**Федеральное государственное бюджетное образовательное учреждение высшего образования   
«Саратовский государственный технический университет имени Ю. А. Гагарина»**

**Кафедра прикладных информационных технологий**

**Отчёт по практической работе «Инструментальные средства документирования и построения отчётности».**

Выполнил студент б1-ИФСТ-31,

Яновский Евгений Валерьевич

Проверил

Старший преподаватель

кафедры «ПИТ»

Пиминов Дмитрий Алексеевич

Саратов, 2023

# Вариант 2: Инструментальные средства документирования API

**Анализ существующих инструментальных средств документирования API.**

1. **Mashape:**

* Mashape был разработан с целью предоставить инфраструктуру и инструменты для управления API, что является важным в современной разработке программного обеспечения и веб-приложений. Вот несколько основных целей и преимуществ использования Mashape.
* Управление API: Mashape обеспечивает удобный и централизованный способ управления всеми вашими API. С его помощью можно легко создавать, публиковать, отслеживать и обновлять API. Mashape предоставляет серверное программное обеспечение, которое позволяет управлять различными аспектами API, включая авторизацию, аутентификацию, маршрутизацию (Рисунок 1).
* Монетизация API: Mashape дает разработчикам возможность монетизировать свои API, позволяя продавать доступ к ним через Marketplace. Разработчики могут определить тарифные планы, установить цены и предложить различные модели оплаты (единовременная покупка, подписка, платежи за использование и т.д.), чтобы заработать на своих API.
* Аналитика и отчетность: Mashape предлагает инструменты аналитики и отчетности для API. С их помощью разработчики могут отслеживать и анализировать данные о использовании API, такие как количество запросов, время ответа, ошибки и другие метрики производительности. Это позволяет разработчикам понимать, как их API используется, и принимать меры для улучшения его работы.
* Маркетплейс API: Mashape предоставляет Marketplace, где разработчики могут публиковать и продавать свои API. Это позволяет пользователям находить и использовать API, которые соответствуют их требованиям, и расширять функциональность своих приложений с помощью сторонних API.
* Кросс-платформенность: Mashape поддерживает различные языки программирования и платформы, что делает его доступным для широкого круга разработчиков и команд разработки.
* Mashape предоставляет удобные и мощные инструменты для управления, монетизации и анализа API, что позволяет разработчикам создавать, развертывать и масштабировать API-инфраструктуру эффективным образом.

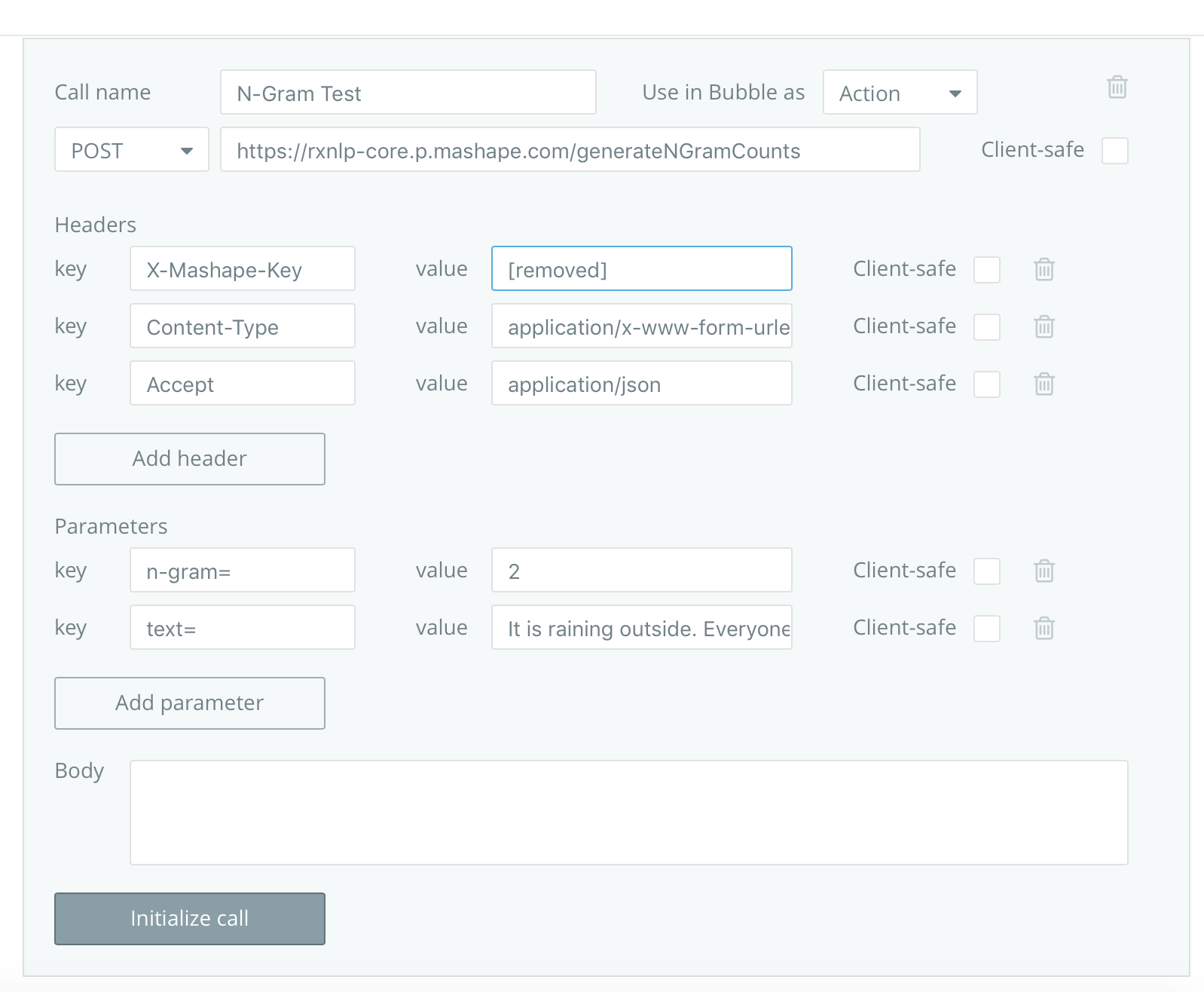
****

Рисунок 1. Интерфейс Mashape

1. **Swagger**

* Swagger является инструментальным средством документирования и описания API. Он предоставляет структурированный и понятный способ описания функций, конечных точек, параметров и ответов API. Вот несколько причин, почему Swagger полезен при разработке и документировании API (Рисунок 2).
* Документация: Swagger позволяет разработчикам создавать подробную и организованную документацию API. Описывая каждую конечную точку, параметры запроса, типы данных и возможные ответы, Swagger помогает другим разработчикам быстро понять, как использовать API и какие данные ожидать.
* Автоматическая генерация кода: Swagger документация может быть использована для автоматической генерации клиентского кода на различных языках программирования. Это упрощает процесс использования API для разработчиков, так как они могут сгенерировать основной код для работы с API на своем предпочитаемом языке.
* Тестирование: Swagger позволяет разработчикам тестировать API напрямую из его документации. Он предоставляет возможность отправлять запросы на конечные точки API и проверять полученные ответы. Это удобно для проверки функциональности и проверки правильности работы API.
* Удобство в использовании: Swagger предлагает легкий в использовании веб-интерфейс для работы с документацией API. Разработчики могут просматривать схемы запросов и ответов, выполнять запросы для тестирования и легко найти нужную информацию.
* Совместная работа: Swagger облегчает взаимодействие между разработчиками, клиентами и другими участниками проекта. Он предоставляет единое и понятное описание API, которое может быть использовано командой разработчиков или передано сторонним разработчикам, которые хотят интегрировать API в свои приложения.
* Swagger является стандартом в индустрии для описания RESTful API и предоставляет много полезных функций, которые упрощают разработку, тестирование и документирование API.

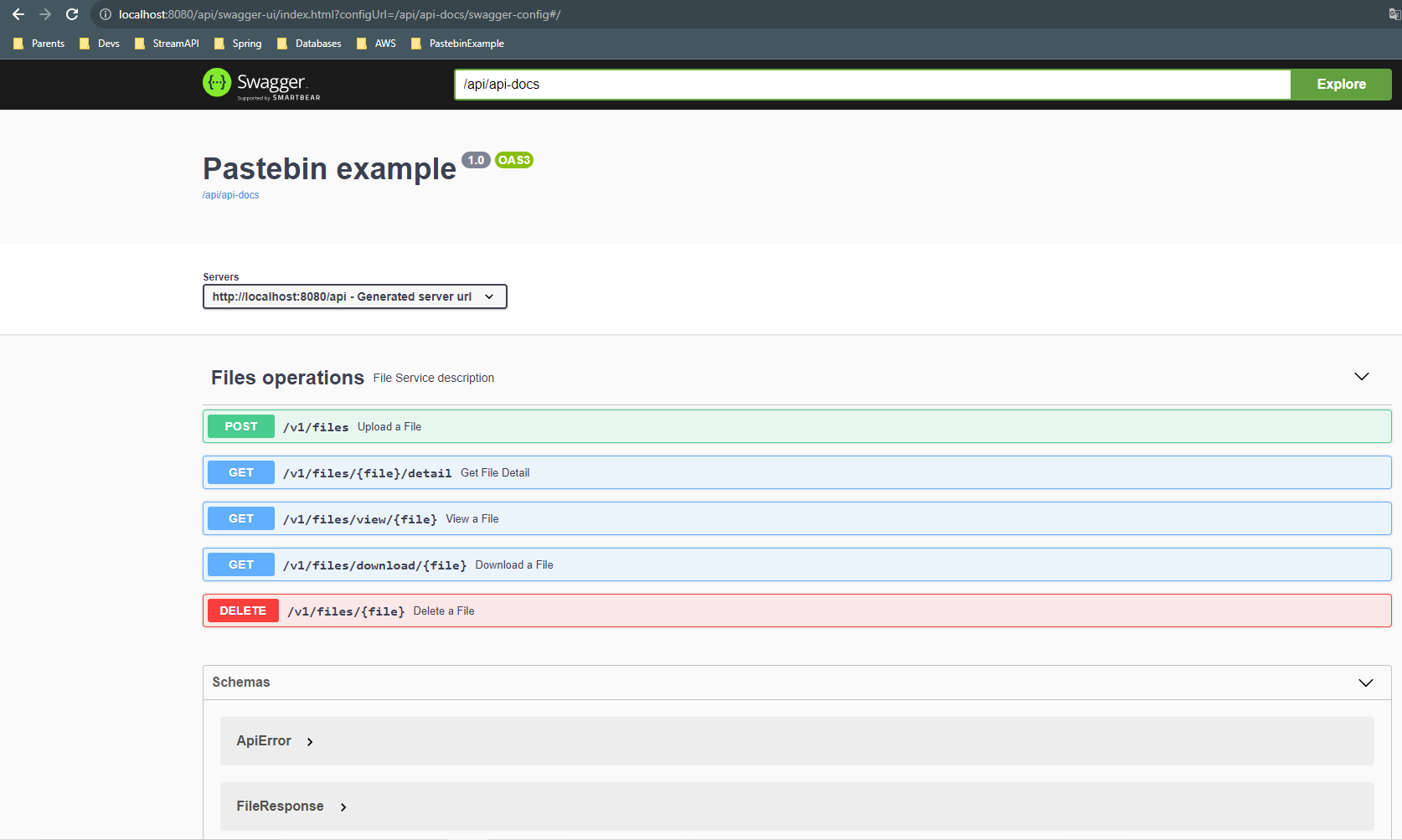


Рисунок 2. Интерфейс Swagger

1. **Apiary**

* Apiary - это онлайн-платформа для документирования и разработки API. Он предоставляет инструменты, которые помогают разработчикам создавать, описывать и тестировать свои API. Вот несколько причин, почему Apiary полезен при разработке и документировании API:
* Документация: Apiary предлагает интуитивный и понятный способ создания подробной документации для вашего API. С помощью языка разметки API Blueprint (или RAML) вы можете легко описать эндпоинты API, параметры запросов, ожидаемые ответы и другую информацию. Результатом является читаемая и структурированная документация, которая помогает другим разработчикам понять ваше API.
* Тестирование: Apiary предоставляет средства для проведения тестирования вашего API. Вы можете отправлять запросы к своему API непосредственно из платформы Apiary и проверять ожидаемые ответы. Это удобно для проверки функциональности и правильности работы вашего API.
* Совместная работа: Apiary облегчает совместную работу между разработчиками, командой поддержки и клиентами. Вы можете делиться своей документацией API с другими и приглашать их просматривать и комментировать ее. Это улучшает коммуникацию и помогает вам получить обратную связь и предложения по улучшению вашего API.
* Генерация клиентского кода: Apiary позволяет автоматически генерировать клиентский код на различных языках программирования на основе описания вашего API. Это упрощает использование вашего API для разработчиков, так как они могут сгенерировать основной код для взаимодействия с вашим API.
* Интеграция с другими инструментами: Apiary интегрируется с другими инструментами разработки, такими как Git, чтобы обеспечить контроль версий вашей документации и синхронизацию с вашими репозиториями кода.
* Apiary является полезным инструментом для разработчиков, предоставляя средства для эффективного документирования, тестирования и совместной работы над вашим API.

1. **Вывод:**

В ходе проведённого анализа было выбрано инструментальное средство документирования API – Swagger. Swagger является одним из наиболее популярных инструментов для документирования API и имеет широкую базу пользователей и активную сообщество. Это означает, что есть множество ресурсов, учебных материалов и поддержки, доступных для разработчиков, использующих Swagger. Swagger предлагает простой и интуитивно понятный способ описания API с использованием YAML или JSON. С его помощью разработчики могут быстро создавать документацию для своего API, описывая конечные точки, параметры и ожидаемые ответы. Swagger также предлагает визуальный интерфейс, который упрощает работу с документацией и тестированием API. Swagger имеет расширяемую архитектуру, которая позволяет добавлять дополнительные функциональные возможности и интеграции по мере необходимости.

**Практическая апробация инструментального средства документирования API.**

1. Осуществить и обосновать выбор инструментального средства. Swagger является одним из наиболее популярных инструментов для документирования API и имеет широкую базу пользователей и активную сообщество. Это означает, что есть множество ресурсов, учебных материалов и поддержки, доступных для разработчиков, использующих Swagger. В совокупности эти факты склоняют разработчиков отдавать предпочтение данному инструментальному средству документирования API (Рисунок 3).

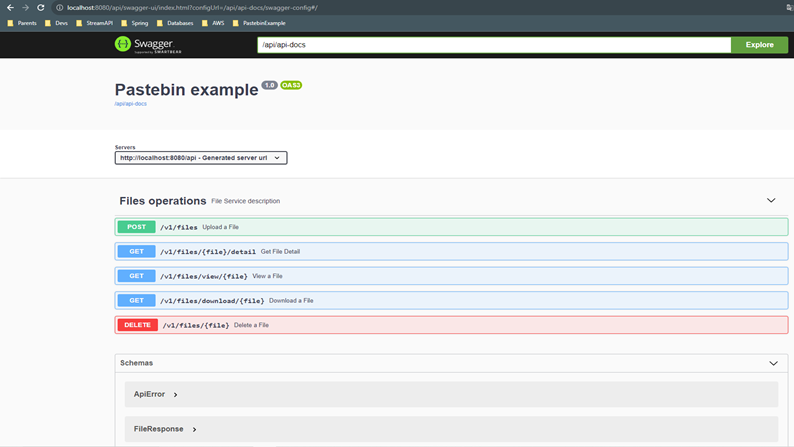


Рисунок 3. Интерфейс Swagger

1. Реализовать простое API, включающее возможности получения, обработки и оправки данных. В ходе выполнения практической работы было реализовано API для взаимодействия с облачным хранилищем MinIO, которое является аналогом сервиса облачного хранилища Amazon S3. Поскольку задача заключается в операциях над файлами – создадим класс FileResponse (Рисунок 4).

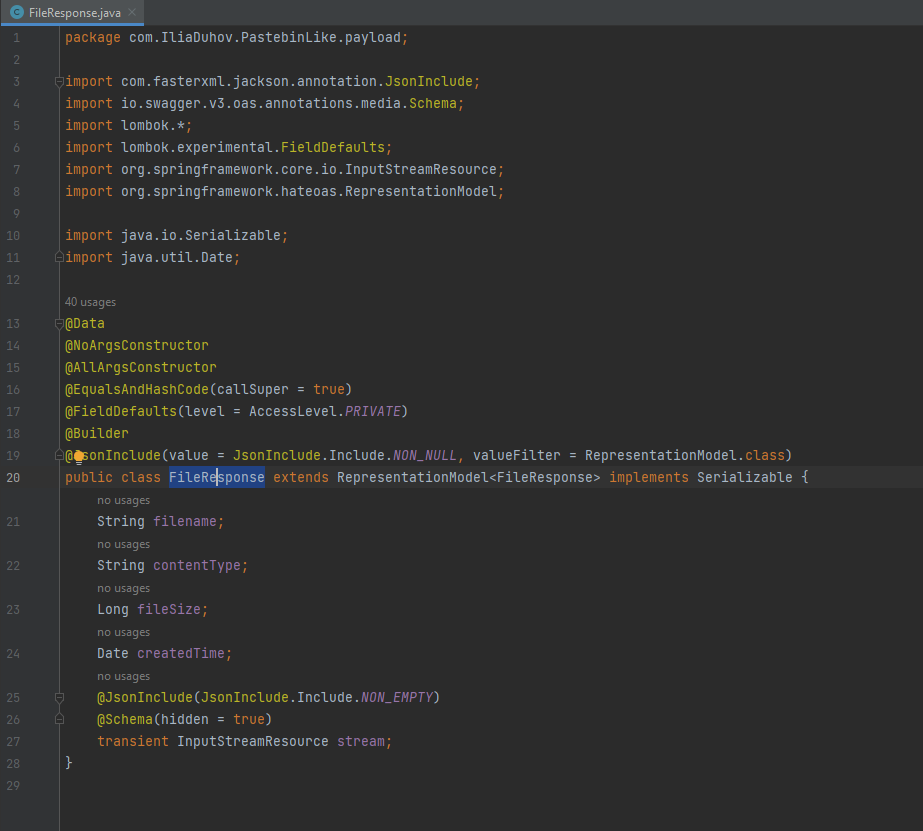


Рисунок 4. Класс FileResponse

Так же будет создан сервис для реализации операций над файлами, доступ к которому будет по интерфейсной ссылке (Рисунок 5).

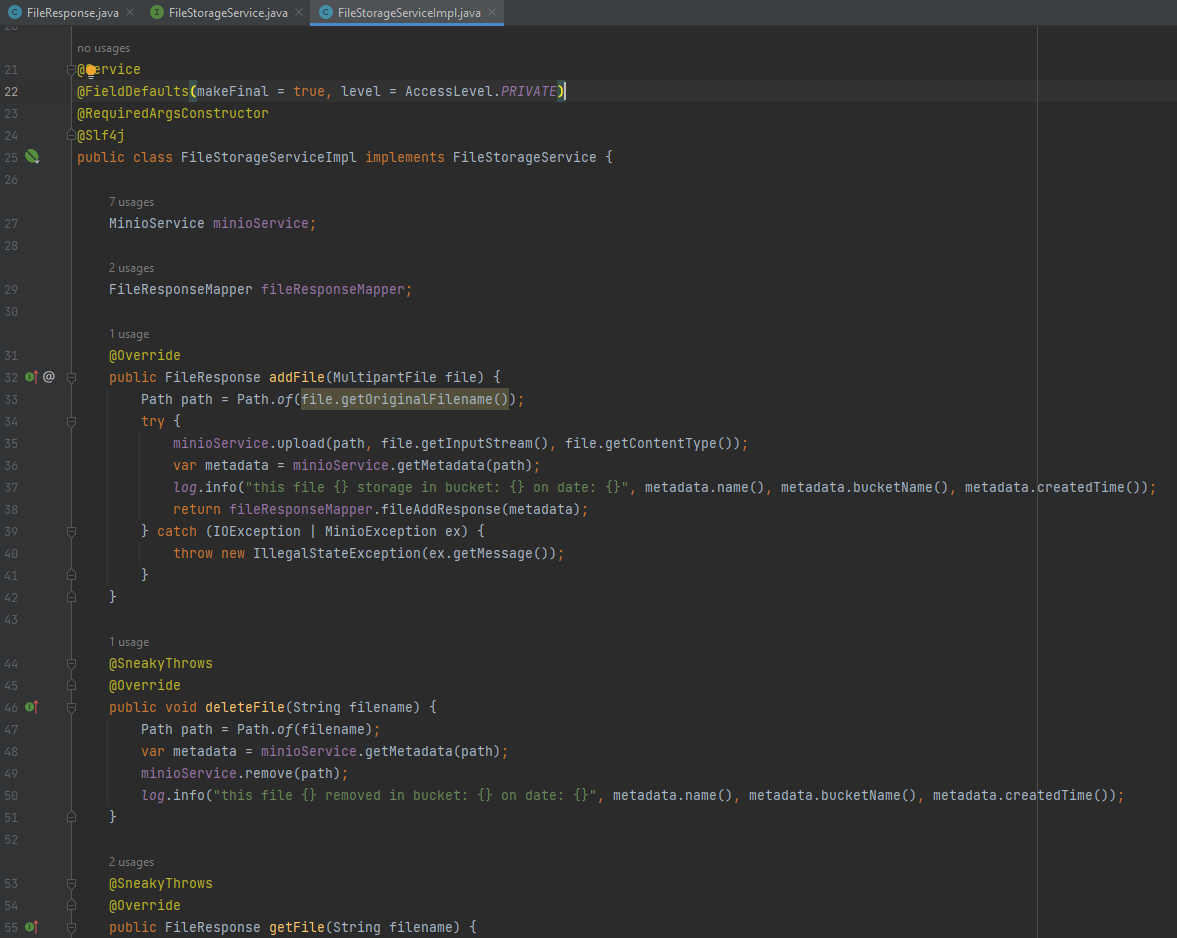


Рисунок 5. Сервис для реализации операций

Далее следует создать котроллер (Рисунок 6) для реализации логики операций над файлами. Благодаря таким аннотациям как - @ResponseStatus, @Controller, выбранный инструмент документирования API выделит поля и методы и создаст схемы, которые дадут дополнительную информацию об ограничениях.

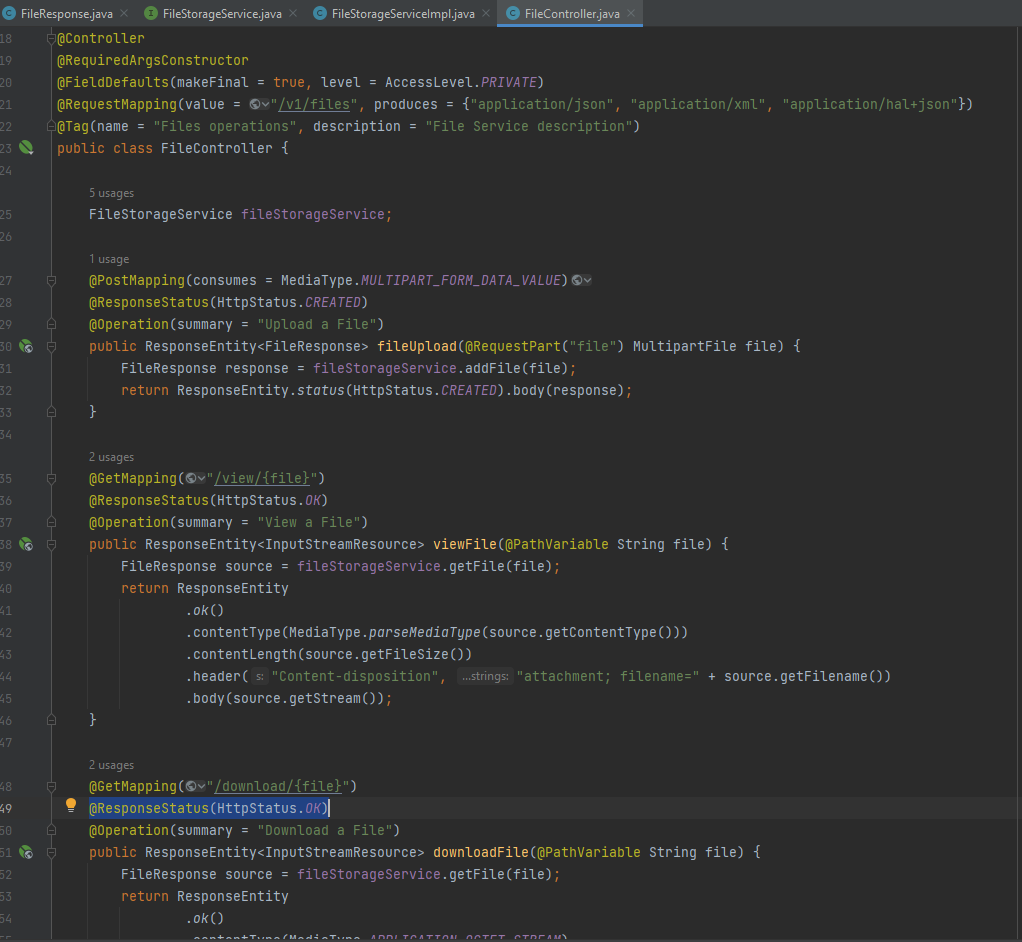


Рисунок 6. Класс FileController

Также следует создать утилитный класс с ошибками, чтобы использовать их в дальнейшем и указывать пользователю в чем может быть проблема (Рисунок 7).

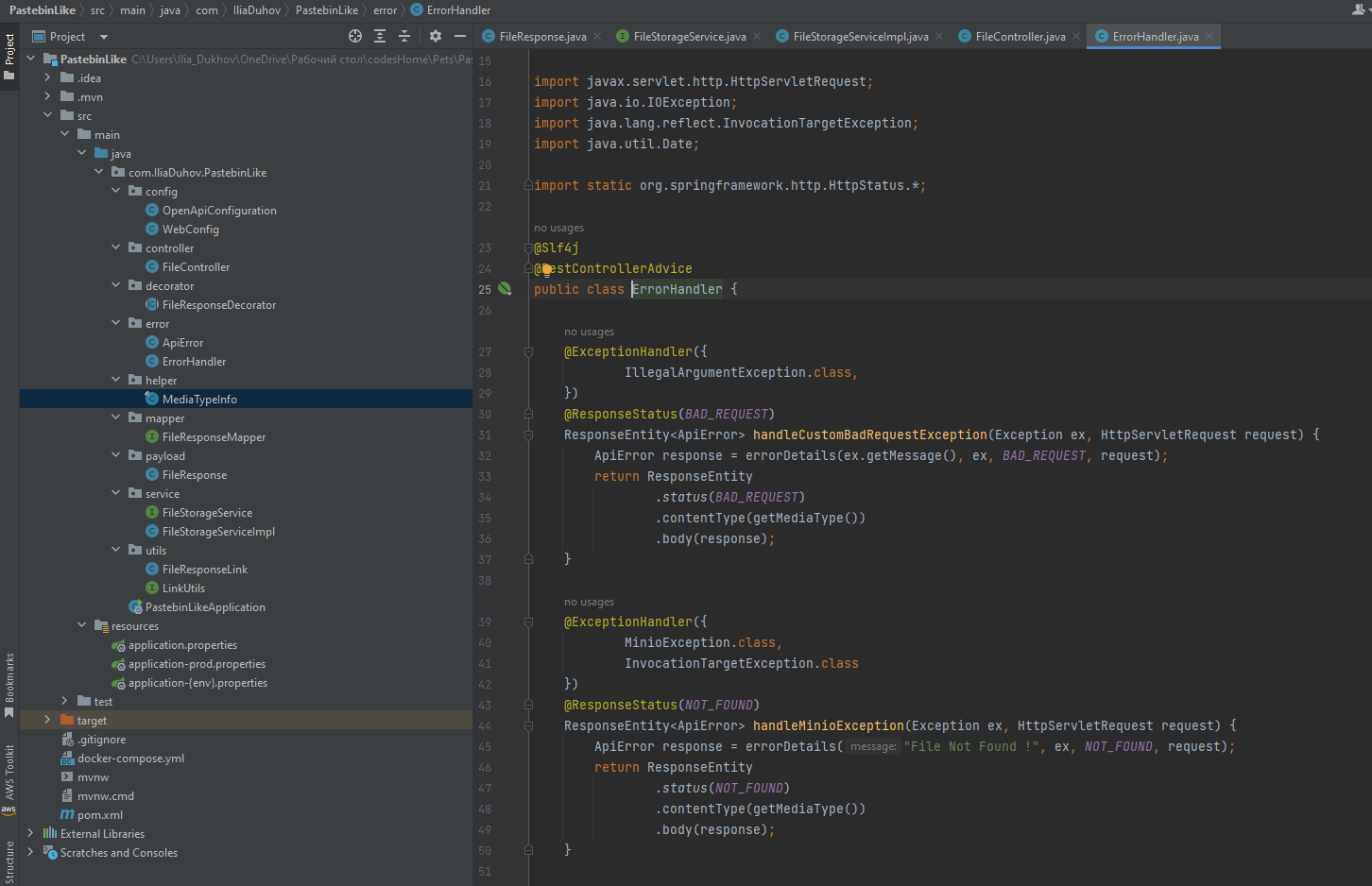


Рисунок 7. Класс ExceptionHandler

Для скачивания файлов и их редактирования необходимо использовать ResponseDecorator (Рисунок 8), чтобы преобразовывать ответы сервера в объекты.

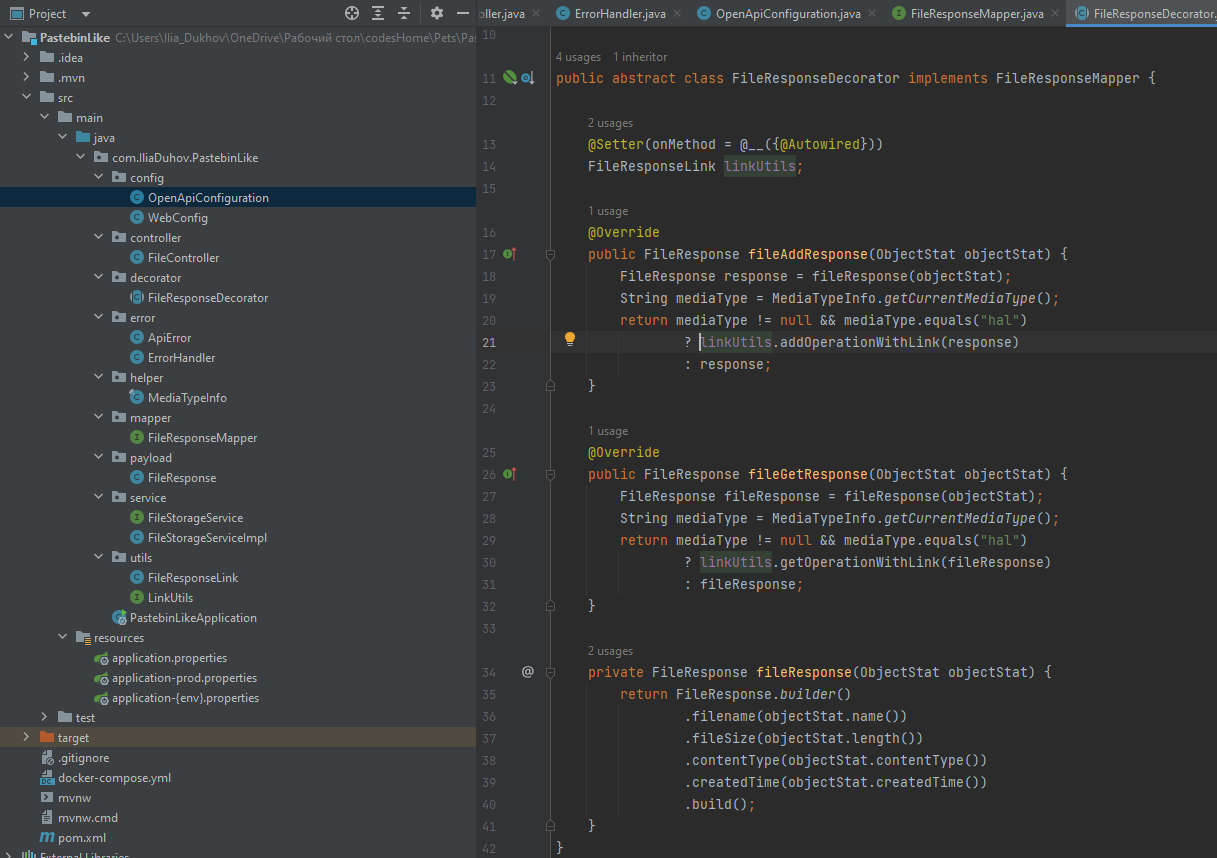


Рисунок 8. Класс ResponseDecorator

1. Настроить выбранное инструментальное средство документирования. Чтобы настроить Swagger и добиться его корректной работы – следует сделать несколько вещей. Нужно добавить конфигурацию в файле “application.properties”, чтобы мы могли указывать свои пути для методов. Так же следует создать веб и API конфигурацию (Рисунок 9).

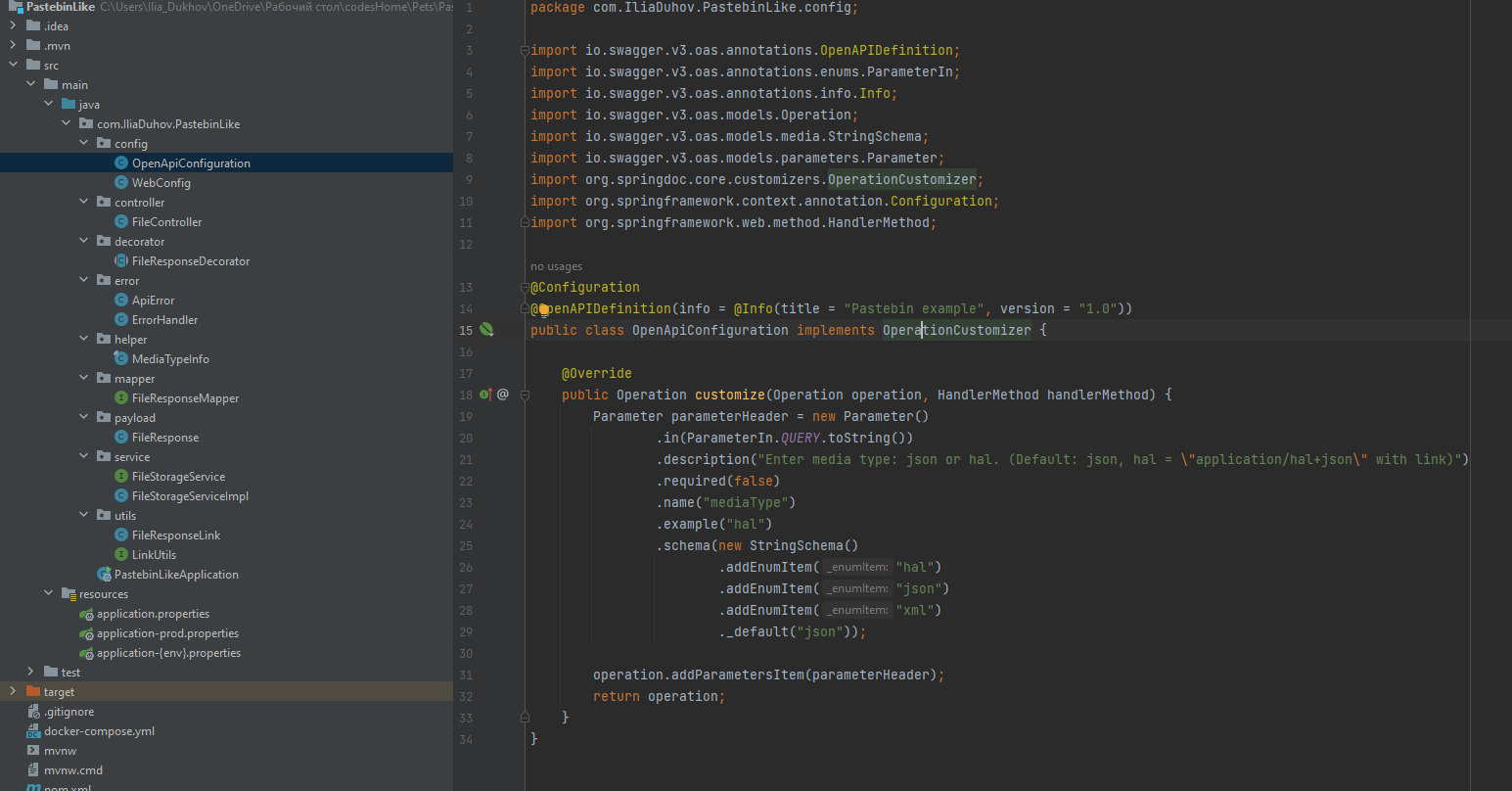


Рисунок 9. Класс OpenApiConfiguration

1. Показать возможности инструментального средства. Swagger используется для быстрого описания и документирования конечных точек (Рисунок 10).

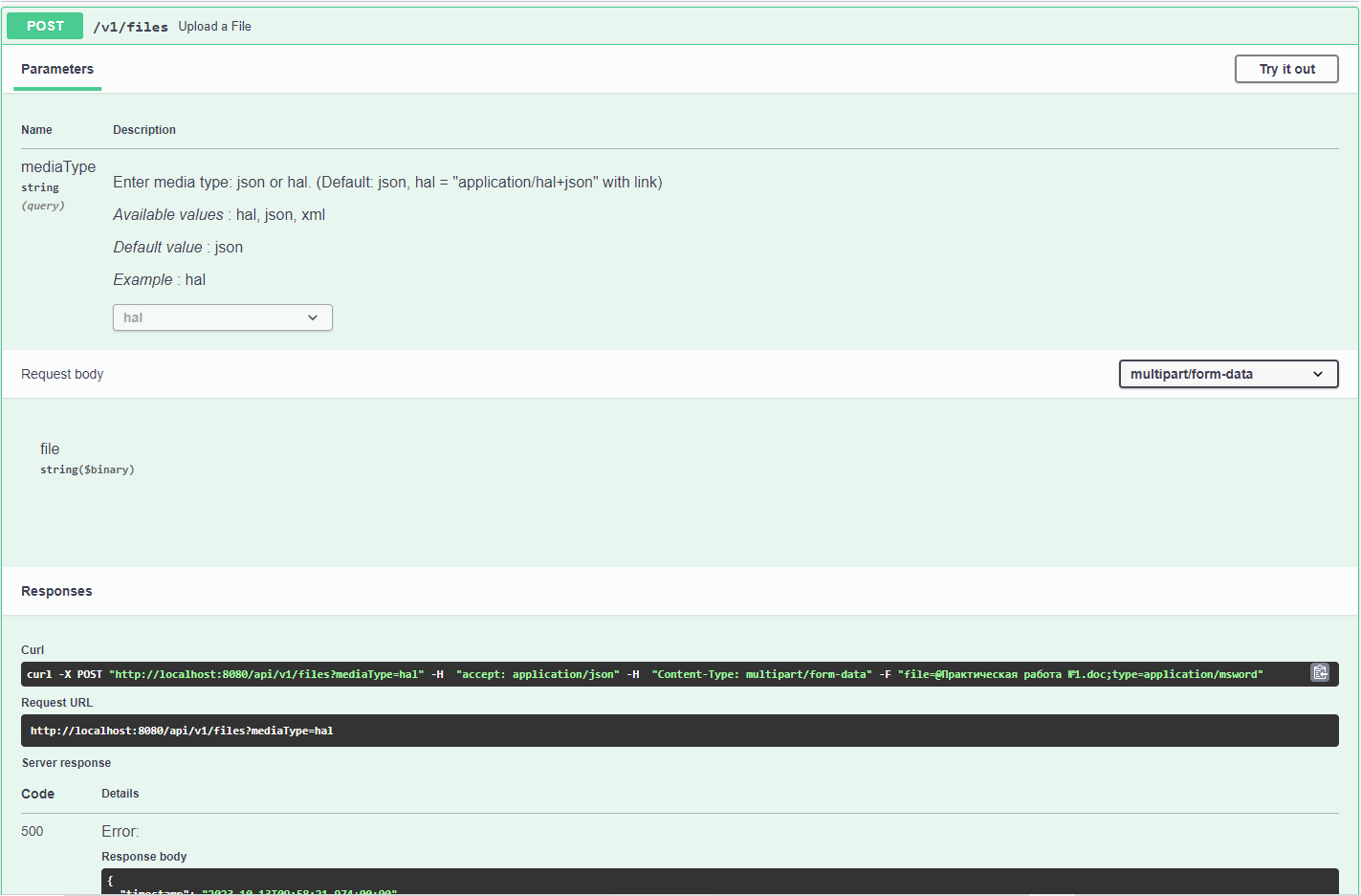


Рисунок 10. EndPoint загрузки файла в хранилище

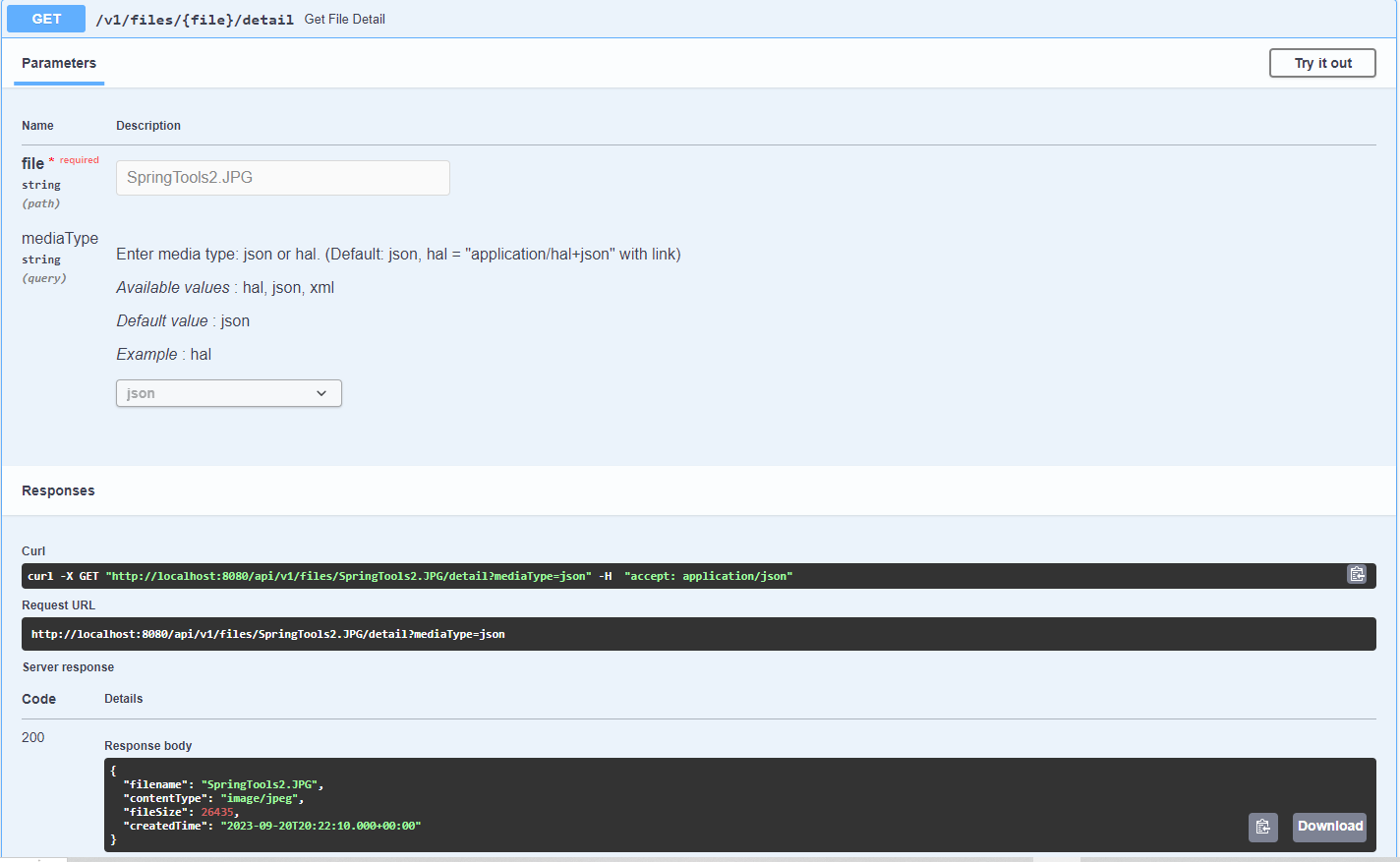


Рисунок 11. EndPoint загрузки файла в хранилище

Так же анализирую классы, поля и методы, помеченные аннотациями из пакета Validator - Swagger составит схемы, которые нужны для более наглядного понимания ограничений (Рисунок 11).



Рисунок 12. Schemas ограничения

1. Вывод: ходе выполнения практической работы я ознакомился с такой технологией, как Swagger для документирования и описания API. Он предоставляет структурированный и понятный способ описания функций, конечных точек, параметров и ответов API. Swagger является стандартом для описания RESTful API и предоставляет много полезных функций, которые упрощают разработку, тестирование и документирование API