Проект по разработке сайта с интеграцией 1С:Предприятие 8.3

«Green Team»

Документация серверного API

Составители:

Патюков Дмитрий

Черепанова Анна

**API.Auth**

Метод Get (URL: /Auth)

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

Email - электронная почта

Password - пароль

Request body:

Отсутствуют

#### Responses(ответ на запрос):

В ответ сервер пришлёт токен с закодированной информацией об Id, Email, FullName и Role пользователя. Данный токен действует 1 час, после чего сессия завершается и нужно получать новый токен. Без этого токена некоторые запросы сервер будет игнорировать отсылая ошибку 401. Если авторизационные данные оказались неверны, то сервер также вернёт ошибку 401. На клиенте следует обрабатывать эту ошибку таким образом, чтобы не было фатального завершения разработанного клиента и чтобы пользователю писалось о неверности его введённых данных.

Тестовые данные:

Email: req@gmail.com

Password: 123456

URL: https://localhost:7230/Auth?email=req%[40gmail.com](https://40gmail.com/" \o "https://40gmail.com" \t "https://web.telegram.org/a/_blank)&password=123456

Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6IjkiLCJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjoic3RyaW5nIHN0cmluZyIsImh0dHA6Ly9zY2hlbWFzLnhtbHNvYXAub3JnL3dzLzIwMDUvMDUvaWRlbnRpdHkvY2xhaW1zL2VtYWlsYWRkcmVzcyI6InJlcUBnbWFpbC5jb20iLCJodHRwOi8vc2NoZW1hcy5taWNyb3NvZnQuY29tL3dzLzIwMDgvMDYvaWRlbnRpdHkvY2xhaW1zL3JvbGUiOiJVc2VyIiwibmJmIjoxNzA5MTgzMjA1LCJleHAiOjE3MDkxODY4MDUsImlzcyI6Ik15QXBpVG9rZW4iLCJhdWQiOiJNeUNsaWVudCJ9.83rTccrOlQvSoWgtIwxpmslD8dZ8oSGavrVnkCJfgN4

Ниже представлен рисунок 1 расшифровки данных из токена, которые можно получить на клиенте:

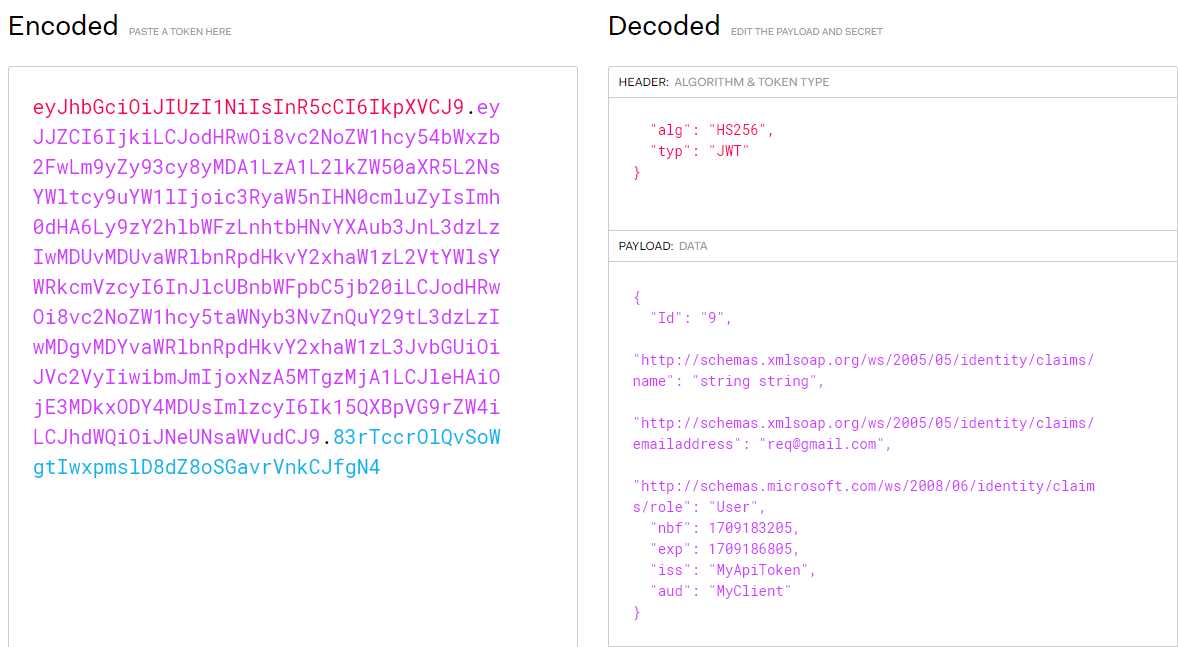


Рисунок 1 - Расшифровка токена

Метод Post (URL: /Auth)

#### Parameters(параметры строки):

Отсутствуют

Request body:

Файл формата .json с полями:

{

"id": 0,

"firstName": "string",

"lastName": "string",

"email": "string",

"password": "string",

"phone": "string",

}

Responses(ответ на запрос):

В ответ сервер пришлёт токен с закодированной информацией об Id, Email, FullName и Role пользователя. Данный токен действует 1 час, после чего сессия завершается и нужно получать новый токен. Без этого токена некоторые запросы сервер будет игнорировать отсылая ошибку 401.

Тестовые данные:

{  
   "id": 0,  
   "firstName": "Test1",  
   "lastName": "Test2",  
   "email": "[test@gmail.com](mailto:test@gmail.com" \t "https://web.telegram.org/a/_blank)",  
   "password": "12345",  
  "phone": "[88003535355](https://web.telegram.org/a/tel:88003535355)"  
 }

URL: https://localhost:7230/Auth

Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6IjEwMDkiLCJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjoiVGVzdDEgVGVzdDIiLCJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9lbWFpbGFkZHJlc3MiOiJ0ZXN0QGdtYWlsLmNvbSIsImh0dHA6Ly9zY2hlbWFzLm1pY3Jvc29mdC5jb20vd3MvMjAwOC8wNi9pZGVudGl0eS9jbGFpbXMvcm9sZSI6IlVzZXIiLCJuYmYiOjE3MDkxODQzNzAsImV4cCI6MTcwOTE4Nzk3MCwiaXNzIjoiTXlBcGlUb2tlbiIsImF1ZCI6Ik15Q2xpZW50In0.eeqDiatl0OEdlqYDtamoVj4wl4eZqNOK499DD2Oq99U12:25

Ниже представлен рисунок 2 с расшифровкой токена:

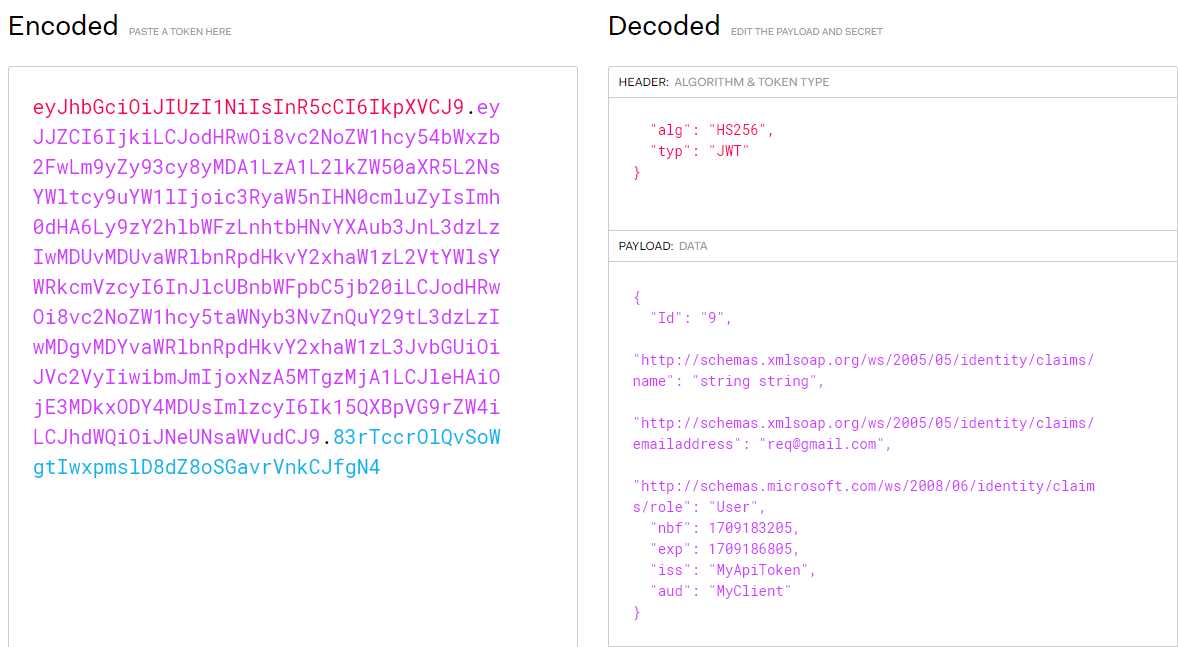


Рисунок 2 - Расшифровка токена

**Примечание:**

При получении токена, на клиенте необходимо сохранить его и далее при **любом** запросе отправлять этот токен в хедерах запроса:

Authorization: Bearer <Token>

## API.Product.Category

Метод Get (URL: /api/Category )

Авторизация: Авторизационные данные не нужны.

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(ответ на запрос):

Файл формата .json с полями:

#### {

#### "id": 0,

#### "name": "string",

#### "description": "string"

#### }

Метод Post (URL: /api/Category )

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры строки):

Отсутствуют

Request body:

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

#### Responses(ответ на запрос):

#### При успешном выполнении запроса: код 200

Метод Put (URL: /api/Category )

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

#### Responses(параметры ответа на запрос):

#### При успешном выполнении запроса: код 200

Метод Delete (URL: /api/Category )

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

#### Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Субметод GetAll (URL: /api/Category/all )

Авторизация: Авторизационные данные не нужны

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

## API.Product.Image

Метод Get (URL: /api/Image)

Все файлы лежат в расшаренной папке Products на сервере

Авторизация: Авторизационные данные не нужны.

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

productId- идентификатор продукта

Request body:

Отсутствуют

#### Responses(ответ на запрос):

Имя файла

Метод Post (URL: /api/Image)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры строки):

Отсутствуют

Request body(multipart/form-data):

Image - файл с изображением

Product\_id- идентификатор поста

#### Responses(ответ на запрос):

#### При успешном выполнении запроса: код 200

Метод Put (URL: /api/Image)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

Отсутствуют

Request body(multipart/form-data):

Image - файл с изображением

Product\_id- идентификатор поста

Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Метод Delete (URL: /api/Image)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

#### Параметр запроса должен содержать:

productId- идентификатор продукта

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Субметод GetAll (URL: /api/Image/all )

Авторизация: Авторизационные данные не нужны

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

Список путей к файлам

## API.Product.Product

Метод Get (URL: /api/Product)

Авторизация: Авторизационные данные не нужны.

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(ответ на запрос):

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

"amount": 0,

"price": 0,

"supplierId": 0,

"categoryId": 0,

}

Метод Post (URL: /api/Product)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры строки):

Отсутствуют

Request body:

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

"amount": 0,

"price": 0,

"supplierId": 0,

"categoryId": 0,

}

Responses(ответ на запрос):

При успешном выполнении запроса: код 200

Метод Put (URL: /api/Product)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

"amount": 0,

"price": 0,

"supplierId": 0,

"categoryId": 0,

}

#### Responses(параметры ответа на запрос):

#### При успешном выполнении запроса: код 200

Метод Delete (URL: /api/Product)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

#### Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Субметод GetAll (URL: /api/Product/all )

Авторизация: Авторизационные данные не нужны.

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Отсутствуют

Responses(параметры ответа на запрос):

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

"amount": 0,

"price": 0,

"supplierId": 0,

"categoryId": 0,

}

## API.Product.Supplier

Метод Get (URL: /api/Supplier)

Авторизация: Авторизационные данные не нужны.

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(ответ на запрос):

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

Метод Post (URL: /api/Supplier)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры строки):

Отсутствуют

Request body:

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

Responses(ответ на запрос):

При успешном выполнении запроса: код 200

Метод Put (URL: /api/Supplier)

Авторизация: Нужно быть пользователем с ролью Admin

Parameters(параметры запроса):

Отсутствуют

Request body:

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

#### Responses(параметры ответа на запрос):

#### При успешном выполнении запроса: код 200

Метод Delete (URL: /api/Supplier)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

#### Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Субметод GetAll (URL: /api/Supplier/all )

Авторизация: Авторизационные данные не нужны.

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

## API.User.Role

Метод Get (URL: /api/Role)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(ответ на запрос):

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

Метод Post (URL: /api/Role)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры строки):

Отсутствуют

Request body:

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

Responses(ответ на запрос):

При успешном выполнении запроса: код 200

Метод Put (URL: /api/Role)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

#### Responses(параметры ответа на запрос):

#### При успешном выполнении запроса: код 200

Метод Delete (URL: /api/Role)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

#### Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Субметод GetAll (URL: /api/Role/all )

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

Файл формата .json с полями:

{

"id": 0,

"name": "string",

"description": "string",

}

## API.User.User

Метод Get (URL: /api/User)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(ответ на запрос):

Файл формата .json с полями:

{

"id": 0,

"firstName": "string",

"lastName": "string",

"email": "string",

"password": "string",

"phone": "string",

}

Метод Post (URL: /api/User)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры строки):

roleId - идентификатор роли, которую мы хотим присвоить пользователю

Request body:

Файл формата .json с полями:

{

"id": 0,

"firstName": "string",

"lastName": "string",

"email": "string",

"password": "string",

"phone": "string"

}

Responses(ответ на запрос):

При успешном выполнении запроса: код 200

Метод Put (URL: /api/User)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

roleId - идентификатор роли, которую мы хотим присвоить пользователю

Request body:

Файл формата .json с полями:

{

"id": 0,

"firstName": "string",

"lastName": "string",

"email": "string",

"password": "string",

"phone": "string"

}

#### Responses(параметры ответа на запрос):

#### При успешном выполнении запроса: код 200

Метод Delete (URL: /api/User)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

#### Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Субметод GetAll (URL: /api/User/all )

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

Файл формата .json с полями:

{

"id": 0,

"firstName": "string",

"lastName": "string",

"email": "string",

"password": "string",

"phone": "string"

}

## API.User.Cheque

Метод Get (URL: /api/Cheque)

Авторизация: Нужно быть авторизированным пользователем с любой ролью

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(ответ на запрос):

Файл формата .json с полями:

{  
  "id": 0,  
  "path": "string",  
  "userId": 0,  
  "date": "2024-02-29T06:07:05.352Z"  
}

Метод Post (URL: /api/Cheque)

Авторизация: Нужно быть авторизированным пользователем с любой ролью

#### Parameters(параметры строки):

Отсутствуют

Request body:

Файл формата .json с полями:

{  
  "id": 0,  
  "path": "string",  
  "userId": 0,  
  "date": "2024-02-29T06:07:05.352Z"  
}

Responses(ответ на запрос):

При успешном выполнении запроса: код 200

Метод Put (URL: /api/Cheque)

Авторизация: Нужно быть пользователем с ролью 1C

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Файл формата .json с полями:

{  
  "id": 0,  
  "path": "string",  
  "userId": 0,  
  "date": "2024-02-29T06:07:05.352Z"  
}

#### Responses(параметры ответа на запрос):

#### При успешном выполнении запроса: код 200

Метод Delete (URL: /api/Cheque)

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

#### Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Субметод GetAll (URL: /api/Cheque/all )

Авторизация: Нужно быть авторизированным пользователем с любой ролью

#### Parameters(параметры запроса):

userId - идентификатор пользователя

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

Список файлов формата .json с полями:

{  
 "id": 0,  
 "path": "string",  
 "userId": 0,  
 "date": "2024-02-29T06:07:05.352Z"  
}

## API.User.ChequeProduct

Метод Get (URL: /api/ChequeProduct)

Авторизация:

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(ответ на запрос):

Файл формата .json с полями:

{  
  "id": 0,  
  "productId": 0,  
  "chequeId": 0,  
  "productAmount": 0  
}

Метод Post (URL: /api/ChequeProduct)

Авторизация:

#### Parameters(параметры строки):

Отсутствуют

Request body:

Файл формата .json с полями:

{  
  "id": 0,  
  "productId": 0,  
  "chequeId": 0,  
  "productAmount": 0  
}

Responses(ответ на запрос):

При успешном выполнении запроса: код 200

Метод Put (URL: /api/ChequeProduct)

Авторизация:

#### Parameters(параметры запроса):

Отсутствуют

Request body:

Файл формата .json с полями:

{  
  "id": 0,  
  "productId": 0,  
  "chequeId": 0,  
  "productAmount": 0  
}

#### Responses(параметры ответа на запрос):

#### При успешном выполнении запроса: код 200

Метод Delete (URL: /api/ChequeProduct)

Авторизация:

#### Parameters(параметры запроса):

#### Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Субметод GetAll (URL: /api/ChequeProduct/all )

Авторизация: Нужно быть пользователем с ролью Admin

#### Parameters(параметры запроса):

chequeId - идентификатор чека

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

Список файлов формата .json с полями:

{

"id": 0,

"productId": 0,

"chequeId": 0,  
 "productAmount": 0  
 }

## API.User.File

Метод Get (URL: /api/File)

Все файлы лежат в расшаренной папке Cheques на сервере

Авторизация: Нужно быть авторизованным пользователем с любой ролью

#### Parameters(параметры запроса):

Параметр запроса должен содержать:

Id - идентификатор сущности в базе данных MS SQL

Request body:

Отсутствуют

#### Responses(ответ на запрос):

Путь до файла чека

Метод Post (URL: /api/File)

Авторизация: Нужно быть пользователем с ролью 1C

#### Parameters(параметры строки):

Отсутствуют

Request body(multipart/form-data):

File - файл чека

User\_id - идентификатор пользователя

Cheque\_id - идентификатор чека

Responses(ответ на запрос):

При успешном выполнении запроса: код 200

Метод Put (URL: /api/File)

Авторизация: Нужно быть пользователем с ролью 1C

#### Parameters(параметры запроса):

Отсутствуют

Request body(multipart/form-data):

File - файл чека

User\_id - идентификатор пользователя

Cheque\_id - идентификатор чека

#### Responses(параметры ответа на запрос):

#### При успешном выполнении запроса: код 200

Метод Delete (URL: /api/File)

Авторизация: Нужно быть пользователем с ролью 1C

#### Parameters(параметры запроса):

#### Параметр запроса должен содержать:

chequeId - идентификатор чека

userId - идентификатор пользователя

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

При успешном выполнении запроса: код 200

Субметод GetAll (URL: /api/File/all )

Авторизация: Нужно быть пользователем с любой ролью

#### Parameters(параметры запроса):

userId - идентификатор пользователя

Request body:

Отсутствуют

#### Responses(параметры ответа на запрос):

Список путей к файлам чеков