СОДЕРЖАНИЕ

[1. Спецификации UserController: 4](#_Toc154230526)

[1.1. Метод login 4](#_Toc154230527)

[1.2. Метод logout 5](#_Toc154230528)

[1.3. Метод refresh 5](#_Toc154230529)

[1.4. Метод getUsers 6](#_Toc154230530)

[2. Спецификации FileController: 7](#_Toc154230531)

[2.1. Метод getFiles 7](#_Toc154230532)

[2.2. Метод uploadFile 8](#_Toc154230533)

[2.3. Метод downloadFile 8](#_Toc154230534)

[2.4. Метод deleteFile 9](#_Toc154230535)

[2.5. Метод searchFile 9](#_Toc154230536)

[2.6. Метод uploadAvatar 9](#_Toc154230537)

[2.7. Метод deleteAvatar 10](#_Toc154230538)

[2.8. Метод downloadFolder 10](#_Toc154230539)

[2.9. Метод uploadFolder 11](#_Toc154230540)

[2.10. Метод checkFile 11](#_Toc154230541)

# Спецификации UserController:

Метод registration

HTTP метод: POST

Путь: /registration

Описание: Регистрирует нового пользователя.

Параметры запроса:

* email (строка): Email нового пользователя.
* password (строка): Пароль нового пользователя.
* name (строка): Имя нового пользователя.
* surname (строка): Фамилия нового пользователя.

Успешный ответ:

* Код статуса: 201 (Created)
* Тело ответа: JSON-объект с данными о зарегистрированном пользователе.

Возможные ошибки:

* Ошибки, вызванные сервисом регистрации.
* Ошибки валидации входных данных.

## Метод login

HTTP метод: POST

Путь: /login

Описание: Аутентификация пользователя.

Параметры запроса:

* email (строка): Email пользователя.
* password (строка): Пароль пользователя.

Успешный ответ:

* Код статуса: 200 (OK)
* Тело ответа: JSON-объект с данными о пользователе и токеном обновления.

Возможные ошибки:

* Неверные учетные данные.
* Ошибки, вызванные сервисом аутентификации.

## Метод logout

HTTP метод: POST

Путь: /logout

Описание: Выход пользователя из системы.

Параметры запроса:

refreshToken (строка): Токен обновления.

Успешный ответ:

* Код статуса: 200 (OK)
* Тело ответа: JSON-объект с информацией об успешном выходе.

Возможные ошибки:

* Ошибки, вызванные сервисом выхода.

## Метод refresh

HTTP метод: POST

Путь: /refresh

Описание: Обновление токена пользователя.

Параметры запроса:

refreshToken (строка): Токен обновления.

Успешный ответ:

* Код статуса: 200 (OK)
* Тело ответа: JSON-объект с новым токеном обновления и данными пользователя.

Возможные ошибки:

* Ошибки, вызванные сервисом обновления токена.

## Метод getUsers

HTTP метод: GET

Путь: /users

Описание: Получение списка всех пользователей.

Успешный ответ:

* Код статуса: 200 (OK)
* Тело ответа: JSON-массив с данными о пользователях.

Возможные ошибки:

* Ошибки, вызванные сервисом получения пользователей.

Примечание:

В коде используются куки для передачи токенов, что предполагает использование механизма аутентификации на основе сессий.

Для обработки ошибок используется передача ошибок в обработчик next, что предполагает использование middleware для обработки ошибок.

# Спецификации FileController:

Метод createDir

HTTP метод: POST

Путь: /create-dir

Описание: Создание новой директории.

Параметры запроса:

* name (строка): Имя новой директории.
* type (строка): Тип файла (в данном контексте всегда "dir").
* parent (строка, необязательный): ID родительской директории.

Успешный ответ:

* Код статуса: 200 (OK)
* Тело ответа: JSON-объект с информацией о созданной директории и пользователе.

Возможные ошибки:

Пользователь не найден.

* Ошибка при создании директории.
* Ошибки валидации входных данных.

## Метод getFiles

HTTP метод: GET

Путь: /get-files

Описание: Получение списка файлов и директорий.

Параметры запроса:

sort (строка, необязательный): Критерий сортировки файлов (по умолчанию - по имени).

parent (строка, необязательный): ID родительской директории.

Успешный ответ:

Код статуса: 200 (OK)

Тело ответа: JSON-массив с информацией о файлах и директориях.

Возможные ошибки:

Ошибки при получении файлов.

## Метод uploadFile

HTTP метод: POST

Путь: /upload-file

Описание: Загрузка файла.

Параметры запроса:

file (файл): Загружаемый файл.

parent (строка, необязательный): ID родительской директории.

Успешный ответ:

Код статуса: 200 (OK)

Тело ответа: JSON-объект с информацией о загруженном файле и пользователе.

Возможные ошибки:

Ошибки при обработке файла.

Ошибки валидации входных данных.

## Метод downloadFile

HTTP метод: GET

Путь: /download-file

Описание: Скачивание файла.

Параметры запроса:

id (строка): ID файла для скачивания.

Успешный ответ:

Код статуса: 200 (OK)

Файл для скачивания.

Возможные ошибки:

Файл не найден.

Ошибки при скачивании файла.

## Метод deleteFile

HTTP метод: POST

Путь: /delete-file

Описание: Удаление файла или директории.

Параметры запроса:

id (строка): ID файла или директории для удаления.

Успешный ответ:

Код статуса: 200 (OK)

Тело ответа: JSON-объект с информацией об удалении и пользователе.

Возможные ошибки:

Файл или директория не найдены.

Ошибки при удалении файла или директории.

## Метод searchFile

HTTP метод: GET

Путь: /search-file

Описание: Поиск файла по имени.

Параметры запроса:

search (строка): Имя файла или директории для поиска.

Успешный ответ:

Код статуса: 200 (OK)

Тело ответа: JSON-массив с информацией о файлах и директориях, соответствующих запросу.

Возможные ошибки:

Ошибки при поиске файлов.

## Метод uploadAvatar

HTTP метод: POST

Путь: /upload-avatar

Описание: Загрузка аватара пользователя.

Параметры запроса:

file (файл): Загружаемый файл (аватар).

Успешный ответ:

Код статуса: 200 (OK)

Тело ответа: JSON-объект с информацией о загруженном аватаре и пользователе.

Возможные ошибки:

Ошибки при обработке файла.

Ошибки валидации входных данных.

## Метод deleteAvatar

HTTP метод: POST

Путь: /delete-avatar

Описание: Удаление аватара пользователя.

Успешный ответ:

Код статуса: 200 (OK)

Тело ответа: JSON-объект с информацией об удалении аватара и пользователе.

Возможные ошибки:

Ошибки при удалении аватара.

## Метод downloadFolder

HTTP метод: GET

Путь: /download-folder

Описание: Скачивание директории в виде архива.

Параметры запроса:

folderId (строка): ID директории для скачивания.

Успешный ответ:

Код статуса: 200 (OK)

Архив с содержимым директории для скачивания.

Возможные ошибки:

Директория не найдена.

Ошибки при создании архива.

## Метод uploadFolder

HTTP метод: POST

Путь: /upload-folder

Описание: Загрузка директории (в разработке).

Параметры запроса:

В разработке.

Успешный ответ:

В разработке.

Возможные ошибки:

В разработке.

## Метод checkFile

HTTP метод: POST

Путь: /check-file

Описание: Проверка возможности загрузки файла.

Параметры запроса:

name (строка): Имя файла.

size (число): Размер файла в байтах.

parent (строка, необязательный): ID родительской директории.

Успешный ответ:

Код статуса: 200 (OK)

Тело ответа: JSON-объект с результатом проверки.

Возможные ошибки:

Пользователь не найден.

Превышен максимальный размер файла.

Недостаточно места на диске.

Файл существует.

Ошибки валидации входных данных.