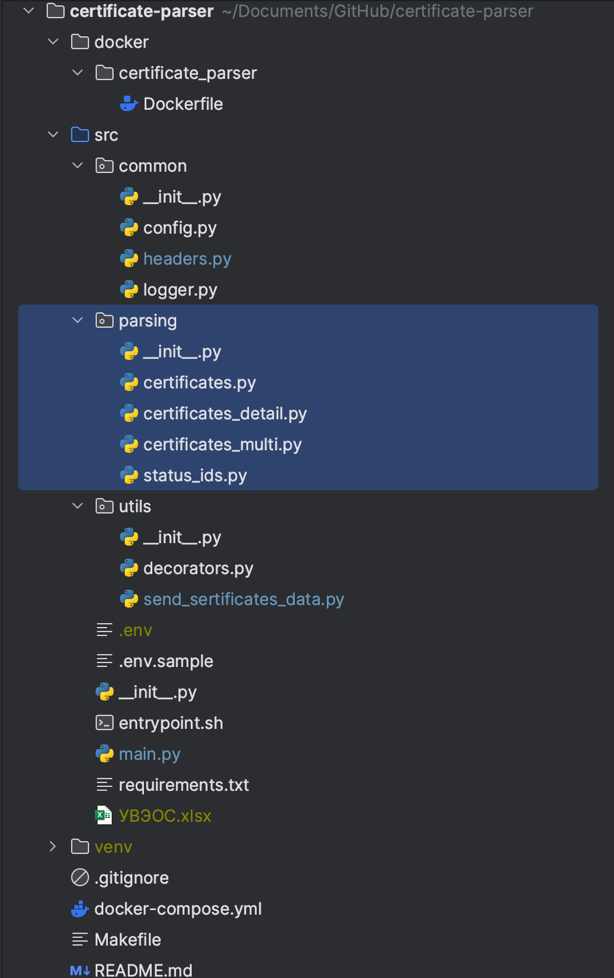


Рис.8. Иерархия файлов(выделены курсором файлы, отвечающие за получение данных).

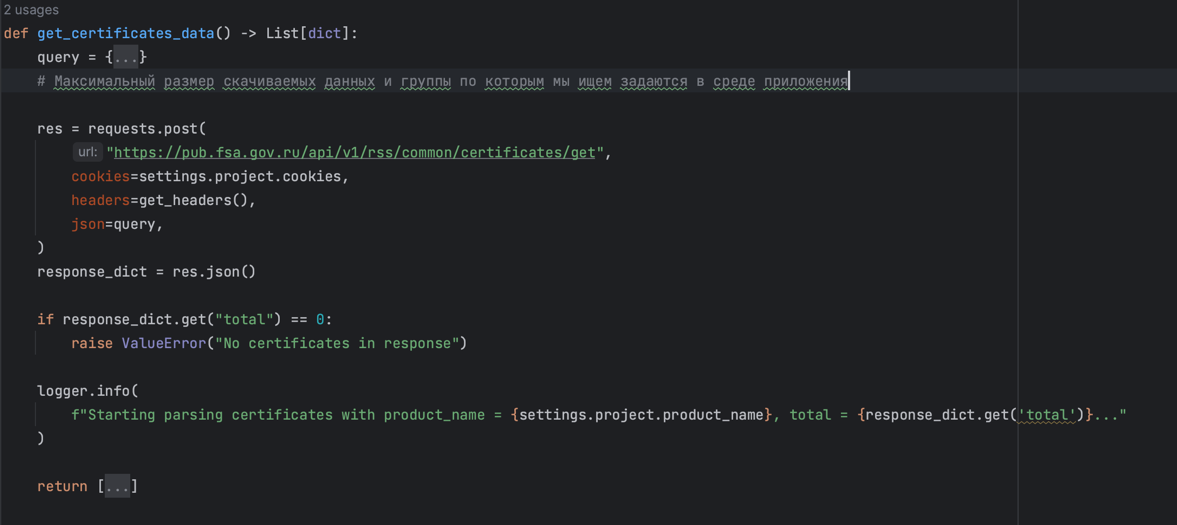


Рис.9. Код запроса в общем виде.

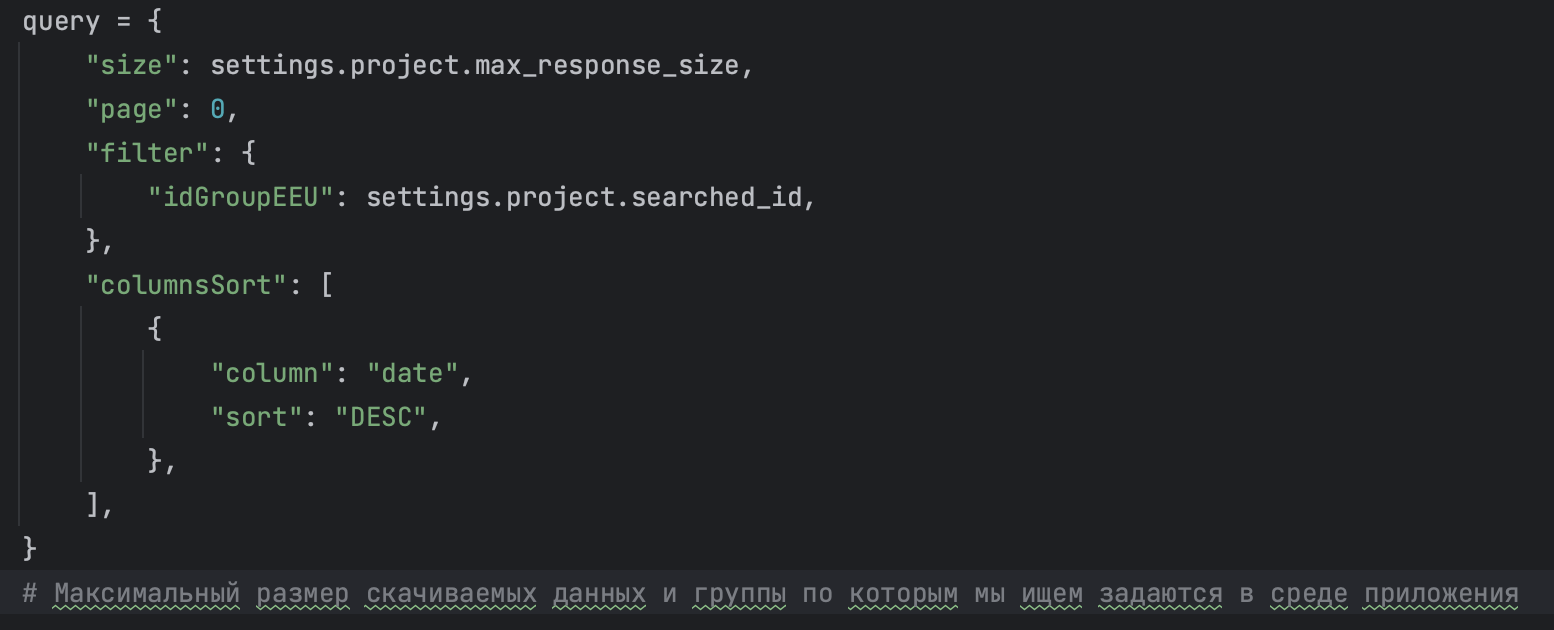


Рис.10. Детальное отображение параметров запроса, с комментариями.



Рис.11. Детальное отображение выведения ответа.

глава 5. Сохранение полученных данных

В результате согласования нужд с начальством была рассмотрена и принята инициатива сохранять полученную информацию в таблицу в удобном для аналитики и построения выборок формате .xlsx, и при изменениях данных на сайте федеральной службы по аккредитации(https://pub.fsa.gov.ru) и отправлять в бот в мессенджере Telegram, так как он бесплатный, быстрый, есть множество клиентов на разных платформах, не требует вложений. Работает даже при плохом канале. Обладает защитой данных, стабильностью работы, удобным, минималистичным интерфейсом, API для интеграций. Это позволит быстрее замечать изменения в выбранной отрасли и подстраивать под них стратегию развития компании.

Скрипт будет каждый день просматривать информацию на сайте федеральной службы по аккредитации(https://pub.fsa.gov.ru), и в случае изменения, аналитики компании АО НТЦ «НЕПТУН» узнают об изменениях в пределах суток.

На данный момент бот интегрирован в канал, в потенциале, если в связи ростом штата будет нужда в экономии ресурсов бота на отправку сообщений. Его можно добавить в группу, в которой будут сотрудники компании.

Аналитики компании, благодаря разработанному решению могут сэкономить свое время, в получаемом наборе данных более 100 строк, по категории, предложенной для тестирования и рассмотрены все данные, которые нужны для построения стратегии, в случае, если данных не будет доставать в столбце «B» представлены ссылки на записи (см рис. 12.).

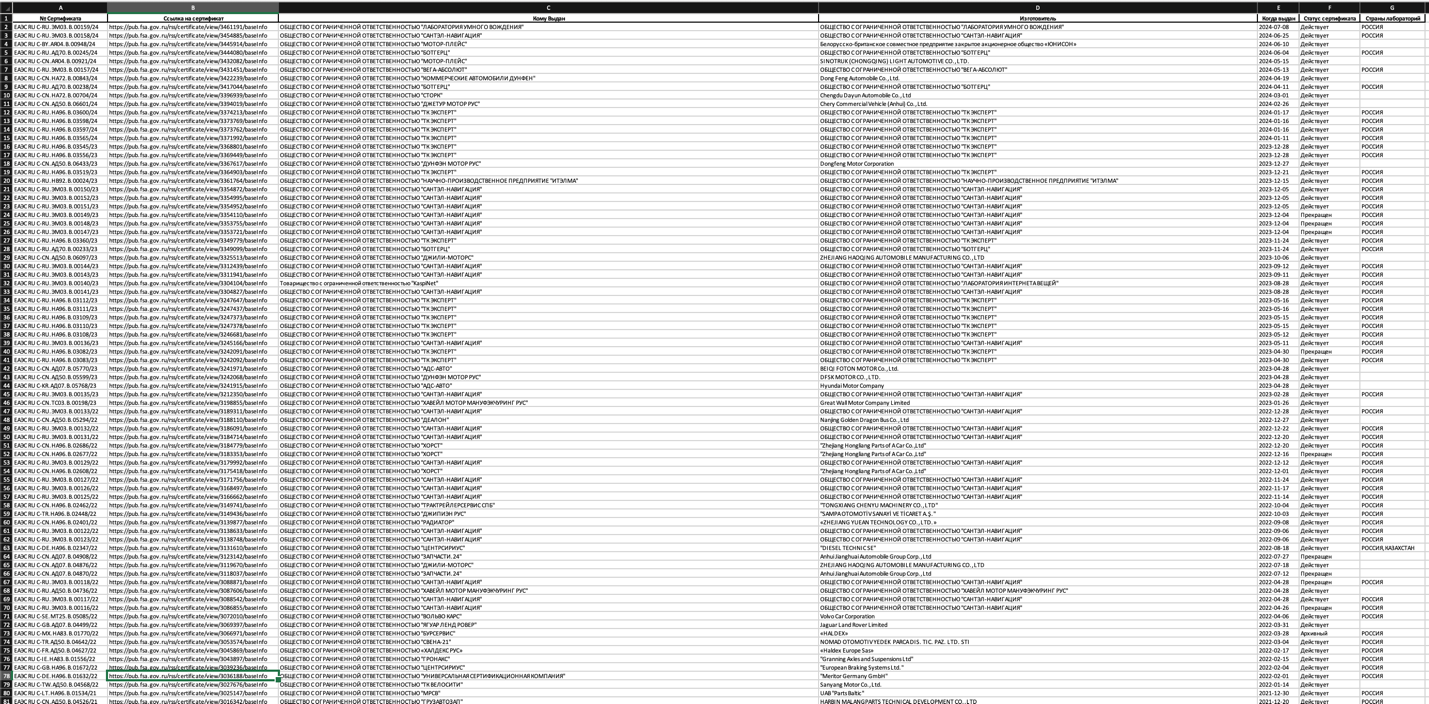


Рис.12. Файл в формате xlsx с полученными данными.

глава 6. Обеспечение равертывания приложения по методологии DEvops

Для развертывания приложения на сервере компании был использован Docker, благодаря контейнеризации можно бытть уверенным, что на корпоративном сервере работа будет так же стабильна как на устройстве, на котром приложение разрабатывалось.

Docker – это программная платформа для быстрой сборки, отладки и развертывания приложений. Docker упаковывает ПО в стандартизованные блоки, которые называются контейнерами. Каждый контейнер включает все необходимое для работы приложения: библиотеки, системные инструменты, код и среду исполнения. Благодаря Docker пользователи могут быстро развертывать и масштабировать свои приложения в любой среде и сохранять уверенность в том, что код будет работать. Контейнеры представляют собой легкие и портативные единицы, содержащие все необходимое для запуска приложения, включая код, среду выполнения и системные библиотеки. Контейнеризация позволяет запускать приложения в единообразной среде независимо от операционной системы и окружающей инфраструктуры, что облегчает миграцию и масштабирование приложений.

Основные преимущества данной технологии:

* Портативность: Контейнеры обеспечивают портативность приложений, что позволяет разработчикам легко передавать и запускать приложения на различных средах разработки, тестирования и развертывания.
* Портативность: Контейнеры обеспечивают портативность приложений, что позволяет разработчикам легко передавать и запускать приложения на различных средах разработки, тестирования и развертывания.
* Упрощение развертывания: Docker позволяет разработчикам упаковывать приложения и их зависимости в один контейнер, что значительно упрощает процесс развертывания приложений на серверах и в облаке.
* Масштабируемость: Docker поддерживает горизонтальное масштабирование, что позволяет легко запускать и управлять несколькими копиями контейнеров при увеличении нагрузки.

Для контейнеризации был написан следующий функционал:

1. «Dockerfile» отвечает за постановку зависимостей, распределение файлов и построение виртуального окружения.

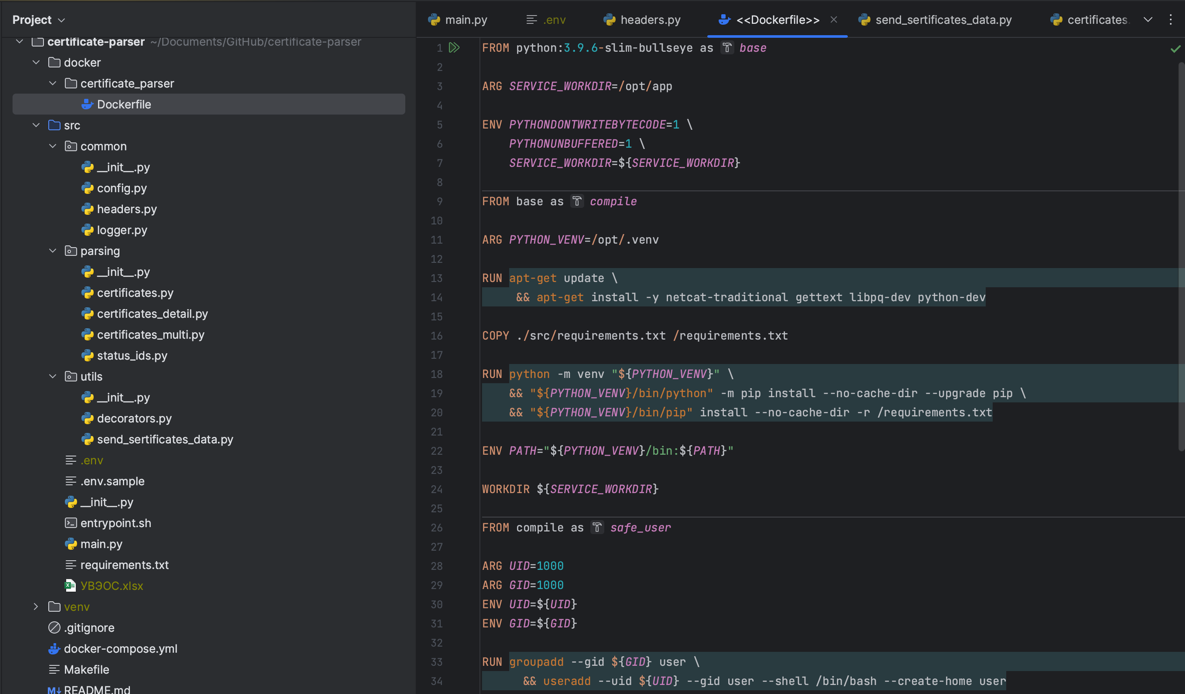


Рис.13. «Dockerfile» написанный для приложения.

1. «Entrypoint» отвечает за запуск приложения, задает команды, для работы с главным файлом.

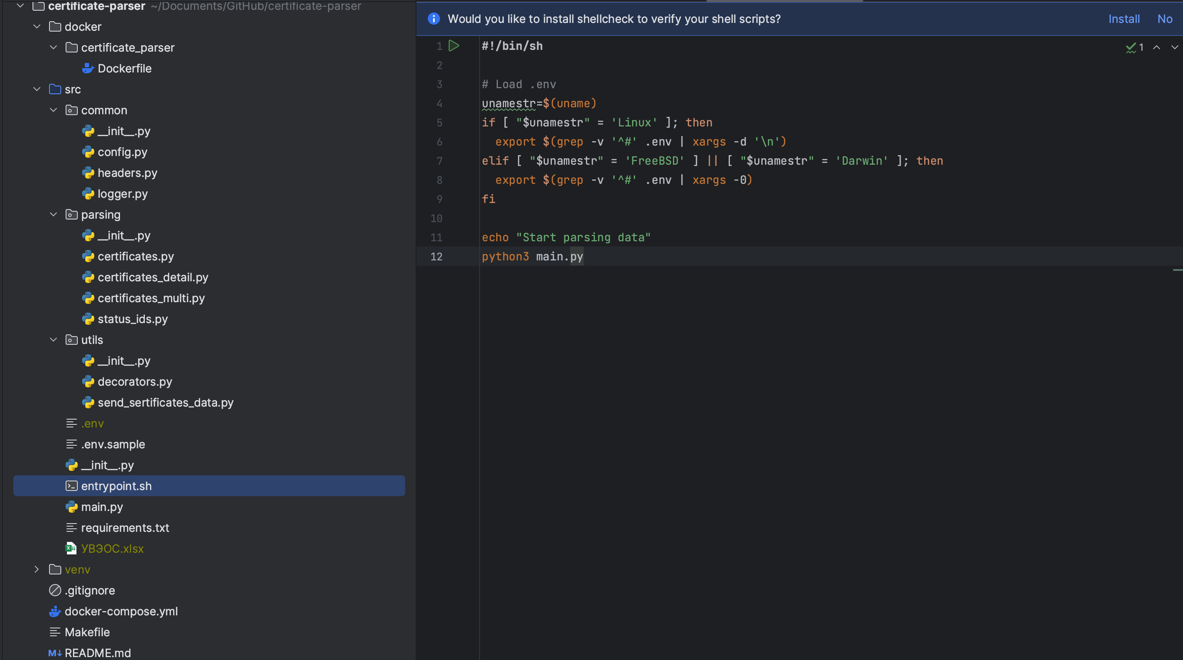


Рис.14. «Entrypoint» написанный для приложения.

1. «Docker Compose» отвечает за установку образов и построение контейнера.

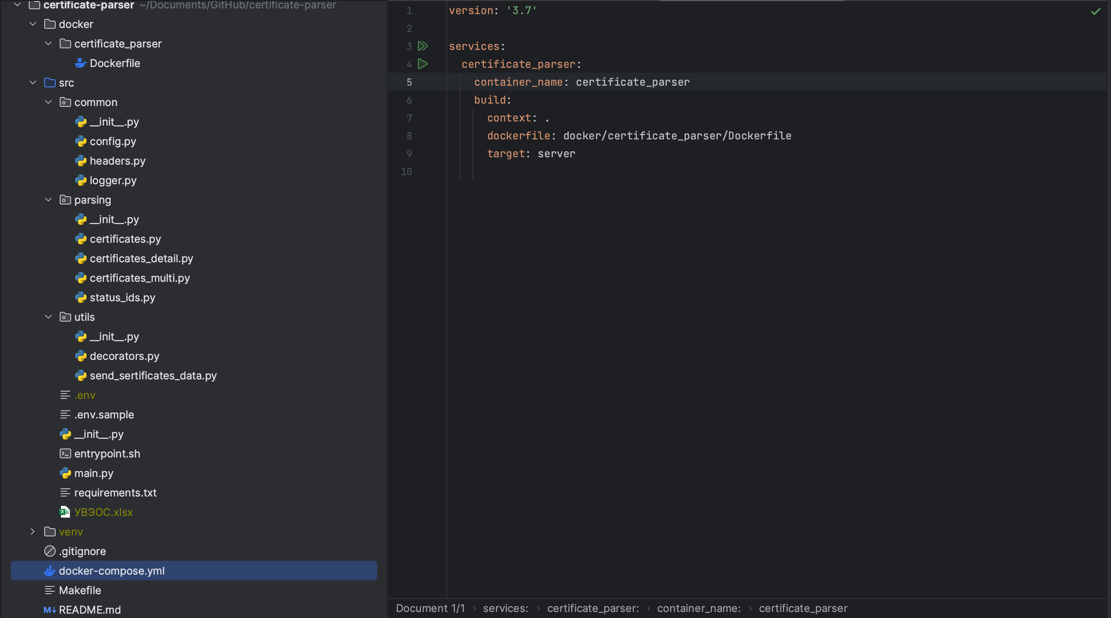


Рис.15. «Docker Compose» написанный для приложения.

1. «.env.sample» дает понять пользователю какие переменные виртуального окружения нужно будет выставить в «.env»

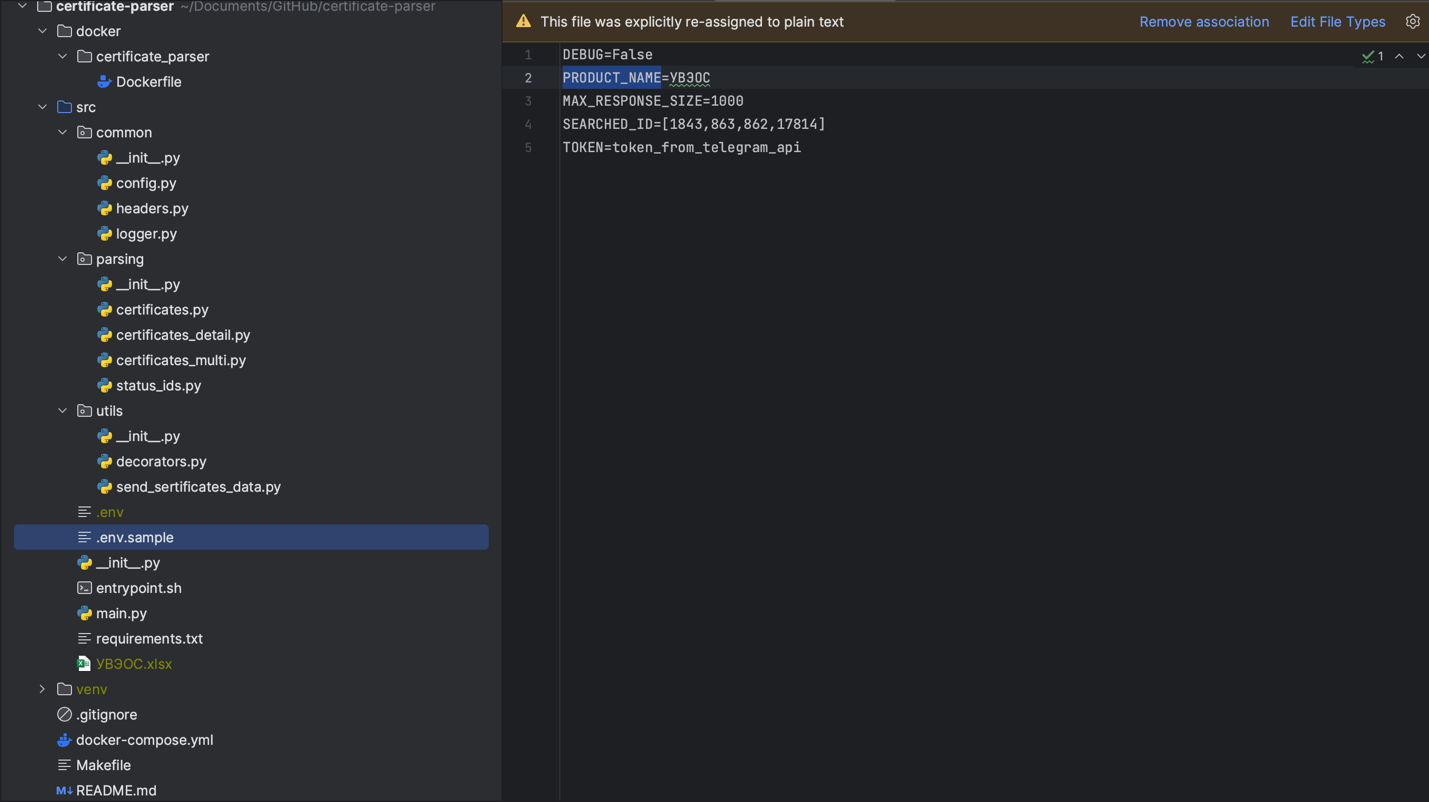


Рис.16. «.env.sample» написанный для приложения.

Заключение

В результате прохождения производственной практики в АО НТЦ «НЕПТУН» были успешно выполнены поставленные задачи, определенные генеральным директором Зарубиным Сергеем Вадимовичем. Был проведен анализ сайта федеральной службы по аккредитации, выделил необходимые поля в информации по сертификатам для организации, разработал программу для сбора данных по средствам HTTP запросов, и помещения результатов ее работы в таблицы формата xlsx. И реализованы оповещения в боте в Telegram. Так же был написан код для быстрого деплоя на сервере. И создана необходимая документация.

Результаты данной практики могут быть использованы для оптимизации процессов внутри организации. Была упрощена аналитика деятельности потенциальных поставщиков и конкурирующих фирм, что может принести более эффективной работе и развитию.

список литературы

1. DIGITAL 2022: JULY GLOBAL STATSHOT REPORT. Режим доступа: URL: https://datareportal.com/reports/digital-2022-july-global-statshot (дата обращения: 12.07.2024).
2. Классы WebRequest и WebResponse. Режим доступа: URL: https://metanit.com/sharp/net/2.2.php (дата обращения: 15.07.2024).
3. Class HttpsURLConnection. Режим доступа: URL: https://docs.oracle.com/javase/1.5.0/docs/api/javax/net/ssl/HttpsURLConnection.html (дата обращения: 15.07.2024).
4. httplib — HTTP protocol client. Режим доступа: URL: https://docs.python.org/2.6/library/httplib.html (дата обращения: 15.07.2024).
5. Fiddler Classic. Режим доступа: URL: https://www.telerik.com/fiddler/fiddler-classic (дата обращения: 15.07.2024).
6. Charles. Режим доступа: URL: https://www.charlesproxy.com/ (дата обращения: 15.07.2024).
7. XMLHttpRequest. Режим доступа: URL: https://www.w3.org/TR/XMLHttpRequest/ (дата обращения: 15.07.2024).

|  |  |  |  |
| --- | --- | --- | --- |
| Практикант | Музафаров Карим Риантович |  |  |
|  | *(ФИО полностью)* |  | *(подпись)* |

Отчет о прохождении практики сдан «19» \_\_\_\_Июля\_\_\_\_\_\_\_\_\_\_\_\_2024г.

|  |  |
| --- | --- |
| Оценка | 54 |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель практики | Зарубин Сергей Вадимович,  генеральный директор |  |  |
|  | *(ФИО полностью, должность)* |  | *(подпись)* |