

**ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»**

**СИСТЕМА РЕДАКТИРОВАНИЯ И СОВМЕСТНОЙ РАБОТЫ**

**СИСТЕМА ХРАНЕНИЯ ДАННЫХ**

**3.2**

**РУКОВОДСТВО ПО РАБОТЕ С API**

**Версия 1**

**На 146 листах**

**Дата публикации: 17.12.2024**

**Москва  
2024**

# МойОфис

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам.

Товарные знаки «МойОфис», «MyOffice» и «Squadus» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

Ни при каких обстоятельствах нельзя истолковывать любое содержимое настоящего документа как прямое или косвенное предоставление лицензии или права на использование товарных знаков, логотипов или знаков обслуживания, приведенных в нем. Любое несанкционированное использование этих товарных знаков, логотипов или знаков обслуживания без письменного разрешения их правообладателя строго запрещено.

## СОДЕРЖАНИЕ

1	Общие сведения .....	8
1.1	Назначение .....	8
1.2	О компонентах .....	8
1.3	Требования к квалификации .....	8
1.4	Системные требования .....	8
1.5	Ограничения .....	9
2	Подготовка к работе .....	10
2.1	Запуск системы .....	10
2.2	Авторизация .....	10
3	Обзор методов .....	11
4	Справочник запросов .....	13
4.1	Запрос токена авторизации .....	13
4.2	Работа с доменами .....	15
4.2.1	Чтение списка доменов .....	15
4.2.2	Чтение списка доменов в тенанте .....	16
4.2.3	Добавление домена в тенант .....	17
4.2.4	Удаление домена из тенанта .....	18
4.3	Работа с параметрами SMTP конфигурации .....	20
4.3.1	Чтение параметров конфигурации SMTP .....	20
4.3.2	Настройка конфигурации SMTP .....	21
4.3.3	Обновление конфигурации SMTP .....	23
4.3.4	Удаление значений параметров конфигурации SMTP .....	25
4.4	Работа с тенантами .....	27
4.4.1	Перечень тенантов .....	27
4.4.2	Создание нового тенанта .....	29
4.4.3	Чтение параметров тенанта .....	32
4.4.4	Деактивация тенанта .....	34
4.4.5	Удаление тенанта .....	36

4.4.6	Чтение установленных параметров настроек тенанта .....	37
4.4.7	Изменение значений параметров настроек тенанта .....	38
4.4.8	Чтение установленных параметров работы с событиями по объектам .....	40
4.4.9	Изменение параметров работы с событиями по объектам .....	41
4.4.10	Изменение параметров журнала безопасности .....	43
4.4.11	Изменение параметров автоверсионирования .....	45
4.4.12	Смена имени организации в тенанте .....	47
4.4.13	Чтение парольных политик в тенанте .....	48
4.4.14	Изменение парольных политик в тенанте .....	49
4.4.15	Чтение параметров квот в тенанте .....	51
4.4.16	Изменение параметров квот в тенанте .....	53
4.4.17	Чтение параметров двухфакторной авторизации в тенанте .....	54
4.4.18	Изменение параметров двухфакторной авторизации в тенанте .....	55
4.4.19	Получить список запретов в тенанте .....	58
4.4.20	Включить запрет определенного типа в тенанте .....	58
4.4.21	Выключить запрет определенного типа в тенанте .....	59
4.4.22	Получить список запретов по типу в тенанте .....	60
4.4.23	Добавить запрет или исключение в тенанте .....	62
4.4.24	Удалить запрет или исключение в тенанте .....	63
4.5	Работа с проверочным кодом (CAPTCHA) .....	64
4.5.1	Установка проверочного кода .....	64
4.5.2	Просмотр настроек проверочного кода .....	65
4.5.3	Изменение настроек проверочного кода .....	66
4.6	Работа с общими папками .....	69
4.6.1	Получение списка общих папок в тенанте .....	69
4.6.2	Создание общей папки в тенанте .....	71
4.6.3	Получение информации об общей папке .....	72
4.6.4	Изменение атрибутов общей папки .....	73
4.6.5	Удаление общей папки .....	75
4.7	Работа с пользователями .....	77
4.7.1	Перечень пользователей .....	77

4.7.2	Создание нового пользователя .....	81
4.7.3	Чтение информации о пользователе .....	83
4.7.4	Изменение информации о пользователе .....	85
4.7.5	Удаление и блокировка пользователя .....	88
4.7.6	Принудительная деавторизация пользователя .....	90
4.7.7	Чтение списка общих папок пользователя .....	91
4.7.8	Чтение списка файлов пользователя .....	92
4.7.9	Чтение информации о файловой квоте пользователя .....	93
4.7.10	Изменение файловой квоты пользователя .....	95
4.7.11	Изменение статуса двухфакторной авторизации пользователя .....	96
4.7.12	Просмотр удаленных файлов пользователя .....	97
4.7.13	Восстановление удаленных файлов пользователя .....	98
4.7.14	Окончательно удалить файлы пользователя .....	101
4.7.15	Получить аватар пользователя .....	102
4.7.16	Изменить аватар пользователя .....	103
4.7.17	Удалить аватар пользователя .....	104
4.7.18	Изменить персональные данные пользователя .....	105
4.8	Работа с группами пользователей .....	107
4.8.1	Перечень групп пользователей .....	107
4.8.2	Создание новой группы пользователей .....	109
4.8.3	Чтение информации о группе .....	111
4.8.4	Изменение имени группы .....	112
4.8.5	Удаление группы .....	114
4.8.6	Добавление пользователя в группу .....	115
4.8.7	Удаление пользователя из группы .....	116
4.8.8	Получение списка подгрупп .....	118
4.8.9	Добавление подгруппы .....	119
4.9	Работа с публичными ссылками .....	122
4.9.1	Получение списка публичных ссылок .....	122
4.9.2	Блокировка и разблокировка публичной ссылки .....	124
4.9.3	Удаление публичной ссылки .....	125

4.9.4	Изменение настроек публичных ссылок в тенанте .....	126
4.10	Работа с ключами доступа .....	129
4.10.1	Получение списка ключей доступа в тенанте .....	129
4.10.2	Создание ключа доступа в тенанте .....	130
4.10.3	Удаление ключа доступа в тенанте .....	131
4.11	Работа с медиатипами .....	133
4.11.1	Получение списка групп с запретами на загрузку медиатипов .....	133
4.11.2	Изменение списка групп с запретами за загрузку медиатипов .....	134
4.11.3	Чтение запретов на загрузку медиатипов для группы .....	136
4.11.4	Изменение запретов на загрузку медиатипов для группы .....	137
4.11.5	Удаление запретов на загрузку медиатипов для группы .....	138
4.11.6	Получение списка пользователей с запретами на загрузку медиатипов .....	140
4.11.7	Чтение запретов на загрузку медиатипов для пользователя .....	141
4.11.8	Изменение запретов на загрузку медиатипов для пользователя .....	143
4.11.9	Удаление запретов на загрузку медиатипов для пользователя .....	145

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

В настоящем документе используются следующие сокращения с соответствующими расшифровками (табл. 1).

Таблица 1 — Сокращения и обозначения

Сокращение, термин	Расшифровка и определение
API	Application Programming Interface, интерфейс программирования приложений
CO	Система редактирования и совместной работы
PGS	Система хранения данных
ОС	Операционная система
Тенант	Логический объект, включающий в себя совокупность вычислительных ресурсов, репозиторий и пользователей
ПО	Программное обеспечение

## 1 ОБЩИЕ СВЕДЕНИЯ

### 1.1 Назначение

В настоящем документе описана архитектура продукта и взаимодействие сервисов Системы редактирования и совместной работы и Системы хранения данных.

### 1.2 О компонентах

Система хранения данных — компонент, предназначенный для безопасного хранения корпоративных файлов и обеспечения возможностей авторизации, аутентификации и разграничения прав доступа пользователей.

Система редактирования и совместной работы — компонент, предназначенный для индивидуального и совместного редактирования текстовых и табличных документов, а также просмотра и демонстрации презентаций.

Представленные компоненты входят в состав следующих продуктов:

- «МойОфис Частное Облако 3»;
- «МойОфис Профессиональный 2»;
- «МойОфис Профессиональный 3»;
- «МойОфис Схема»;
- Squadus PRO.

Список возможностей приложения приведен в документе «Функциональные возможности»

### 1.3 Требования к квалификации

Пользователь Admin API должен иметь следующие навыки:

- работа с REST API;
- администрирование информационных систем.

Для работы с ПО «МойОфис» администратору необходимо ознакомиться со следующими документами:

- «Руководство по настройке»;
- «Руководство по администрированию».

### 1.4 Системные требования

Перечень требований к программному и аппаратному обеспечению приведен в документе «Системные требования».



## **1.5 Ограничения**

Запросы к API выполняются по протоколу HTTPS. По умолчанию сервис возвращает данные в формате JSON.

## 2 ПОДГОТОВКА К РАБОТЕ

### 2.1 Запуск системы

Для начала работы с Admin API выполните последовательность действий:

1. Откройте веб-браузер при активном сетевом подключении.
2. Введите адрес Admin API ПО МойОфис в адресную строку веб-браузера и перейдите по ссылке.

Пример адреса:

```
https://admin-<ENV>.<DEFAULT_DOMAIN>/adminapi
```

где: <ENV> и <DEFAULT\_DOMAIN> — переменные, указанные при установке PGS.

Подробная информация по переменным указана в документе «Система хранения данных. Руководство по установке».

### 2.2 Авторизация

Для авторизации и аутентификации используется протокол OAuth 2.0. Пользователь в соответствии с разрешениями получает доступ к API сервиса PGS Euclid по адресу:

```
https://admin-<ENV>.<DEFAULT_DOMAIN>/adminapi
```

## 3 ОБЗОР МЕТОДОВ

В интерфейсе программирования приложений (API) доступны функции работы с доменами, тенантами, пользователями и группами пользователей. Полный перечень представлен в таблице 2.

Таблица 2— Перечень доступных функций и методов

Наименование ресурса	PUT	POST	GET	DELETE
Auth — Получение токена авторизации	-	+	-	-
Domains — Работа с доменами	-	-	+	-
SmtпConfig — Работа с параметрами SMTP конфигурации	+	+	+	+
Tenants — Работа с тенантами	-	+	+	-
Tenant — Работа с тенантом	-	-	+	+
TenantAuditSettings — Параметры журнала безопасности	+	-	-	-
TenantAutoversionSettings — Параметры автоверсионирования	+	-	-	-
Captcha — Параметры проверочного кода (капчи)	-	+	-	-
TenantCaptchaSettings — Параметры капчи для тенанта	+	-	+	-
TenantCommonSettings — Параметры общих настроек тенанта	+	-	+	-
CorpFolders — Работа с общими папками	-	+	+	-
CorpFolder — Настройка общей папки	+	-	+	+
GroupsCtypesProhibition — Работа с запретами для групп	+	-	+	-
GroupCtypesProhibition — Работа с запретами для группы	+	-	+	+
UsersCtypesProhibition — Работа с запретами для пользователей	+	-	+	-
UserCtypesProhibition — Работа с запретами для пользователя	+	-	+	+
DomainsInTenant — Работа с доменами в тенанте	-	+	+	-
Domain — Работа с доменом в тенанте	-	-	-	+
PromoCodes — Работа с ключами доступа	+	+	+	-
PromoCode — Работа с ключом доступа	+	-	+	+
Groups — Работа с группами	-	+	+	-

Наименование ресурса	PUT	POST	GET	DELETE
Group — Работа с группой	+	-	+	+
Subgroups — Работа с подгруппами	-	+	+	-
GroupMembers — Работа с пользователями в группах	-	+	-	+
TenantHistorySettings — Работа с историей по объектам	+	-	+	-
TenantOrganisationName — Изменение имени организации в тенанте	+	-	-	-
TenantPasswordPolicies — Работа с парольными политиками тенанта	+	-	+	-
TenantPublicLinkSettings — Настройка публичных ссылок	+	-	-	-
PublicLinks — Работа с публичными ссылками	-	-	+	+
PublicLink — Работа с публичной ссылкой	+	-	-	-
TenantQuota — Работа с квотой для тенанта	+	-	+	-
FileRecovery — Работа с восстановлением файлов	-	+	+	+
TenantTwoFaPolicies — Работа с двухфакторной авторизацией в тенанте	+	-	+	-
Users — Работа с пользователями	-	+	-	-
User — Работа с пользователем	+	-	+	+
UserCorpFolders — Работа с общими папками пользователей	-	-	+	-
Files — Работа с пользовательскими файлами	-	-	+	-
Logout — Работа с авторизацией пользователя в системе	-	+	-	-
UserQuota — Работа с пользовательской квотой	+	-	+	-
ResetUserTwoFa — Работа с двухфакторной авторизацией пользователя	-	-	-	+

## 4 СПРАВОЧНИК ЗАПРОСОВ

Данный раздел содержит сведения по работе с функциональными возможностями продукта через API сервиса PGS Euclid, включая возможности, не реализованные в пользовательском интерфейсе администратора.

### 4.1 Запрос токена авторизации

#### Метод и конечная точка

POST/auth

#### Описание

Отправляет данные пользователя. Возвращает токен авторизации администратору тенанта, в остальных случаях выдает ошибку.

#### Параметры

Параметры для метода указаны в таблице 3.

Таблица 3 — Перечень доступных параметров для авторизации

Название	Обязательный	Тип данных	Описание
username	да	string	Имя пользователя администратора веб-интерфейса
password	да	string	Пароль администратора веб-интерфейса
otp	нет	string	Одноразовый пароль (one-time password) для входа, если подключена двухфакторная аутентификация

#### Запрос

```
curl -X 'POST' \  
"https://admin-<ENV>.<DEFAULT_DOMAIN>/adminapi/auth?\  
username=<ADMIN_USERNAME>&password=<ADMIN_PASSWORD>&otp=<OTP>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- `<ADMIN_USERNAME>` — имя пользователя администратора веб-интерфейса;
- `<ADMIN_PASSWORD>` — пароль администратора веб-интерфейса;
- `<ОТР>` — одноразовый пароль для двухфакторной аутентификации, от 6 до 8 цифр.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешная авторизация. Пример вывода:

```
{
  "id": 0,
  "password_expiration_time": 0,
  "tenant": "string",
  "token": "string"
}
```

- 401 — Unauthorized — Не авторизован. Пример вывода:

```
{
  "message": "string"
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.2 Работа с доменами

### 4.2.1 Чтение списка доменов

#### Метод и конечная точка

GET/domains

#### Описание

Выводит перечень всех доменов во всех тенантах.

#### Параметры

Не требует ввода параметров.

#### Запрос

```
curl -X 'GET' \  
'https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/domains' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

#### Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{  
  "domains":  
  [ "string"  
  ]  
}
```

## 4.2.2 Чтение списка доменов в тенанте

### Метод и конечная точка

GET/tenants/{tenant\_name}/domains

### Описание

Выводит перечень всех доменов в указанном тенанте.

### Параметры

Перечень параметров представлен в таблице 4.

Таблица 4 — Перечень параметров для вывода списка доменов в тенанте

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/domains" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

— **200** — Success — Успешно. Пример вывода:

```
{  
  "domains":  
  [  
    "string"  
  ]  
}
```



## 4.2.3 Добавление домена в тенант

### Метод и конечная точка

POST/tenants/{tenant\_name}/domains

### Описание

Добавляет домен в указанный тенант.

### Параметры

Перечень параметров представлен в таблице 5.

Таблица 5 — Перечень параметров для добавления домена в тенант

Название	Обязательный	Тип данных	Описание
name	да	string	Имя добавляемого домена
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/domains?name=<DOMAIN_NAME>" \  
-H 'accept:application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "domains":  
    [ "string" ],  
  "name": "string"  
}
```

- 400 — Domain is invalid — Неверно указан домен. Пример вывода:

```
{  
  "message": "string"  
}
```

- 409 — Domain already exists — Домен уже существует. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.2.4 Удаление домена из тенанта

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/domains/{domain}

### Описание

Удаляет домен из тенанта.

### Параметры

Перечень параметров представлен в таблице 6.

Таблица 6 — Перечень параметров для удаления домена

Название	Обязательный	Тип данных	Описание
domain	да	string	Имя удаляемого домена
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/domains?domain=<DOMAIN_NAME>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "domains":  
  [ "string"  
  ]  
}
```

- 400 — Cannot delete default domain or cannot delete domain because some users have emails with it — Невозможно удалить домен, так как у некоторых пользователей есть почтовый ящик с его именем. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.3 Работа с параметрами SMTP конфигурации

### 4.3.1 Чтение параметров конфигурации SMTP

#### Метод и конечная точка

GET/sntp\_conf/{tenant\_name}

#### Описание

Возвращает значения параметров конфигурации SMTP тенанта.

#### Параметры

Параметры для метода указаны в таблице 7.

Таблица 7 — Перечень доступных параметров для чтения параметров SMTP

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

#### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\sntp_conf/<TENANT_NAME>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где <TENANT\_NAME> — имя тенанта.

#### Ответ

– 200 — Success — Успешно. Пример вывода:

```
{  
  "response": {  
    "default_domain": {  
      "enc_ssl_key": "string",  
      "name": "string",  
      "ssl_cert": "string"},  
    "id": "string",  
    "name": "string",  
    "props": {  
      "esia": {  
        "can_esia_auth": 0,  
        "esia_cert": "string",  
        "esia_client_id": "string",  
        "esia_expire_ts": 0,  
      }  
    }  
  }  
}
```

```
    "esia_key": "string",
    "support_esia_auth": 0
  },
  "support": {
    "email": "string",
    "phone": "string",
    "smtp": {
      "address": "string",
      "enc_password": "string",
      "login": "string",
      "port": 0,
      "ssl": 0
    }
  }
},
"success": "string",
"support_domains": [
"string"
]
},
"success": "string"
}
```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.3.2 Настройка конфигурации SMTP

### Метод и конечная точка

POST/smtp\_conf/{tenant\_name}

### Описание

Отправляет значения параметров конфигурации SMTP тенанта.

### Параметры

Параметры для метода указаны в таблице 8.

Таблица 8 — Перечень доступных параметров для настройки конфигурации SMTP

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
smtp_conf/<TENANT_NAME>" \  
-H 'accept: application/json' \  
-H 'Content-Type: application/json' \  
-d '{  
  "cert": "string",  
  "config": {  
    "can_esia_auth": 0,  
    "esia_cert": "string",  
    "esia_client_id": "string",  
    "esia_expire_ts": 0,  
    "esia_key": "string",  
    "smtp_login": "string",  
    "smtp_port": 0,  
    "support_esia_auth": 0  
  },  
  "key": "string"  
}'  
-H 'Authorization: <TOKEN>'
```

## Ответ

– 200 — Success — Успешно. Пример вывода:

```
{  
  "response": {  
    "default_domain": {  
      "enc_ssl_key": "string",  
      "name": "string",  
      "ssl_cert": "string"},  
    "id": "string",  
    "name": "string",  
    "props": {  
      "esia": {  
        "can_esia_auth": 0,  
        "esia_cert": "string",  
        "esia_client_id": "string",  
        "esia_expire_ts": 0,  
        "esia_key": "string",  
        "support_esia_auth": 0  
      },  
      "support": {  
        "email": "string",  
        "phone": "string",  
        "smtp": {  
          "address": "string",  
          "enc_password": "string",  
          "login": "string",  
          "port": 0,  
          "ssl": 0  
        }  
      }  
    }  
  }  
}
```

```
},  
"success": "string",  
"support_domains": [  
"string"  
]  
},  
"success": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

### 4.3.3 Обновление конфигурации SMTP

#### Метод и конечная точка

PUT/smtp\_conf/{tenant\_name}

#### Описание

Отправляет обновление параметров конфигурации SMTP тенанта.

#### Параметры

Параметры для метода указаны в таблице 9.

Таблица 9 — Перечень доступных параметров для обновления конфигурации SMTP

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

#### Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
smtp_conf/<TENANT_NAME>" \  
-H 'accept: application/json' \  
-H 'Content-Type: application/json' \  
-d '{  
  "cert": "string",  
  "config": {  
    "can_esia_auth": 0,  
    "esia_cert": "string",  
    "esia_client_id": "string",  
    "esia_expire_ts": 0,
```

```
"esia_key": "string",
"smtp_login": "string",
"smtp_port": 0,
"support_esia_auth": 0
},
"key": "string"
}'
-H 'Authorization: <TOKEN>'
```

## Ответ

– 200 — Success — Успешно. Пример вывода:

```
{
"response": {
"default_domain": {
"enc_ssl_key": "string",
"name": "string",
"ssl_cert": "string"},
"id": "string",
"name": "string",
"props": {
"esia": {
"can_esia_auth": 0,
"esia_cert": "string",
"esia_client_id": "string",
"esia_expire_ts": 0,
"esia_key": "string",
"support_esia_auth": 0
},
"support": {
"email": "string",
"phone": "string",
"smtp": {
"address": "string",
"enc_password": "string",
"login": "string",
"port": 0,
"ssl": 0
}
}
},
"success": "string",
"support_domains": [
"string"
]
},
"success": "string"
}
```



- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.3.4 Удаление значений параметров конфигурации SMTP

### Метод и конечная точка

DELETE/smtp\_conf/{tenant\_name}

### Описание

Удаляет значения параметров конфигурации SMTP тенанта.

### Параметры

Параметры для метода указаны в таблице 10.

Таблица 10— Перечень доступных параметров для удаления параметров SMTP

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
smtp_conf/<TENANT_NAME>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где <TENANT\_NAME> — имя тенанта.

### Ответ

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4 Работа с тенантами

### 4.4.1 Перечень тенантов

#### Метод и конечная точка

GET/tenants

#### Описание

Выводит перечень тенантов.

#### Параметры

Параметры для метода указаны в таблице 11.

Таблица 11 — Перечень доступных параметров для работы с тенантами

Название	Обязательный	Тип данных	Описание
query	нет	string	Если этот параметр установлен, то поиск осуществляется в соответствии со значением, если параметр не установлен, выводится полный список всех тенантов
page	нет	string	Параметр для разделения вывода на страницы. Размер страницы — 50 строк

#### Запрос

```
curl -X 'GET' \  
"https://admin[<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants?query=<QUERY>?page=<PAGE>' \  
<H 'accept: application/json' \  
<H 'Authorization: <TOKEN>'
```

Где:

- <QUERY> — параметр условий поиска;
- <PAGE> — параметр, задающий разделение вывода на страницы.

## Ответ

Возможные варианты ответов:

- 200 — Success — Success — Успешно. Пример вывода:

```
{
"tenants": [
  {
    "autoversion_settings": {
      "autoversion_time": 0,
      "enable_autoversion": 0,
      "max_autoversions": 0
    },
    "default_domain": "string",
    "domains": [
      "string"
    ],
    "enable_recovery_email": 0,
    "enabled": true,
    "enabled_users_count": 0,
    "excluded_locales": true,
    "history_settings": {
      "has_history_events": true,
      "history_enabled": true,
      "history_events_max_count": 0,
      "history_expiration_period": 0,
      "history_strategy": "string"
    },
    "id": "string",
    "lang": "string",
    "max_users": 0,
    "name": "string",
    "passwordPolicies": {
      "length": 0,
      "max_keep_passwords": 0,
      "min_digits": 0,
      "min_lower_case": 0,
      "min_upper_case": 0,
      "password_exp_time": 0,
      "special_characters": 0
    },
    "public_link_settings": {
      "public_link_access_time_without_password": 0,
      "public_link_corp_folders_enabled": true,
      "public_link_enabled": true,
      "public_link_password_required": true,
      "public_link_threshold_attempts_number": 0
    },
    "security_log_settings": {
      "is_protect_data_in_security_log": true,
      "is_security_log": true,
      "siem_host": "string",
      "siem_port": "string",
```

```

        "siem_protocol": "string"
    },
    "users_count": 0
}
]
}

```

## 4.4.2 Создание нового тенанта

### Метод и конечная точка

POST/tenants

### Описание

Создает новый тенант с заданными параметрами.

### Параметры

Параметры для метода указаны в таблице 12.

Таблица 12 — Перечень доступных параметров для создания тенанта

Название	Обязательный	Тип данных	Описание
name	да	string	Имя нового тенанта
default_domain	да	string	Имя домена по умолчанию
admin_password	да	string	Пароль администратора
admin_recovery_email	да	string	Адрес электронной почты для восстановления пароля
max_users	нет	string	Максимальное количество пользователей тенанта
admin_username	нет	string	Имя пользователя для учетной записи администратора, по умолчанию admin@{default_domain}
quota_per_user	нет	integer	Допустимое количество байтов

Название	Обязательный	Тип данных	Описание
			(квота) для документов пользователя. По умолчанию 1 ГБ.
tenant_quota	нет	integer	Количество байтов, разрешенное для документов пользователей в этом тенанте
tenant_corp_quota	нет	integer	Допустимое количество байтов для корпоративных документов в этом тенанте
autoversion_settings	нет	string	Настройка автоматического версионирования в тенанте
lang	нет	string	Язык по умолчанию
smtp_config	нет	string	Строка JSON со значениями параметров smtp и esia

## Запрос

```
curl -X 'POST' \
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/tenants?\
name=<TENANT_NAME>&default_domain=<DEFAULT_DOMAIN_NAME>&\
admin_password=<ADMIN_PASSWORD>&admin_recovery_email=<ADMIN_RECOVERY_EMAIL>&\
max_user_count=<MAX_USER_COUNT>" \
-H 'accept: application/json' \
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя нового тенанта;
- <DEFAULT\_DOMAIN\_NAME> — имя домена по умолчанию;
- <ADMIN\_PASSWORD> — пароль администратора;
- <ADMIN\_RECOVERY\_EMAIL> — адрес электронной почты для восстановления пароля;
- <MAX\_USER\_COUNT> — максимальное количество пользователей тенанта, значение по умолчанию — 1000.

## Ответ

Возможные варианты ответов:

- 200 — Success — Success — Успешно. Пример вывода:

```
{
  "autoversion_settings": {
    "autoversion_time": 0,
    "enable_autoversion": 0,
    "max_autoversions": 0
  },
  "default_domain": "string",
  "domains": [
    "string"
  ],
  "enable_recovery_email": 0,
  "enabled": true,
  "enabled_users_count": 0,
  "excluded_locales": true,
  "history_settings": {
    "has_history_events": true,
    "history_enabled": true,
    "history_events_max_count": 0,
    "history_expiration_period": 0,
    "history_strategy": "string"
  },
  "id": "string",
  "lang": "string",
  "max_users": 0,
  "name": "string",
  "passwordPolicies": {
    "length": 0,
    "max_keep_passwords": 0,
    "min_digits": 0,
    "min_lower_case": 0,
    "min_upper_case": 0,
    "password_exp_time": 0,
    "special_characters": 0
  },
  "public_link_settings": {
    "public_link_access_time_without_password": 0,
    "public_link_corp_folders_enabled": true,
    "public_link_enabled": true,
    "public_link_password_required": true,
    "public_link_threshold_attempts_number": 0
  },
  "security_log_settings": {
    "is_protect_data_in_security_log": true,
    "is_security_log": true,
    "siem_host": "string",
    "siem_port": "string",
    "siem_protocol": "string"
  },
}
```

```
    "users_count": 0  
  }
```

- 400 — Tenant name or default domain contains space symbols — Имя тенанта или домена по умолчанию содержит символы пробела. Пример вывода:

```
{  
  "message": "string"  
}
```

- 409 — Tenant or domain already exists — Тенант или домен уже существуют. Пример вывода:

```
{  
  "message": "string"  
}
```

### 4.4.3 Чтение параметров тенанта

#### Метод и конечная точка

GET/tenants/{tenant\_name}

#### Описание

Выводит перечень параметров тенанта.

#### Параметры

Параметры для метода указаны в таблице 13.

Таблица 13 — Перечень доступных параметров для работы с тенантом

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

#### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где <TENANT\_NAME> — имя тенанта.



## Ответ

Возможные варианты ответов:

– 200 — Success — Success — Успешно. Пример вывода:

```
{
  "autoversion_settings": {
    "autoversion_time": 0,
    "enable_autoversion": 0,
    "max_autoversions": 0
  },
  "default_domain": "string",
  "domains": [
    "string"
  ],
  "enable_recovery_email": 0,
  "enabled": true,
  "enabled_users_count": 0,
  "excluded_locales": true,
  "history_settings": {
    "has_history_events": true,
    "history_enabled": true,
    "history_events_max_count": 0,
    "history_expiration_period": 0,
    "history_strategy": "string"
  },
  "id": "string",
  "lang": "string",
  "max_users": 0,
  "name": "string",
  "passwordPolicies": {
    "length": 0,
    "max_keep_passwords": 0,
    "min_digits": 0,
    "min_lower_case": 0,
    "min_upper_case": 0,
    "password_exp_time": 0,
    "special_characters": 0
  },
  "public_link_settings": {
    "public_link_access_time_without_password": 0,
    "public_link_corp_folders_enabled": true,
    "public_link_enabled": true,
    "public_link_password_required": true,
    "public_link_threshold_attempts_number": 0
  },
  "security_log_settings": {
    "is_protect_data_in_security_log": true,
    "is_security_log": true,
    "siem_host": "string",
    "siem_port": "string",
    "siem_protocol": "string"
  },
}
```

```
    "users_count": 0  
  }
```

- 401 — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.4 Деактивация тенанта

### Метод и конечная точка

DELETE/tenants/{tenant\_name}

### Описание

Отключает тенант.

### Параметры

Параметры для метода указаны в таблице 14.

Таблица 14 — Перечень доступных параметров для работы с отключением тенанта

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где <TENANT\_NAME> — имя тенанта.

## Ответ

Возможные варианты ответов:

– 200 — Success — Success — Успешно. Пример вывода:

```
{
  "autoversion_settings": {
    "autoversion_time": 0,
    "enable_autoversion": 0,
    "max_autoversions": 0
  },
  "default_domain": "string",
  "domains": [
    "string"
  ],
  "enable_recovery_email": 0,
  "enabled": true,
  "enabled_users_count": 0,
  "excluded_locales": true,
  "history_settings": {
    "has_history_events": true,
    "history_enabled": true,
    "history_events_max_count": 0,
    "history_expiration_period": 0,
    "history_strategy": "string"
  },
  "id": "string",
  "lang": "string",
  "max_users": 0,
  "name": "string",
  "passwordPolicies": {
    "length": 0,
    "max_keep_passwords": 0,
    "min_digits": 0,
    "min_lower_case": 0,
    "min_upper_case": 0,
    "password_exp_time": 0,
    "special_characters": 0
  },
  "public_link_settings": {
    "public_link_access_time_without_password": 0,
    "public_link_corp_folders_enabled": true,
    "public_link_enabled": true,
    "public_link_password_required": true,
    "public_link_threshold_attempts_number": 0
  },
  "security_log_settings": {
    "is_protect_data_in_security_log": true,
    "is_security_log": true,
    "siem_host": "string",
    "siem_port": "string",
    "siem_protocol": "string"
  },
  "users_count": 0
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.5 Удаление тенанта

Опция доступна только при непосредственном посещении сервера, где развернут сервис Euclid. Рекомендации:

- убедитесь в необходимости удаления тенанта, либо воспользуйтесь функцией отключения тенанта;
- производите удаление в часы, когда основная активность отсутствует. Удаление тенанта в рабочие часы может вернуть ошибку;
- обратите внимание на регистр в имени удаляемого тенанта.

Для удаления тенанта выполните шаги:

1. Зайдите на сервер, где развёрнут сервис Euclid.
2. Проверьте существование сервиса командой `docker ps | grep euclid`.
3. Для удаления одного тенанта — выполните команду:

```
docker exec -i $(docker ps -qf name=euclid) python3.11 delete_tenant/main.py \  
--tenant_name <TENANT_NAME>
```

где `<TENANT_NAME>` — имя удаляемого тенанта;

4. Для удаления нескольких тенантов:

- последовательно выполните команду для каждого удаляемого тенанта;
- для удаления "всех тенантов, КРОМЕ ..." выполните команду:

```
docker exec -i $(docker ps -qf name=euclid) \  
python3.11 delete_tenant/main.py --all_except  
<TENANT_NAME_1>,<TENANT_NAME_2>,...,<TENANT_NAME_N>
```

Где `<TENANT_NAME_1>`, `<TENANT_NAME_2>`, `<TENANT_NAME_N>` — имена тенантов, которые НЕ будут удалены.

## 4.4.6 Чтение установленных параметров настроек тенанта

### Метод и конечная точка

GET/tenants/{tenant\_name}/common\_settings

### Описание

Выводит установленные значения параметров настроек тенанта.

### Параметры

Перечень параметров представлен в таблице 15.

Таблица 15 — Перечень параметров для чтения установленных настроек тенанта

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/common_settings" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где <TENANT\_NAME> — имя тенанта.

### Ответ

Возможные варианты ответов:

- 200 — Success — Success — Успешно. Пример вывода:

```
{  
  "ctypes_prohibition_enabled": true,  
  "enable_recovery_email": true,  
  "excluded_locales": true,  
  "id": "string",  
  "lang": "string",  
  "max_users": 0,  
  "name": "string"  
}
```

- 401 — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.7 Изменение значений параметров настроек тенанта

### Метод и конечная точка

PUT/tenants/{tenant\_name}/common\_settings

### Описание

Отправляет обновление значений параметров настроек тенанта. Метод может одновременно обрабатывать все указанные в запросе параметры.

### Параметры

Перечень параметров представлен в таблице 16.

Таблица 16 — Перечень параметров для изменения значений параметров тенанта

Название	Обязательный	Тип данных	Описание
max_users	нет	integer	Максимальное количество пользователей тенанта
enable_recovery_email	нет	string	Включить/выключить recovery_email
excluded_locales	нет	string	Включить/выключить excluded_locales
ctypes_prohibition_enabled	нет	string	Включить/выключить ctypes_prohibition
lang	нет	string	Изменить основной язык тенанта
autolock_users_enabled	нет	string	Включить/выключить автоматическую блокировку пользователей в тенанте

Название	Обязательный	Тип данных	Описание
users_session_duration	нет	string	Длительность пользовательской сессии в тенанте
signature_enabled	нет	string	Включить/выключить цифровую подпись для пользователя в тенанте
security_notification_enabled	нет	string	Включить/выключить уведомления безопасности в тенанте
files_notifications_enabled	нет	string	Включить/выключить уведомления о размере файлов в тенанте
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
\tenants/<TENANT_NAME>/common_settings?max_users=<MAX_USERS_COUNT>&\  
lang=<LANGUAGE>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <MAX\_USERS\_COUNT> — максимальное количество пользователей в тенанте;
- <LANGUAGE> — основной язык тенанта.

## Ответ

Возможные варианты ответов:

- 200 — Success — Success — Успешно. Пример вывода:

```
{  
  "ctypes_prohibition_enabled": true,  
  "enable_recovery_email": true,  
  "excluded_locales": true,  
  "id": "string",  
  "lang": "string",  
  "max_users": 0,  
  "name": "string"  
}
```

- 401 — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{
  "message": "string"
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.4.8 Чтение установленных параметров работы с событиями по объектам

### Метод и конечная точка

GET/tenants/{tenant\_name}/history\_settings

### Описание

Выводит установленные значения параметров работы с событиями по объектам.

### Параметры

Перечень параметров представлен в таблице 17.

Таблица 17 — Перечень параметров для чтения параметров работы с событиями

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\
tenants/<TENANT_NAME>/history_settings" \
-H 'accept: application/json' \
-H 'Authorization: <TOKEN>'
```

где <TENANT\_NAME> — имя настраиваемого тенанта.

### Ответ

Возможные варианты ответов:

- 200 — Success — Success — Успешно. Пример вывода:



```
{
  "history_settings": {
    "has_history_events": true,
    "history_enabled": true,
    "history_events_max_count": 0,
    "history_expiration_period": 0,
    "history_strategy": "string"
  },
  "id": "string",
  "name": "string"
}
```

- 401 — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{
  "message": "string"
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.4.9 Изменение параметров работы с событиями по объектам

### Метод и конечная точка

PUT/tenants/{tenant\_name}/history\_settings

### Описание

Отправляет обновление значений параметров работы с событиями по объектам.

### Параметры

Перечень параметров представлен в таблице 18.

Таблица 18 — Перечень параметров для изменения работы с событиями по объектам

Название	Обязательный	Тип данных	Описание
history_settings	нет	string	Параметр для настройки работы с событиями по объектам
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/tenants/<TENANT_NAME>" \  
-d history_settings='{ "history_enabled": false, "history_events_max_count": 100, \  
\  
"history_expiration_period": 31536000 }' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- `<TENANT_NAME>` — имя настраиваемого тенанта;
- `history_settings` — настройки для работы с событиями по объектам в тенанте, например:
  - `"history_enabled": false` — включение/отключение возможности регистрировать и показывать события по объектам; возможные значения `true` или `false`, по умолчанию `false`;
  - `"history_events_max_count": 100` — максимальное количество регистрируемых и показываемых событий по одному объекту; значение параметра — целое число (`int`), по умолчанию `100`;
  - `"history_expiration_period": 31536000` — максимальный период в секундах, за который показываются события по объекту; значение параметра — целое число (`int`), по умолчанию `31536000`.

## Ответ

Возможные варианты ответов:

- `200` — Success — Success — Успешно. Пример вывода:

```
{  
  "history_settings": {  
    "has_history_events": true,  
    "history_enabled": true,  
    "history_events_max_count": 0,  
    "history_expiration_period": 0,  
    "history_strategy": "string"  
  },  
  "id": "string",  
  "name": "string"  
}
```

- 401 — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.10 Изменение параметров журнала безопасности

### Метод и конечная точка

PUT/tenants/{tenant\_name}/audit\_settings

### Описание

Позволяет изменить параметры журнала безопасности (security log) и настроить взаимодействие с SIEM-системой.

### Параметры

Перечень параметров представлен в таблице 19.

Таблица 19 — Перечень параметров для изменения работы с журналом безопасности

Название	Обязательный	Тип данных	Описание
security_log_settings	нет	string	Параметр для настройки работы с журналом безопасности
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'PUT' \  
"https://admin[<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/audit_settings\  
-d security_log_settings='{ "is_protect_data_in_security_log": true, \  
"is_security_log": true, "siem_host": <HOST>, "siem_port": <PORT>, \  
'
```

```
"siem_protocol": <PROTOCOL>}" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- `<TENANT_NAME>` — имя настраиваемого тенанта;
- `security_log_settings` — настройки для работы с событиями по объектам в тенанте, например:
  - `"is_protect_data_in_security_log": true` — включение/отключение защищенного журнала безопасности; возможные значения true или false, по умолчанию true;
  - `"is_security_log": true` — включение/отключение журнала безопасности; возможные значения true или false, по умолчанию true;
  - `"siem_host": <HOST>` — адрес хоста подключаемой SIEM-системы;
  - `"siem_host": <PORT>` — порт подключаемой SIEM-системы;
  - `"siem_protocol": <PROTOCOL>` — протокол, по которому подключается SIEM-система.

## Ответ

Возможные варианты ответов:

- `200` — Success — Успешно. Пример вывода:

```
{  
  "id": "string",  
  "name": "string"  
  "security_log_settings": {  
    "is_protect_data_in_security_log": true,  
    "is_security_log": true,  
    "siem_host": string,  
    "siem_port": string,  
    "siem_protocol": "string"  
  },  
}
```

- `401` — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

- `404` — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.11 Изменение параметров автоверсионирования

### Метод и конечная точка

PUT/tenants/{tenant\_name}/autoversion\_settings

### Описание

Позволяет изменить параметры автоверсионирования для тенанта.

### Параметры

Перечень параметров представлен в таблице 20.

Таблица 20 — Перечень параметров для изменения работы с автоверсионированием

Название	Обязательный	Тип данных	Описание
autoversion_settings	нет	string	Параметр для настройки работы с автоверсионированием
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/autoversion_settings\  
-d autoversion_settings='{ "enable_autoversion": 0, \  
"autoversion_time": 0, "max_autoversions": 0}' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя настраиваемого тенанта;
- autoversion\_settings — настройки для работы с автоверсионированием в тенанте, например:
  - "enable\_autoversion": 0 — включение/отключение автоверсионирования; возможные значения 1 или 0, по умолчанию 0 (выключено);
  - "autoversion\_time": 0 — частота выполнения автоверсионирования в миллисекундах;

- `"max_autoversions": 0` — глубина выполняемого версионирования.

## Ответ

Возможные варианты ответов:

- `200` — Success — Успешно. Пример вывода:

```
{
  "id": "string",
  "name": "string"
  "autoversion_settings": {
    "autoversion_time": 0,
    "enable_autoversion": 0,
    "max_autoversions": 0
  },
}
```

- `401` — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{
  "message": "string"
}
```

- `404` — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.4.12 Смена имени организации в тенанте

### Метод и конечная точка

PUT/tenants/{tenant\_name}/organisation\_name

### Описание

Позволяет изменить имя организации в тенанте.

### Параметры

Перечень параметров представлен в таблице 21.

Таблица 21 — Перечень параметров для изменения имени организации в тенанте

Название	Обязательный	Тип данных	Описание
name	да	string	Новое имя организации
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/organisation_name?name=<NAME>' \  
-H 'accept: application/json" \  
-H 'Authorization: <TOKEN>'
```

где:

- **<TENANT\_NAME>** — имя настраиваемого тенанта;
- **<NAME>** — новое имя для организации в тенанте.

### Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- **401** — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

- **404** — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.13 Чтение парольных политик в тенанте

### Метод и конечная точка

GET/tenants/{tenant\_name}/password\_policies

### Описание

Позволяет просмотреть параметры парольных политик тенанта.

### Параметры

Перечень параметров представлен в таблице 22.

Таблица 22 — Перечень параметров для изменения имени организации в тенанте

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/password_policies" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:



- 200 — Success — Успешно. Пример вывода:

```
{
  "id": "string",
  "name": "string"
  "password_policies": {
    "length": 0,
    "max_keep_passwords": 0,
    "min_digits": 0,
    "min_lower_case": 0,
    "min_upper_case": 0,
    "password_exp_time": 0,
    "special_characters": 0,
  },
}
```

- 401 — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{
  "message": "string"
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.4.14 Изменение парольных политик в тенанте

### Метод и конечная точка

PUT/tenants/{tenant\_name}/password\_policies

### Описание

Позволяет просмотреть параметры парольных политик тенанта.

### Параметры

Перечень параметров представлен в таблице 23.

Таблица 23 — Перечень параметров для изменения имени организации в тенанте

Название	Обязательный	Тип данных	Описание
password_policies	нет	string	Настройки для парольной политики тенанта
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/password_policies" \  
-d password_policies='{ "length": 0, "max_keep_passwords": 0, \  
"min_digits": 0, "min_lower_case": 0, "min_upper_case": true, \  
"password_exp_time": 0, "special_characters": 0}' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя настраиваемого тенанта;
- password\_policies — настройки для работы с капчей в тенанте:
  - "length": 0 — длина пароля;
  - "max\_keep\_passwords": 0 — количество уникальных паролей перед повторным использованием;
  - "min\_digits": 0 — минимальное количество цифр;
  - "min\_lower\_case": 0 — минимальное количество строчных букв;
  - "min\_upper\_case": true — минимальное количество прописных букв;
  - "password\_exp\_time": 0 — срок действия пароля в днях;
  - "special\_characters": 0 — минимальное количество спецсимволов.

## Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{
  "id": "string",
  "name": "string"
  "password_policies": {
    "length": 0,
    "max_keep_passwords": 0,
    "min_digits": 0,
    "min_lower_case": 0,
    "min_upper_case": 0,
    "password_exp_time": 0,
    "special_characters": 0,
  },
}
```

- **401** — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{
  "message": "string"
}
```

- **404** — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.4.15 Чтение параметров квот в тенанте

### Метод и конечная точка

GET/tenants/{tenant\_name}/quota

### Описание

Позволяет просмотреть параметры квот (обычной и для общих папок) в тенанте.

### Параметры

Перечень параметров представлен в таблице 24.

Таблица 24 — Перечень параметров для чтения параметров квот в тенанте

Название	Обязательный	Тип данных	Описание
with_used_bytes	нет	string	Параметр для просмотра использованного места в тенанте
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/quota?with_used_bytes=<USED_BYTES>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

## Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{  
  "reserved_bytes": 0,  
  "storage_space": 0,  
  "success": true,  
  "tenant_corp_quota": 0,  
  "tenant_quota": 0,  
  "tenant_used_bytes": 0,  
  "used_corp_bytes": 0,  
}
```

- **401** — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

- **404** — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.16 Изменение параметров квот в тенанте

### Метод и конечная точка

PUT/tenants/{tenant\_name}/quota

### Описание

Позволяет изменить размеры квот (обычной и для общих папок) в тенанте.

### Параметры

Перечень параметров представлен в таблице 25.

Таблица 25 — Перечень параметров для изменения размеров квот в тенанте

Название	Обязательный	Тип данных	Описание
tenant_corp_quota	нет	string	Новый размер квоты для общих папок в тенанте (в байтах)
tenant_quota	нет	string	Новый размер квоты для пользовательских файлов в тенанте (в байтах)
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/quota?tenant_corp_quota=<BYTES>&tenant_quota=<BYTES>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

— **200** — Success — Успешно. Пример вывода:

```
{  
  "tenant_corp_quota": 0,  
  "tenant_quota": 0,  
}
```

- **401** — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

- **404** — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.17 Чтение параметров двухфакторной авторизации в тенанте

### Метод и конечная точка

GET/tenants/{tenant\_name}/two\_fa\_settings

### Описание

Позволяет просмотреть параметры двухфакторной авторизации в тенанте.

### Параметры

Перечень параметров представлен в таблице 26.

Таблица 26 — Перечень параметров двухфакторной авторизации в тенанте

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/two_fa_settings' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{  
  "id": "string",
```

```
"name": "string"
"password_policies": {
  "length": 0,
  "max_keep_passwords": 0,
  "min_digits": 0,
  "min_lower_case": 0,
  "min_upper_case": 0,
  "password_exp_time": 0,
  "special_characters": 0,
},
}
```

- **401** — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{
  "message": "string"
}
```

- **404** — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.4.18 Изменение параметров двухфакторной авторизации в тенанте

### Метод и конечная точка

PUT/tenants/{tenant\_name}/two\_fa\_settings

### Описание

Позволяет изменить размеры параметры двухфакторной авторизации в тенанте.

### Параметры

Перечень параметров представлен в таблице 27.

Таблица 27 — Перечень параметров для изменения параметров двухфакторной авторизации в тенанте

Название	Обязательный	Тип данных	Описание
two_fa_settings	нет	string	Параметр для изменения парольных политик в тенанте
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'PUT' \  
  "https://admin[<ENV>].<DEFAULT_DOMAIN>/adminapi/\tenants/<TENANT_NAME>/two_fa_settings' \  
-d password_policies='{ "length": 0, "max_keep_passwords": 0, \  
"min_digits": 0, "min_lower_case": 0, "min_upper_case": true, \  
"password_exp_time": 0, "special_characters": 0}' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя настраиваемого тенанта;
- password\_policies — настройки для работы с двухфакторной авторизацией в тенанте:
  - "length": 0 — длина пароля;
  - "max\_keep\_passwords": 0 — количество уникальных паролей перед повторным использованием;
  - "min\_digits": 0 — минимальное количество цифр;
  - "min\_lower\_case": 0 — минимальное количество строчных букв;
  - "min\_upper\_case": true — минимальное количество прописных букв;
  - "password\_exp\_time": 0 — срок действия пароля в днях;
  - "special\_characters": 0 — минимальное количество спецсимволов.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "id": "string",  
  "name": "string"  
  "password_policies": {  
    "length": 0,  
    "max_keep_passwords": 0,  
    "min_digits": 0,  
    "min_lower_case": 0,  
    "min_upper_case": 0,  
    "password_exp_time": 0  
    "special_characters": 0,
```



```
},  
}
```

- 401 — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.19 Получить список запретов в тенанте

### Метод и конечная точка

GET/tenants/{tenant\_name}/permissions

### Описание

Выводит перечень запретов тенанта.

### Параметры

Перечень параметров представлен в таблице 28.

Таблица 28— Перечень параметров запретов тенанта

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/premission' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

**200** — Success — Успешно. Пример вывода:

```
{  
  "rules": {  
    "attributes": "string",  
    "code": "string",  
    "is_enabled": true  
  }  
}
```

## 4.4.20 Включить запрет определенного типа в тенанте

### Метод и конечная точка

POST/tenants/{tenant\_name}/permissions/code/{code}/enable

## Описание

Включить запрет по ID.

## Параметры

Перечень параметров представлен в таблице 29.

Таблица 29— Перечень параметров включения запрета

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
code	да	string	ID запрета

## Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/code/<CODE>/enable' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

## Ответ

Возможные варианты ответов:

— **200** — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

### 4.4.21 Выключить запрет определенного типа в тенанте

#### Метод и конечная точка

POST/tenants/{tenant\_name}/permissions/code/{code}/disable

#### Описание

Выключить запрет по ID.

#### Параметры

Перечень параметров представлен в таблице 30.

Таблица 30 — Перечень параметров отключения запрета

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
code	да	string	ID запрета

## Запрос

```
curl -X 'POST' \
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\
tenants/<TENANT_NAME>/code/<CODE>/disable" \
-H 'accept: application/json' \
-H 'Authorization: <TOKEN>'
```

## Ответ

Возможные варианты ответов:

— 200 — Success — Успешно. Пример вывода:

```
{
  "message": "string"
}
```

### 4.4.22 Получить список запретов по типу в тенанте

#### Метод и конечная точка

GET/tenants/{tenant\_name}/permissions/code/{code}/items

#### Описание

Получить список запретов в этом тенанте, где code это тип запрета (например на скачивание и печать).

#### Параметры

Перечень параметров представлен в таблице 31.

Таблица 31 — Перечень параметров списка запретов по типу

Название	Обязательный	Тип данных	Описание
context_type	да	string	Тип контекста (пользователь / группа)

Название	Обязательный	Тип данных	Описание
permission_type	да	string	Тип ограничения или исключения
order_by	нет	string	Кем создан заказ
order_type	нет	string	Тип заказа
tenant_name	да	string	Имя тенанта
code	да	string	ID запрета

## Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/tenants/\<br><TENANT_NAME>/permissions/code/<CODE>/items?context_type=<CONTEXT_TYPE>\<br>permission_type=<PERMISSION_TYPE>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "has_next": true,  
  "rules": {  
    "attributes": "string",  
    "code": "string",  
    "created_at": "string",  
    "has_children": true,  
    "holder": {  
      "reference": "string",  
      "title": "string",  
      "type": "string"  
    },  
    "owner": {  
      "reference": "string",  
      "title": "string",  
      "type": "string"  
    },  
    "reference": "string",  
    "type": "string"  
  }  
}
```

## 4.4.23 Добавить запрет или исключение в тенанте

### Метод и конечная точка

POST/tenants/{tenant\_name}/permissions/{context\_type}/{context\_reference}

### Описание

Добавить новый запрет или исключение для тенанта.

### Параметры

Перечень параметров представлен в таблице 32.

Таблица 32 — Перечень параметров добавления запрета или исключения

Название	Обязательный	Тип данных	Описание
rule	да	string	Правило
context_type	да	string	Тип контекста (пользователь / группа)
tenant_name	да	string	Имя тенанта
context_reference	да	string	Образец контекста

### Запрос

```
curl -X 'POST' \  
'https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/tenants/  
<TENANT_NAME>/permissions/<CONTEXT_TYPE>/<CONTEXT_REFERENCE>?' \  
rule_reference=<RULE_REFERENCE>' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.4.24 Удалить запрет или исключение в тенанте

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/permissions/{context\_type}/{context\_reference}

### Описание

Удалить запрет или исключение для тенанта.

### Параметры

Перечень параметров представлен в таблице 33.

Таблица 33 — Перечень параметров удаления запрета или исключения

Название	Обязательный	Тип данных	Описание
rule_reference	да	string	Образец созданного правила
context_type	да	string	Тип контекста (пользователь / группа)
tenant_name	да	string	Имя тенанта
context_reference	да	string	Образец контекста

### Запрос

```
curl -X 'DELETE' \  
'https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/permissions/<CONTEXT_TYPE>/\  
<CONTEXT_REFERENCE>?rule=<RULE>' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.5 Работа с проверочным кодом (CAPTCHA)

### 4.5.1 Установка проверочного кода

#### Метод и конечная точка

POST/tenants/{tenant\_name}/captcha

#### Описание

Позволяет установить CAPTCHA для тенанта.

#### Параметры

Перечень параметров представлен в таблице 34.

Таблица 34 — Перечень параметров для установки CAPTCHA в тенанте

Название	Обязательный	Тип данных	Описание
key	нет	string	Ключ проверочного кода (капча)
tenant_name	да	string	Имя тенанта

#### Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/captcha?key=<KEY>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

#### Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- **400** — Missing parameter — Не указан параметр авторизации. Пример вывода:

```
{  
  "message": "string"  
}
```

- **404** — Tenant not found — Тенант не найден. Пример вывода:



```
{  
  "message": "string"  
}
```

## 4.5.2 Просмотр настроек проверочного кода

### Метод и конечная точка

GET/tenants/{tenant\_name}/captcha\_settings

### Описание

Позволяет получить список настроек для капчи тенанта.

### Параметры

Перечень параметров представлен в таблице 35.

Таблица 35 — Перечень параметров для просмотра настроек CAPTCHA

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/captcha_settings' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

– **200** — Success — Успешно. Пример вывода:

```
{  
  "id": "string",  
  "name": "string"  
  "captcha_settings": {  
    "bruteForceProtected": true,  
    "failureFactor": 0,  
    "maxDeltaTimeSeconds": 0,  
    "minimumQuickLoginWaitSeconds": 0,  
    "permanentLockout": true,  
    "quickLoginCheckMilliseconds": 0  
    "waitIncrementSeconds": 0,  
  }  
}
```

```
},  
}
```

- 400 — Auth header is not provided or token is invalid — Не указан параметр авторизации или неверный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.5.3 Изменение настроек проверочного кода

### Метод и конечная точка

PUT/tenants/{tenant\_name}/captcha\_settings

### Описание

Позволяет изменить настройки для капчи тенанта.

### Параметры

Перечень параметров представлен в таблице 36.

Таблица 36 — Перечень параметров для изменения настроек капчи

Название	Обязательный	Тип данных	Описание
captcha_settings	нет	string	Параметр для изменения настроек капчи тенанта
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/captcha_settings\  
-d captcha_settings='{ "bruteForceProtected": true, \  
"failureFactor": 0, "maxDeltaTimeSeconds": 0,\  
"minimumQuickLoginWaitSeconds": 0, "permanentLockout": true, \  
"quickLoginCheckMilliseconds": 0, "waitIncrementSeconds": 0}' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- `<TENANT_NAME>` — имя настраиваемого тенанта;
- `captcha_settings` — настройки для работы с капчей в тенанте:
  - `"bruteForceProtected": true` — включение/отключение защиты при подозрительной активности; возможные значения true или false, по умолчанию true (включено);
  - `"failureFactor": 0` — количество неверных попыток ввода пароля;
  - `"maxDeltaTimeSeconds": 0` — время до сброса счетчика попыток;
  - `"minimumQuickLoginWaitSeconds": 0` — время показа капчи при превышении лимита попыток;
  - `"permanentLockout": true` — включение/отключение блокировки учетной записи пользователя при достижении максимального количества попыток ввода капчи; возможные значения true или false, по умолчанию true (включено);
  - `"quickLoginCheckMilliseconds": 0` — минимальный интервал между попытками ввода пароля;
  - `"waitIncrementSeconds": 0` — длительность показа капчи при более частых попытках.

## Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{
  "id": "string",
  "name": "string"
  "captcha_settings": {
    "bruteForceProtected": true,
    "failureFactor": 0,
    "maxDeltaTimeSeconds": 0,
    "minimumQuickLoginWaitSeconds": 0,
    "permanentLockout": true,
    "quickLoginCheckMilliseconds": 0
    "waitIncrementSeconds": 0,
  },
}
```

- **400** — Auth header is not provided or token is invalid — Не указан параметр авторизации или неверный токен. Пример вывода:

```
{
  "message": "string"
}
```

— 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.6 Работа с общими папками

### 4.6.1 Получение списка общих папок в тенанте

#### Метод и конечная точка

GET/tenants/{tenant\_name}/corporate

#### Описание

Позволяет получить список общих папок в тенанте.

#### Параметры

Перечень параметров представлен в таблице 37.

Таблица 37 — Перечень параметров получения списка общих папок тенанта

Название	Обязательный	Тип данных	Описание
offset	нет	string	Параметр настройки выдачи результатов поиска
limit	нет	string	Ограничения поиска
query	да	string	Поисковый запрос, в соответствии с которым выполняется поиск (если параметр не установлен, выводится полный список папок)
sort_order	нет	string	Порядок сортировки результатов выдачи, может принимать значения asc или desc
sort_by	нет	string	Сортировать по параметрам, допустимые поля сортировки: admin, size, id, name.
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/corporate?query=<QUERY>&offset=<OFFSET>&\  
limit=<LIMIT>&sort_order=<SORT_ORDER>&sort_by=<SORT_BY>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>
```

где:

- **<TENANT\_NAME>** — имя тенанта;
- **<QUERY>** — поисковый запрос. Если этот параметр установлен, то поиск осуществляется в соответствии со значением. Если параметр не установлен, выводится полный список всех групп;
- **<LIMIT>** — ограничения поиска;
- **<OFFSET>** — параметр настройки выдачи результатов поиска;
- **<SORT\_ORDER>** — порядок сортировки результатов поиска;
- **<SORT\_BY>** — сортировка по указанному параметру.

## Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{  
  "corp_folders": [  
    {  
      "admin_email": "string",  
      "admin_first_name": "string",  
      "admin_id": "string",  
      "admin_last_name": "string",  
      "admin_username": "string",  
      "id": 0,  
      "name": "string",  
      "size": 0  
    },  
  ],  
}
```

- **404** — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.6.2 Создание общей папки в тенанте

### Метод и конечная точка

POST/tenants/{tenant\_name}/corporate

### Описание

Позволяет создать общую папку в тенанте.

### Параметры

Перечень параметров представлен в таблице 38.

Таблица 38 — Перечень параметров для создания общей папки в тенанте

Название	Обязательный	Тип данных	Описание
name	нет	string	Имя создаваемой папки
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/corporate&name=<NAME>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <NAME> — имя создаваемой папки.

### Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- **403** — Corpfolder with same name exists — Общая папка с таким именем уже существует. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.6.3 Получение информации об общей папке

### Метод и конечная точка

GET/tenants/{tenant\_name}/corporate/{id}

### Описание

Позволяет получить список общих папок в тенанте.

### Параметры

Перечень параметров представлен в таблице 39.

Таблица 39 — Перечень параметров получения информации об общей папке

Название	Обязательный	Тип данных	Описание
id	да	string	Идентификатор папки
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/corporate/<ID>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификатор искомой папки.

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:



```
{
  "admin_email": "string",
  "admin_first_name": "string",
  "admin_id": "string",
  "admin_last_name": "string",
  "admin_username": "string",
  "id": 0,
  "name": "string"
  "size": 0
}
```

- **401** — Unauthorized — Неверный токен аутентификации. Пример вывода:

```
{
  "message": "string"
}
```

- **404** — Corpfolder or tenant not found — Общая папка или тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.6.4 Изменение атрибутов общей папки

### Метод и конечная точка

PUT/tenants/{tenant\_name}/corporate/{id}

### Описание

Позволяет изменить атрибуты общей папки в тенанте.

### Параметры

Перечень параметров представлен в таблице 40.

Таблица 40 — Перечень параметров изменения атрибутов общей папки

Название	Обязательный	Тип данных	Описание
new_name	нет	string	Новое имя изменяемой папки
admin_username	нет	string	Новый владелец изменяемой папки
tenant_name	да	string	Имя тенанта

Название	Обязательный	Тип данных	Описание
id	да	string	Идентификатор изменяемой папки

## Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/corporate/<ID>&new_name=<NEW_NAME>&  
admin_username=<NEW_OWNER>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификатор искомой папки;
- <NEW\_NAME> — новое имя изменяемой папки;
- <NEW\_OWNER> — имя нового администратора изменяемой папки.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- 400 — Admin with such admin\_username not found — администратор с таким именем пользователя не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

- 403 — Corp folder with same name exists — Общая папка с таким именем уже существует. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Corpfolder or tenant not found — Общая папка или тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.6.5 Удаление общей папки

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/corporate/{id}

### Описание

Позволяет удалить общую папку со всеми содержимым.

### Параметры

Перечень параметров представлен в таблице 41.

Таблица 41 — Перечень параметров для удаления общей папки

Название	Обязательный	Тип данных	Описание
id	да	string	Идентификатор удаляемой папки
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
\tenants/<TENANT_NAME>/corporate/<ID>' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификатор удаляемой папки.

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Corpfolder or tenant not found — Общая папка или тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.7 Работа с пользователями

### 4.7.1 Перечень пользователей

#### Метод и конечная точка

GET/tenants/{tenant\_name}/users

#### Описание

Выводит перечень пользователей.

#### Параметры

Параметры для метода указаны в таблице 42.

Таблица 42 — Перечень параметров для запроса перечня с пользователями

Название	Обязательный	Тип данных	Описание
query	нет	string	Поисковый запрос, в соответствии с которым выполняется поиск (если параметр не установлен, выводится полный список групп)
role	нет	string	Роль пользователя из перечня доступных ролей в текущем тенанте
enabled	нет	string	Если этот параметр указан, то пользователь будет разблокирован (сможет войти в систему), если он был заблокирован ранее
limit	нет	string	Ограничение поиска
offset	нет	string	параметр настройки выдачи результатов поиска
start_ctime	нет	string	Начало диапазона времени для фильтрации

Название	Обязательный	Тип данных	Описание
			пользователей, отметка времени unix в миллисекундах
end_ctime	нет	string	Окончание диапазона времени для фильтрации пользователей, отметка времени unix в миллисекундах
sort_field	нет	string	Допустимые поля сортировки: ctime, used_bytes, first_name, middle_name, position, last_name, username, email, password_expiration, enabled, role
sort_type	нет	string	Тип сортировки результатов выдачи, может принимать значения asc или `desc`
with_deleted	нет	string	Отображать удаленных пользователей, если значение параметра True
status	нет	string	Фильтрация по статусу; доступные значения active, blocked, deleted
with_prohibited_ctypes	нет	string	Запрос вывода пользователей с указанием запрещенных медиа типов
has_deleted_files	нет	string	Запрос вывода пользователей с удаленными файлами
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/users?query=<QUERY>&role=<ROLE>&  
enabled=<ENABLED>&limit=<LIMIT>&offset=<OFFSET>&start_ctime=<START_CTIME>&  
end_ctime=<END_CTIME>&sort_field=<SORT_FIELD>&  
sort_type=<SORT_TYPE>&with_deleted=<WITH_DELETED>&status=<STATUS>&  
with_prohibited_ctypes=<WITH_PROHIBITED_CTYPES>&has_deleted_files=<HAS_DELETED_FI  
LES>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <QUERY> — поисковый запрос. Если этот параметр установлен, то поиск осуществляется в соответствии со значением. Если параметр не установлен, выводится полный список всех групп;
- <ROLE> — роль пользователя из перечня доступных ролей в текущем тенанте;
- <ENABLED> — если этот параметр указан, то пользователь будет разблокирован (сможет войти в систему), если он был заблокирован ранее;
- <LIMIT> — ограничения поиска;
- <OFFSET> — параметр настройки выдачи результатов поиска;
- <START\_CTIME> — начало диапазона времени для фильтрации пользователей, отметка времени unix в миллисекундах;
- <END\_CTIME> — окончание диапазона времени для фильтрации пользователей, отметка времени unix в миллисекундах;
- <SORT\_FIELD> — допустимые поля сортировки;
- <SORT\_TYPE> — тип сортировки результатов выдачи;
- <WITH\_DELETED> — отображать удаленных пользователей, если значение параметра True;
- <STATUS> — фильтрация по статусу;
- <WITH\_PROHIBITED\_CTYPES> — запрос вывода пользователей с указанием запрещенных медиатипов;
- <HAS\_DELETED\_FILES> — запрос вывода пользователей с удаленными файлами.

## Ответ

Возможные варианты ответов:

— 200 — Success — Успешно. Пример вывода:

```
{
  "count": 0,
  "users": [
    {
      "ctime": "string",
      "domain": "string",
      "enabled": true,
      "except_ctype_prohibition": true,
      "first_name": "string",
      "id": "string",
      "is_deleted": true,
      "lang": "string",
      "last_name": "string",
      "middle_name": "string",
      "password": "string",
      "position": "string",
      "quota": 0,
      "recovery_email": "string",
      "roles": [
        {
          "clientRole": true,
          "composite": true,
          "containerId": "string",
          "description": "string",
          "id": "string",
          "name": "string"
        }
      ],
      "username": "string"
    }
  ],
  "users_count": 0
}
```

— 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```



## 4.7.2 Создание нового пользователя

### Метод и конечная точка

POST/tenants/{tenant\_name}/users

### Описание

Создать нового пользователя с заданными параметрами.

### Параметры

Параметры для метода указаны в таблице 43.

Таблица 43 — Перечень доступных параметров для создания нового пользователя

Название	Обязательный	Тип данных	Описание
username	да	string	При создании единичного пользователя обязательно укажите имя пользователя, пароль и адрес электронной почты для восстановления, в противном случае будет возвращен статус 400
password	да	string	Пароль нового пользователя
recovery_email	да	string	Адрес электронной почты для восстановления пароля
user_id	нет	string	Идентификационный номер пользователя, параметр используется LDAP-плагином (не вводить вручную)
role	нет	string	Роль пользователя из перечня доступных ролей в текущем тенанте

Название	Обязательный	Тип данных	Описание
email	да	string	Адрес электронной почты пользователя
quota	нет	string	Максимальный объем хранилища для пользователя, по умолчанию назначается в соответствии с настройками текущего тенанта
personal	да	string	Личные данные пользователя
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'POST' \  
"https://admin[<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/users?username=<USER_NAME>&password=<PASSWORD>&  
recovery_email=<RECOVERY_EMAIL>&email=<USER_EMAIL>&quota=<QUOTA>&  
role=<ROLE>" \  
--data-urlencode "personal={  
"first_name": "Фамилия", \  
"last_name": "Имя", \  
"middle_name": "Отчество"}\  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <USER\_NAME> — имя пользователя;
- <PASSWORD> — пароль пользователя;
- <RECOVERY\_EMAIL> — адрес электронной почты для восстановления пароля;
- <USER\_EMAIL> — адрес электронной почты пользователя;
- <QUOTA> — максимальный объем хранилища для пользователя. По умолчанию назначается в соответствии с настройками текущего тенанта;
- <ROLE> — роль пользователя из перечня доступных ролей в текущем тенанте;

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{
  "ctime": "string",
  "domain": "string",
  "enabled": true,
  "except_ctype_prohibition": true,
  "first_name": "string",
  "id": "string",
  "is_deleted": true,
  "lang": "string",
  "last_name": "string",
  "middle_name": "string",
  "password": "string",
  "position": "string",
  "quota": 0,
  "recovery_email": "string",
  "roles": [
    {
      "clientRole": true,
      "composite": true,
      "containerId": "string",
      "description": "string",
      "id": "string",
      "name": "string"
    }
  ],
  "username": "string"
}
```

- 400 — Bad username, password or other parameter, check error message — Имя пользователя, пароль или другой параметр не соответствует требованиям, проверьте сообщение об ошибке. Пример вывода:

```
{
  "message": "string"
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.7.3 Чтение информации о пользователе

### Метод и конечная точка

GET/tenants/{tenant\_name}/users/{id}

### Описание

Запрос установленных значений параметров пользователя.

## Параметры

Параметры для метода указаны в таблице 44.

Таблица 44 — Перечень доступных параметров для чтения информации о пользователях

	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

## Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/users/<ID>' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификационный номер пользователя.

## Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{  
  "ctime": "string",  
  "domain": "string",  
  "enabled": true,  
  "except_ctype_prohibition": true,  
  "first_name": "string",  
  "id": "string",  
  "is_deleted": true,  
  "lang": "string",  
  "last_name": "string",  
  "middle_name": "string",  
  "password": "string",  
  "position": "string",  
  "quota": 0,  
  "recovery_email": "string",  
  "roles": [  
    {  
      "clientRole": true,  
      "composite": true,  
      "containerId": "string",  
      "description": "string",
```

```

        "id": "string",
        "name": "string"
    },
    "username": "string"
}

```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```

{
  "message": "string"
}

```

## 4.7.4 Изменение информации о пользователе

### Метод и конечная точка

PUT/tenants/{tenant\_name}/users/{id}

### Описание

Отправляет обновление значений параметров пользователя.

### Параметры

Параметры для метода указаны в таблице 45.

Таблица 45 — Перечень доступных параметров для изменения информации о пользователях

Название	Обязательный	Тип данных	Описание
enabled	нет	string	Если этот параметр указан, то пользователь будет разблокирован (сможет войти в систему), если он был заблокирован ранее
reset_password	нет	string	Параметр может принимать 3 значения: with_email (пользователю будет отправлено электронное письмо с просьбой восстановить пароль), hard (необходимо ввести новое значение пароля, или

Название	Обязательный	Тип данных	Описание
			new_password), simply (срок действия пароля пользователя истечет и администратор сможет изменить пароль)
role	нет	string	Роль пользователя из перечня доступных ролей в текущем тенанте
quota	нет	string	Максимальный объем хранилища для пользователя. По умолчанию назначается в соответствии с настройками текущего тенанта
recovery_email	нет	string	Адрес электронной почты для восстановления пароля
except_ctype_prohibition	нет	string	Исключения для запрета загрузки медиатипов
personal	нет	string	Личные данные пользователя, такие как Имя, Фамилия и др.
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя, параметр используется LDAP-плагином. Не вводить вручную

## Запрос

```
curl -X 'PUT' \
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\
tenants/<TENANT_NAME>/users/<ID>?username=<USER_NAME>&password=<PASSWORD>&\
recovery_email=<RECOVERY_EMAIL>&email=<USER_EMAIL>&quota=<QUOTA>&role=<ROLE>" \
--data-urlencode "personal={\
```

```
"first_name": "Фамилия", \
"last_name": "Имя", \
"middle_name": "Отчество"} \
-H 'accept: application/json' \
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификационный номер пользователя;
- <USER\_NAME> — имя пользователя;
- <PASSWORD> — пароль пользователя;
- <RECOVERY\_EMAIL> — адрес электронной почты для восстановления пароля;
- <USER\_EMAIL> — адрес электронной почты пользователя;
- <QUOTA> — максимальный объем хранилища для пользователя. По умолчанию назначается в соответствии с настройками текущего тенанта;
- <ROLE> — роль пользователя из перечня доступных ролей в текущем тенанте;
- <PERSONAL> — личные данные пользователя, такие как Имя, Фамилия и др.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{
  "ctime": "string",
  "domain": "string",
  "enabled": true,
  "except_ctype_prohibition": true,
  "first_name": "string",
  "id": "string",
  "is_deleted": true,
  "lang": "string",
  "last_name": "string",
  "middle_name": "string",
  "password": "string",
  "position": "string",
  "quota": 0,
  "recovery_email": "string",
  "roles": [
    {
      "clientRole": true,
      "composite": true,
      "containerId": "string",
      "description": "string",
      "id": "string",
      "name": "string"
    }
  ],
}
```

```
"username": "string"  
}
```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.7.5 Удаление и блокировка пользователя

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/users/{id}

### Описание

Удалить или заблокировать пользователя.

### Параметры

Параметры для метода указаны в таблице 46.

Таблица 46 — Перечень доступных параметров для удаления пользователей

Название	Обязательный	Тип данных	Описание
delete_mode	нет	string	Определяет параметры ограничений работы пользователя. Основные значения: – disable, заблокировать пользователя; – reassign, переназначить все документы пользователю с предоставленным user_id и свободным именем пользователя; – keep_files, сохранить файлы и освободить имя пользователя; – full_delete, удалить все файлы и



Название	Обязательный	Тип данных	Описание
			освободить имя пользователя; – для всех остальных значений параметра — отключить пользователя
new_user_id	нет	string	Идентификационный номер нового администратора/владельца папки, указывается при значении параметра delete_mode == 'reassign'
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер папки

## Запрос

```
curl -X 'DELETE' \
  "https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\
  tenants/<TENANT_NAME>/users/<FOLDER_ID>?delete_mode=<DELETE_MODE>&\
  new_user_id=<NEW_USER_ID>' \
  -H 'accept: application/json' \
  -H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <FOLDER\_ID> — идентификационный номер папки;
- <DELETE\_MODE> — значение параметра, определяющего действия с пользователем;
- <NEW\_USER\_ID> — идентификационный номер нового администратора/владельца папки.

## Ответ

Возможные варианты ответов:

- 200 – Success – Успешно. Пример вывода:

```
{
  "message": "string"
}
```

## 4.7.6 Принудительная деавторизация пользователя

### Метод и конечная точка

POST/tenants/{tenant\_name}/users/{id}/logout

### Описание

Принудительно деавторизует пользователя из системы (делает log out).

### Параметры

Параметры для метода указаны в таблице 47.

Таблица 47 — Перечень доступных параметров для деавторизации пользователя

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

### Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/users/<ID>/logout" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификационный номер деавторизируемого пользователя.

### Ответ

Возможные варианты ответов:

- **200** — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- **401** — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.7.7 Чтение списка общих папок пользователя

Метод и конечная точка

GET/tenants/{tenant\_name}/users/{id}/corporate

### Описание

Позволяет получить список общих папок, к которым пользователь имеет доступ.

### Параметры

Параметры для метода указаны в таблице 48.

Таблица 48 — Перечень доступных параметров для чтения списка общих папок пользователя

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/users/<ID>/corporate" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификационный номер искомого пользователя.

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "corp_folders": [  
    {  
      "admin_email": "string",  
      "admin_first_name": "string",  
      "admin_id": "string",
```

```
"admin_last_name": "string",
"admin_username": "string",
"id": 0,
"name": "string"
"size": 0
},
]
```

- 404 — User not found — Пользователь не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.7.8 Чтение списка файлов пользователя

### Метод и конечная точка

GET/tenants/{tenant\_name}/users/{id}/files

### Описание

Позволяет получить количество файлов пользователя и их общий размер.

### Параметры

Параметры для метода указаны в таблице 49.

Таблица 49 — Перечень доступных параметров для чтения списка файлов пользователя

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

### Запрос

```
curl -X 'GET' \
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\
tenants/<TENANT_NAME>/users/<ID>/corporate" \
-H 'accept: application/json' \
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификационный номер искомого пользователя.

## Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{  
  "files_count": 0;  
  "success": true;  
  "used_bytes": 0;  
}
```

## 4.7.9 Чтение информации о файловой квоте пользователя

### Метод и конечная точка

GET/tenants/{tenant\_name}/users/{id}/quota

### Описание

Позволяет получить информацию о файловой квоте пользователя.

### Параметры

Параметры для метода указаны в таблице 50.

Таблица 50 — Перечень доступных параметров для чтения информации о файловой квоте пользователя

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

### Запрос

```
curl -X 'GET' \  
"https://admin[<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/users/<ID>/quota" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификационный номер искомого пользователя.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
"id": "string";  
"quota": 0;  
"success": true;  
"used_bytes": 0;  
"username": "string";  
}
```

## 4.7.10 Изменение файловой квоты пользователя

### Метод и конечная точка

PUT/tenants/{tenant\_name}/users/{id}/quota

### Описание

Позволяет изменить файловую квоту пользователя.

### Параметры

Параметры для метода указаны в таблице 51.

Таблица 51 — Перечень доступных параметров для изменения файловой квоты пользователя

Название	Обязательный	Тип данных	Описание
quota	да	string	Размер требуемой квоты в байтах
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

### Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/users/<ID>/quota?quota=<QUOTA>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификационный номер искомого пользователя;
- <QUOTA> — размер требуемой квоты в байтах.

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
"id": "string";  
"quota": 0;  
"success": true;
```

```
"used_bytes": 0;  
"username": "string";  
}
```

## 4.7.11 Изменение статуса двухфакторной авторизации пользователя

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/users/{id}/two\_fa\_settings

### Описание

Позволяет удалить (сделать reset) статус двухфакторной авторизации пользователя.

### Параметры

Параметры для метода указаны в таблице 52.

Таблица 52 — Перечень доступных параметров для изменения статуса двухфакторной авторизации пользователя

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
id	да	string	Идентификатор пользователя

### Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/users/<ID>/two_fa_settings" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <ID> — идентификационный номер искомого пользователя.

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
"files_count": 0;  
"success": true;  
}
```



```
"used_bytes": 0;  
}
```

## 4.7.12 Просмотр удаленных файлов пользователя

### Метод и конечная точка

GET/tenants/{tenant\_name}/recover/{user\_id}

### Описание

Позволяет вывести список удаленных за определенный временной промежуток файлов пользователя.

### Параметры

Параметры для метода указаны в таблице 53.

Таблица 53 — Перечень доступных параметров для просмотра списка удаленных файлов пользователя

Название	Обязательный	Тип данных	Описание
from	нет	string	Момент времени в формате unix timestamp, если параметр указан — показывать файлы, начиная с данного момента
to	нет	string	Момент времени в формате unix timestamp, если параметр указан — показывать файлы, заканчивая данным моментом
tenant_name	да	string	Имя тенанта
user_id	да	string	Идентификатор пользователя

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\
```

```
tenants/<TENANT_NAME>/recover/<USER_ID>&from=<TIME1>&to=<TIME2>" \
-H 'accept: application/json' \
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <USER\_ID> — идентификационный номер искомого пользователя;
- from=<TIME1> — временная точка в формате unix timestamp, обозначающая начало отбора файлов;
- to=<TIME2> — временная точка в формате unix timestamp, обозначающая конец отбора файлов.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{
  "files": [
    {
      "id": "string",
      "name": "string",
      "owner_id": "string",
      "mtime": 0
    }
  ]
}
```

- 400 — User does not belong to current tenant — Пользователь не найден в текущем тенанте. Пример вывода:

```
{
  "message": "string"
}
```

- 404 — Tenant does not exist — Тенант не существует. Пример вывода:

```
{
  "message": "string"
}
```

### 4.7.13 Восстановление удаленных файлов пользователя

#### Метод и конечная точка

POST/tenants/{tenant\_name}/recover/{user\_id}

#### Описание

Позволяет восстановить удаленные файлы пользователя.

## Параметры

Параметры для метода указаны в таблице 54.

Таблица 54 — Перечень доступных параметров для восстановления удаленных файлов пользователя

Название	Обязательный	Тип данных	Описание
file_ids	да	string	Идентификаторы файлов подлежащих восстановлению
new_owner_id	нет	string	Идентификатор нового владельца (в случае передачи прав)
tenant_name	да	string	Имя тенанта
user_id	да	string	Идентификатор пользователя

## Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/tenants/<TENANT_NAME>\   
/recover/<USER_ID>&file_ids=<ID1>,<ID2>,<ID3>&new_owner_id=<OWNER_ID>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <USER\_ID> — идентификационный номер искомого пользователя;
- file\_ids=<ID1>,<ID2>,<ID3> — идентификаторы файлов, подлежащих восстановлению (через запятую);
- new\_owner\_id=<OWNER\_ID> — идентификатор нового владельца (в случае передачи файлов)

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "ids": [  
    0  
  ]  
}
```

– 400 — File or user do not belong to current tenant — Файл или пользователь не найдены в текущем тенанте. Пример вывода:

```
{  
  "message": "string"  
}
```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.7.14 Окончательно удалить файлы пользователя

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/recover/{user\_id}

### Описание

Позволяет окончательно удалить файлы пользователя (из корзины).

### Параметры

Параметры для метода указаны в таблице 55.

Таблица 55 — Перечень доступных параметров для окончательного удаления файлов пользователя

Название	Обязательный	Тип данных	Описание
file_ids	да	string	Идентификаторы файлов подлежащих окончательному удалению
tenant_name	да	string	Имя тенанта
user_id	да	string	Идентификатор пользователя

### Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/recover/<USER_ID>&file_ids=<ID1>,<ID2>,<ID3>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <USER\_ID> — идентификационный номер искомого пользователя;
- file\_ids=<ID1>,<ID2>,<ID3> — идентификаторы файлов, подлежащих окончательному удалению (через запятую).

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- 400 — File or user do not belong to current tenant — Файл или пользователь не найдены в текущем тенанте. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.7.15 Получить аватар пользователя

### Метод и конечная точка

GET/tenants/{tenant\_name}/users/{id}/avatar

### Описание

Получить аватар пользователя.

### Параметры

Перечень параметров представлен в таблице 56.

Таблица 56 — Перечень параметров

Название	Обязательный	Тип данных	Описание
avatar	да	string	Аватар пользователя в формате base64 str
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/users/<ID>/avatar?avatar=<AVATAR>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{
  "avatar": "string",
  "id": "string",
  "success": true
}
```

## 4.7.16 Изменить аватар пользователя

### Метод и конечная точка

PUT/tenants/{tenant\_name}/users/{id}/avatar

### Описание

Изменить аватар пользователя.

### Параметры

Перечень параметров представлен в таблице 57.

Таблица 57 — Перечень параметров

Название	Обязательный	Тип данных	Описание
avatar	да	string	Аватар пользователя в формате base64 str
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

### Запрос

```
curl -X 'PUT' \
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\
tenants/<TENANT_NAME>/users/<ID>/avatar?avatar=<AVATAR>" \
-H 'accept: application/json' \
-H 'Authorization: <TOKEN>'
```

## Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{  
  "avatar": "string",  
  "id": "string",  
  "success": true  
}
```

## 4.7.17 Удалить аватар пользователя

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/users/{id}/avatar

### Описание

Удалить аватар пользователя.

### Параметры

Перечень параметров представлен в таблице 58.

Таблица 58 — Перечень параметров

Название	Обязательный	Тип данных	Описание
avatar	да	string	Аватар пользователя в формате base64 str
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

### Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/users/<ID>/avatar?avatar=<AVATAR>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{  
  "avatar": "string",  
  "id": "string",  
}
```



```
"success": true  
}
```

## 4.7.18 Изменить персональные данные пользователя

### Метод и конечная точка

PUT/tenants/{tenant\_name}/users/{id}/personal

### Описание

Изменить персональные данные пользователя.

### Параметры

Перечень параметров представлен в таблице 59.

Таблица 59 — Перечень параметров изменения персональных данных пользователя

Название	Обязательный	Тип данных	Описание
quota	да	string	Размер требуемой квоты в байтах
tenant_name	да	string	Имя тенанта
id	да	string	Идентификационный номер пользователя

### Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/users/<ID>/personal?quota=<QUOTA>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

### Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{  
  "ctime": "string",  
  "domain": "string",  
  "enabled": true,  
  "except_ctype_prohibition": true,  
  "first_name": "string",  
  "id": "string",  
  "is_deleted": true,
```

```
"lang": "string",
"last_name": "string",
"middle_name": "string",
"password": "string",
"position": "string",
"quota": 0,
"recovery_email": "string",
"roles": [
  {
    "clientRole": true,
    "composite": true,
    "containerId": "string",
    "description": "string",
    "id": "string",
    "name": "string"
  }
],
"username": "string"
}
```

## 4.8 Работа с группами пользователей

### 4.8.1 Перечень групп пользователей

#### Метод и конечная точка

GET/tenants/{tenant\_name}/groups

#### Описание

Запрос перечня существующих групп.

#### Параметры

Параметры для метода указаны в таблице 60.

Таблица 60 — Перечень параметров для вывода списка групп пользователей

Название	Обязательный	Тип данных	Описание
query	нет	string	Поисковый запрос. Если этот параметр установлен, то поиск осуществляется в соответствии со значением. Если параметр не установлен, выводится полный список всех групп
limit	нет	string	Ограничения поиска
offset	нет	string	Параметр настройки выдачи результатов поиска
with_ctypes	нет	string	Перечень запретов на загрузку файлов определенных типов для группы
exact	да	string	Точный поисковый запрос
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/groups?query=<QUERY>&limit=<LIMIT>&offset=<OFFSET>&\  
with_ctypes=<WITH_CTYPES>&exact=<EXACT>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <QUERY> — поисковый запрос;
- <LIMIT> — ограничения поиска;
- <OFFSET> — параметр настройки выдачи результатов поиска;
- <WITH\_CTYPES> — возможность работы с файлами в соответствии с правилами для определенных медиатипов;
- <EXACT> — точный поисковый запрос.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "groups": [  
    {  
      "ctypes": [  
        {  
          "grouped_ctype": "string",  
          "prohibited": true  
        }  
      ],  
      "id": 0,  
      "name": "string",  
      "subgroups": [  
        {  
          "id": 0,  
          "name": "string"  
        }  
      ],  
      "users": [  
        {  
          "first_name": "string",  
          "id": "string",  
          "last_name": "string",  
          "middle_name": "string",  
          "username": "string"  
        }  
      ]  
    }  
  ]  
}
```

```
]
}
```

- **400** — File or user do not belong to current tenant — Файл или пользователь не найдены в текущем тенанте. Пример вывода:

```
{
  "message": "string"
}
```

- **404** — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.8.2 Создание новой группы пользователей

### Метод и конечная точка

POST/tenants/{tenant\_name}/groups

Описание

Создать новую группу пользователей.

### Параметры

Параметры для метода указаны в Таблице 51.

Таблица 61 — Перечень доступных параметров для создания групп

Название	Обязательный	Тип данных	Описание
name	да	string	Имя новой группы
users	нет	string	Перечень идентификационных номеров пользователей добавляемых в группу
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'POST' \
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\
tenants/<TENANT_NAME>/groups?name=<NEW_GROUP_NAME>&users=<USER_ID>" \
-H 'accept: application/json' \
-H 'Authorization: <TOKEN>
```

где:

- <TENANT\_NAME> — имя тенанта;
- <NEW\_GROUP\_NAME> — имя новой группы;
- <USER\_ID> — идентификационные номера пользователей добавляемых в группу.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{
"groups": [
  {
    "ctypes": [
      {
        "grouped_ctype": "string",
        "prohibited": true
      }
    ],
    "id": 0,
    "name": "string",
    "subgroups": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "users": [
      {
        "first_name": "string",
        "id": "string",
        "last_name": "string",
        "middle_name": "string",
        "username": "string"
      }
    ]
  }
]
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

- 409 — Group with such name already exists — Группа с таким именем уже существует.  
Пример вывода:

```
{
  "message": "string"
}
```

## 4.8.3 Чтение информации о группе

### Метод и конечная точка

GET/tenants/{tenant\_name}/groups/{group\_id}

### Описание

Запрос установленных значений параметров групп.

### Параметры

Параметры для метода указаны в таблице 62.

Таблица 62 — Перечень доступных параметров для чтения информации о группе

Название	Обязательный	Тип данных	Описание
with_ctypes	нет	string	Перечень запретов на загрузку файлов определенных типов для группы
tenant_name	да	string	Имя тенанта
group_id	да	string	Идентификационный номер группы

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/groups/<GROUP_ID>?with_ctypes=<WITH_CTYPES>' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — имя новой группы;
- <WITH\_CTYPES> — возможность работы с файлами в соответствии с правилами для определенных медиатипов.

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{
  "groups": [
    {
      "ctypes": [
        {
          "grouped_ctype": "string",
          "prohibited": true
        }
      ],
      "id": 0,
      "name": "string",
      "subgroups": [
        {
          "id": 0,
          "name": "string"
        }
      ],
      "users": [
        {
          "first_name": "string",
          "id": "string",
          "last_name": "string",
          "middle_name": "string",
          "username": "string"
        }
      ]
    }
  ]
}
```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.8.4 Изменение имени группы

### Метод и конечная точка

PUT/tenants/{tenant\_name}/groups/{group\_id}

### Описание

Отправляет обновление значений параметров группы.

### Параметры

Параметры для метода указаны в таблице 63.



Таблица 63 — Перечень доступных параметров для переименования группы

Название	Обязательный	Тип данных	Описание
new_name	нет	string	Новое имя группы
tenant_name	да	string	Имя тенанта
group_id	да	string	Идентификационный номер группы

## Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/groups/<GROUP_ID>?new_name=<NEW_NAME>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — имя новой группы;
- <WITH\_STYPES> — возможность работы с файлами в соответствии с правилами для определенных медиатипов.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
"groups": [  
  {  
    "ctypes": [  
      {  
        "grouped_ctype": "string",  
        "prohibited": true  
      }  
    ],  
    "id": 0,  
    "name": "string",  
    "subgroups": [  
      {  
        "id": 0,  
        "name": "string"  
      }  
    ],  
    "users": [  
      {  
        "first_name": "string",  
        "id": "string",
```

```
    "last_name": "string",  
    "middle_name": "string",  
    "username": "string"  
  }  
]  
}  
]
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

- 409 — Group with such name already exists — Группа с таким именем уже существует. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.8.5 Удаление группы

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/groups/{group\_id}

### Описание

Удаляет заданную группу.

### Параметры

Параметры для метода указаны в таблице 64.

Таблица 64 — Перечень доступных параметров для удаления группы

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
group_id	да	string	Идентификационный номер группы

### Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/groups/<GROUP_ID>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — имя удаляемой группы.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.8.6 Добавление пользователя в группу

### Метод и конечная точка

POST/tenants/{tenant\_name}/groups/{group\_id}/users

### Описание

Добавляет пользователя в заданную группу.

### Параметры

Параметры для метода указаны в таблице 65.

Таблица 65 — Перечень доступных параметров для добавления пользователя в группу

Название	Обязательный	Тип данных	Описание
users	нет	string	Перечень добавляемых пользователей
tenant_name	да	string	Имя тенанта
group_id	да	string	Идентификационный номер группы

### Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\
```

```
tenants/<TENANT_NAME>/groups/<GROUP_ID>/users?users=<USER>" \
-H 'accept: application/json' \
-H 'Authorization: <TOKEN>
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — имя группы;
- <USER> — добавляемый пользователь.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{
  "message": "string"
}
```

- 400 — One of the following parameters is required: username, id. Or user with such id/username does not exist — Требуется один из следующих параметров: имя пользователя, id. Или пользователь с таким идентификатором/именем пользователя не существует. Пример вывода:

```
{
  "message": "string"
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.8.7 Удаление пользователя из группы

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/groups/{group\_id}/users

### Описание

Удаляет пользователя из группы.

### Параметры

Параметры для метода указаны в таблице 66.

Таблица 66 — Перечень доступных параметров для удаления пользователя из группы

Название	Обязательный	Тип данных	Описание
users	нет	string	Перечень добавляемых пользователей
tenant_name	да	string	Имя тенанта
group_id	да	string	Идентификационный номер группы

## Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/groups/<GROUP_ID>/users?users=<USER>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — имя группы;
- <USER> — удаляемый пользователь.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- 400 — User is not a group member — Пользователь не входит в группу. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant, user or group not found — Тенант, пользователь или группа не найден.

Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.8.8 Получение списка подгрупп

### Метод и конечная точка

GET/tenants/{tenant\_name}/groups/{group\_id}/groups

### Описание

Выводит список подгрупп в искомой группе.

### Параметры

Параметры для метода указаны в таблице 67.

Таблица 67— Перечень доступных параметров для получения списка подгрупп

Название	Обязательный	Тип данных	Описание
with_ctypes	нет	string	Вывод подгрупп с правилами для медиатипов
tenant_name	да	string	Имя тенанта
group_id	да	string	Идентификационный номер группы

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/groups/<GROUP_ID>/groups?\  
with_ctypes=<WITH_CTYPES>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — имя группы;
- <WITH\_CTYPES> — правила для медиатипов.

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "ctypes": [  

```

```
{
  "grouped_ctype": "string",
  "prohibited": true
},
],
"id": 0,
"name": "string",
"subgroups": [
  {
    "id": 0,
    "name": "string"
  }
],
"users": [
  {
    "first_name": "string",
    "id": "string",
    "last_name": "string",
    "middle_name": "string",
    "username": "string"
  }
] }
```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```

## 4.8.9 Добавление подгруппы

### Метод и конечная точка

POST/tenants/{tenant\_name}/groups/{group\_id}/groups

Описание

Добавление подгруппы в группу.

### Параметры

Параметры для метода указаны в таблице 68.

Таблица 68 — Перечень доступных параметров для добавления подгруппы

Название	Обязательный	Тип данных	Описание
subgroup_name	да	string	Имя создаваемой подгруппы
users	нет	string	Имена пользователей, которых требуется добавить в подгруппу

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта
group_id	да	string	Идентификационный номер группы, в которой создается подгруппа

## Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/groups/<GROUP_ID>/groups?  
subgroup_name=<SUBGROUP_NAME>&users=<USER1>,<USER2>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — идентификатор группы;
- <SUBGROUP\_NAME> — правила для медиатипов;
- users=<USER1>,<USER2> — имена добавляемых в подгруппу пользователей.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "ctype": [  
    {  
      "grouped_ctype": "string",  
      "prohibited": true  
    }  
  ],  
  "id": 0,  
  "name": "string",  
  "subgroups": [  
    {  
      "id": 0,  
      "name": "string"  
    }  
  ],  
  "users": [  
    {  
      "first_name": "string",  
      "id": "string",  
      "last_name": "string",  
      "middle_name": "string",
```



```
"username": "string"  
}  
]  
}
```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

– 409 — Failed to create subgroup — Невозможно создать подгруппу, скорее всего, имя уже занято. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.9 Работа с публичными ссылками

### 4.9.1 Получение списка публичных ссылок

#### Метод и конечная точка

GET/tenants/{tenant\_name}/public\_links

#### Описание

Возвращает список публичных ссылок в тенанте.

#### Параметры

Параметры для метода указаны в таблице 69.

Таблица 69 — Перечень доступных параметров для получения списка публичных ссылок тенанта

Название	Обязательный	Тип данных	Описание
offset	нет	string	Параметр настройки выдачи результатов поиска
limit	нет	string	Ограничения поиска
sort_by	нет	string	Сортировать по параметрам
asc	нет	string	Порядок сортировки результатов выдачи
tenant_name	да	string	Имя тенанта

#### Запрос

```
curl -X 'GET' \  
"https://admin[<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/public_links?offset=<OFFSET>&  
limit=<LIMIT>&sort_by=<SORT_BY>&asc=<ASC>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <OFFSET> — параметр настройки выдачи результатов поиска;
- <LIMIT> — ограничения поиска;
- <SORT\_BY> — сортировка по параметрам;

– `<ASC>` — сортировка результатов поиска.

## Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{
  "count": 0, {
  "public_links": [
    "end_time": 0,
    "first_name": "string",
    "is_password": true,
    "key": 0,
    "last_name": "string",
    "object_name": "string",
    "object_type": "string",
    "owner_id": "string",
    "start_time": "string",
    "status": "string",
    "type": "string",
    }
  ]
}
```

## 4.9.2 Блокировка и разблокировка публичной ссылки

### Метод и конечная точка

PUT/tenants/{tenant\_name}/public\_links/{link\_key}

### Описание

Позволяет заблокировать и разблокировать указанную публичную ссылку.

### Параметры

Параметры для метода указаны в таблице 70.

Таблица 70 — Перечень доступных параметров для блокировки и разблокировки публичной ссылки

Название	Обязательный	Тип данных	Описание
enabled	нет	string	Заблокировать / разблокировать публичную ссылку
tenant_name	да	string	Имя тенанта
link_key	да	string	Идентификатор (ключ) ссылки, может быть получен методом GET из предыдущего раздела

### Запрос

```
curl -X 'PUT' \  
"https://admin[<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/public_links/<LINK_KEY>?enabled=<ENABLED>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <LINK\_KEY> — значение идентификатора (ключа) ссылки, параметра key, полученного методом GET из предыдущего раздела;
- enabled=<ENABLED> — если параметр указан, то ссылка будет разблокирована (true) или заблокирована (false).

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.9.3 Удаление публичной ссылки

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/public\_links

### Описание

Позволяет удалить указанную публичную ссылку или все публичные ссылки указанного пользователя.

### Параметры

Параметры для метода указаны в таблице 71.

Таблица 71 — Перечень доступных параметров для удаления публичной ссылки

Название	Обязательный	Тип данных	Описание
query	нет	string	В зависимости от значения — ключ ссылки или идентификатор пользователя — удаляет указанную ссылку или все ссылки пользователя соответственно
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/public_links?query=<QUERY>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- `<TENANT_NAME>` — имя тенанта;
- `query=<QUERY>` — если параметр указан как идентификатор ссылки (`key`), то публичная ссылка будет удалена, если параметр указан как идентификатор пользователя (`owner_id`), то все публичные ссылки указанного пользователя будут удалены.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "message": "string"  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

### 4.9.4 Изменение настроек публичных ссылок в тенанте

#### Метод и конечная точка

PUT/tenants/{tenant\_name}/public\_link\_settings

#### Описание

Позволяет изменить настройки публичных ссылок в тенанте.

#### Параметры

Параметры для метода указаны в таблице 72.

Таблица 72 — Перечень доступных параметров для изменения настроек публичных ссылок  
тенанта

Название	Обязательный	Тип данных	Описание
public_link_settings	нет	string	Параметр для настройки публичных ссылок
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'PUT' \  
"https://admin[<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>" \  
-d public_link_settings=  
"{\"public_link_access_time_without_password\": 0,  
\"public_link_corp_folders_enabled\": true,  
\"public_link_enabled\": true,  
\"public_link_password_required\": true,  
\"public_link_threshold_attempts_number\": 0}" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- `<TENANT_NAME>` — имя тенанта;
- `public_link_settings` — настройки для работы с публичными ссылками в тенанте:
  - `"public_link_enabled": true` — разрешить создание публичных ссылок; возможные значения true или false, по умолчанию true (включено);
  - `"public_link_corp_folders_enabled": true` — разрешить публичные ссылки для общих папок и их содержимого; возможные значения true или false, по умолчанию true (включено);
  - `"public_link_password_required": true` — включить защиту паролем для всех ссылок;
  - `"public_link_access_time_without_password": 0` — продолжительность доступа после ввода пароля;
  - `"public_link_threshold_attempts_number": 0` — количество допустимых неудачных попыток ввода пароля.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{
  "id": "string",
  "name": "string",
  "public_link_settings": {
    "public_link_access_time_without_password": 0,
    "public_link_corp_folders_enabled": true,
    "public_link_enabled": true,
    "public_link_password_required": true,
    "public_link_threshold_attempts_number": 0
  }
}
```

- 401 — Auth header is not provided or token is invalid — Не указан параметр авторизации или некорректный токен. Пример вывода:

```
{
  "message": "string"
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{
  "message": "string"
}
```



## 4.10 Работа с ключами доступа

### 4.10.1 Получение списка ключей доступа в тенанте

и конечная точка

GET/tenants/{tenant\_name}/domains/{domain}/promo\_codes

#### Описание

Позволяет получить список ключей доступа, сгенерированных в указанном домене тенанта.

#### Параметры

Параметры для метода указаны в таблице 73.

Таблица 73 — Перечень доступных параметров для получения списка ключей доступа

Название	Обязательный	Тип данных	Описание
creator	нет	string	Админ тенанта, при указании будут возвращены только созданные им ключи доступа
status	нет	string	Показать только ключи доступа с указанным статусом
tenant_name	да	string	Имя тенанта
domain	да	string	Домен, в котором сгенерированы ключи доступа

#### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/domains/<DOMAIN>/promo_codes?\  
creator=<CREATOR_NAME>&status=<STATUS>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <DOMAIN> — домен, в котором выданы ключи доступа;
- <CREATOR\_NAME> — ограничение вывода по указанному создателю ключей доступа;

– `<STATUS>` — ограничение вывода по статусу ключей доступа.

## Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{
  "promo_codes": [
    "string",
  ]
}
```

## 4.10.2 Создание ключа доступа в тенанте

### Метод и конечная точка

POST/tenants/{tenant\_name}/domains/{domain}/promo\_codes

### Описание

Позволяет создать ключ доступа в указанном домене тенанта.

### Параметры

Параметры для метода указаны в таблице 74.

Таблица 74 — Перечень доступных параметров для создания ключа доступа

Название	Обязательный	Тип данных	Описание
count	да	string	Количество создаваемых ключей доступа
lifespan	нет	string	Длительность действия создаваемых ключей доступа
output	нет	string	Формат вывода сгенерированных ключей доступа: json или file
tenant_name	да	string	Имя тенанта
domain	да	string	Домен, в котором сгенерированы ключи доступа

## Запрос

```
curl -X 'POST' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/domains/<DOMAIN>/promo_codes' \  
-d promo_codes='{ "count": <COUNT>, "lifespan": <LIFESPAN>, "output": <OUTPUT>}' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <DOMAIN> — домен, в котором выданы ключи доступа;
- <COUNT> — количество выдаваемых ключей доступа;
- <LIFESPAN> — длительность действия выдаваемых ключей доступа;
- <OUTPUT> — формат вывода сгенерированных ключей доступа: json или file, по умолчанию json.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "promo_codes": [  
    "string",  
  ]  
}
```

### 4.10.3 Удаление ключа доступа в тенанте

#### Метод и конечная точка

DELETE/tenants/{tenant\_name}/domains/{domain}/promo\_codes/{promo\_code}

#### Описание

Позволяет удалить указанный ключ доступа.

#### Параметры

Параметры для метода указаны в таблице 75.

Таблица 75 — Перечень доступных параметров для удаления ключа доступа

Название	Обязательный	Тип данных	Описание
promo_code	да	string	Ключ доступа, подлежащий удалению
tenant_name	да	string	Имя тенанта
domain	да	string	Домен, в котором сгенерированы ключи доступа

## Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/domains/<DOMAIN>/promo_codes" \  
-d promo_codes='{ "count": <COUNT>, "lifespan": <LIFESPAN>, "output": <OUTPUT>}' \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <DOMAIN> — домен, в котором выданы ключи доступа;
- <PROMO\_CODE> — удаляемый ключ доступа.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "promo_codes": [  
    "string",  
  ]  
}
```

- 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.11 Работа с медиатипами

### 4.11.1 Получение списка групп с запретами на загрузку медиатипов

#### Метод и конечная точка

GET/tenants/{tenant\_name}/ctypes\_prohibition/groups

#### Описание

Позволяет получить список групп, для которых установлен запрет на загрузку определенных медиатипов.

#### Параметры

Параметры для метода указаны в таблице 76.

Таблица 76 — Перечень доступных параметров для просмотра списка групп с запретами на медиатипы

Название	Обязательный	Тип данных	Описание
sort_order	нет	string	Порядок сортировки результатов выдачи, может принимать значения asc или desc
tenant_name	да	string	Имя тенанта

#### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/ctypes_prohibition/groups?\  
sort_order=<SORT_ORDER>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <SORT\_ORDER> — порядок сортировки результатов выдачи, может принимать значения asc или desc.

#### Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{
  "count": 0,
  "ctypes": [
    {
      "ctime": 0,
      "ctypes": [
        {
          "grouped_ctype": "string",
          "prohibited": true
        }
      ],
      "etime": 0,
      "id": 0,
      "name": "string",
    }
  ]
}
```

– 404 — Tenant or group not found — Тенант или группа не найдены. Пример вывода:

```
{
  "message": "string"
}
```

## 4.11.2 Изменение списка групп с запретами за загрузку медиатипов

### Метод и конечная точка

PUT/tenants/{tenant\_name}/ctypes\_prohibition/groups

### Описание

Позволяет изменить список групп, для которых установлен запрет на загрузку определенных медиатипов.

### Параметры

Параметры для метода указаны в таблице 77.

Таблица 77 — Перечень доступных параметров для изменения списка групп с запретами для медиатипов

Название	Обязательный	Тип данных	Описание
groups	да	string	Группы, для которых будут изменены запреты на загрузку медиатипов

Название	Обязательный	Тип данных	Описание
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/ctypes_prohibition/groups?groups=<GROUPS>"  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUPS> — идентификаторы групп пользователей, для которых будут изменены запреты на загрузку медиатипов.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  {  
    "ctime": 0,  
    "ctypes": [  
      {  
        "grouped_ctype": "string",  
        "prohibited": true  
      }  
    ],  
    "etime": 0,  
    "id": 0,  
    "name": "string",  
  }  
}
```

- 404 — Tenant or group not found — Тенант или группа не найдены. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.11.3 Чтение запретов на загрузку медиатипов для группы

### Метод и конечная точка

GET/tenants/{tenant\_name}/ctypes\_prohibition/groups/{group\_id}

### Описание

Позволяет получить список запретов для медиатипов определенной группы пользователей.

### Параметры

Параметры для метода указаны в таблице 78.

Таблица 78 — Перечень доступных параметров для просмотра запретов медиатипов у группы

Название	Обязательный	Тип данных	Описание
group_id	да	string	Группа, для которой необходимо найти запреты на загрузку медиатипов
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/ctypes_prohibition/groups/<GROUP_ID>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — идентификатор искомой группы.

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  {  
    "ctime": 0,  
    "ctypes": [  
      {
```



```
    "grouped_ctype": "string",  
    "prohibited": true  
  },  
  ],  
  "etime": 0,  
  "id": 0,  
  "name": "string",  
  }  
}
```

– 404 — Tenant or group not found — Тенант или группа не найдены. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.11.4 Изменение запретов на загрузку медиатипов для группы

### Метод и конечная точка

PUT/tenants/{tenant\_name}/ctypes\_prohibition/groups/{group\_id}

### Описание

Позволяет изменить список запретов для медиатипов определенной группы пользователей.

### Параметры

Параметры для метода указаны в таблице 79.

Таблица 79 — Перечень доступных параметров для изменения запретов медиатипов у группы

Название	Обязательный	Тип данных	Описание
prohibited_ctype	да	string	Изменяемые запреты для медиатипов
group_id	да	string	Группа, для которой необходимо изменить запреты на загрузку медиатипов
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'PUT' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/ctypes_prohibition/groups/<GROUP_ID>?"
```

```
prohibited_ctypes=<CTYPES>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — идентификатор искомой группы;
- <CTYPES> — изменяемые запреты для медиатипов.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  {  
    "ctime": 0,  
    "ctypes": [  
      {  
        "grouped_ctype": "string",  
        "prohibited": true  
      }  
    ],  
    "etime": 0,  
    "id": 0,  
    "name": "string",  
  }  
}
```

- 404 — Tenant or group not found — Тенант или группа не найдены. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.11.5 Удаление запретов на загрузку медиатипов для группы

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/ctypes\_prohibition/groups/{group\_id}

### Описание

Позволяет удалить список запретов для медиатипов определенной группы пользователей.

### Параметры

Параметры для метода указаны в таблице 80.

Таблица 80 — Перечень доступных параметров для удаления запретов медиатипов у группы

Название	Обязательный	Тип данных	Описание
group_id	да	string	Группа, для которой необходимо удалить запреты на загрузку медиатипов
tenant_name	да	string	имя тенанта

## Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
tenants/<TENANT_NAME>/ctypes_prohibition/groups/<GROUP_ID>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <GROUP\_ID> — идентификатор искомой группы.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  {  
    "ctime": 0,  
    "ctypes": [  
      {  
        "grouped_ctype": "string",  
        "prohibited": true  
      }  
    ],  
    "etime": 0,  
    "id": 0,  
    "name": "string",  
  }  
}
```

- 404 — Tenant or group not found — Тенант или группа не найдены. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.11.6 Получение списка пользователей с запретами на загрузку медиатипов

### Метод и конечная точка

GET/tenants/{tenant\_name}/ctypes\_prohibition/users

### Описание

Позволяет получить список пользователей, для которых установлен запрет на загрузку определенных медиатипов.

### Параметры

Параметры для метода указаны в таблице 81.

Таблица 81 — Перечень доступных параметров для просмотра списка пользователей с запретами на медиатипы

Название	Обязательный	Тип данных	Описание
sort_order	нет	string	Порядок сортировки результатов выдачи, может принимать значения asc или desc
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/ctypes_prohibition/users?sort_order=<SORT_ORDER>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <SORT\_ORDER> — порядок сортировки результатов выдачи, может принимать значения asc или desc.

### Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "count": 0,
```

```
"users": [  
  {  
    "ctime": "string",  
    "domain": "string",  
    "enabled": true,  
    "except_ctype_prohibition": true,  
    "first_name": "string",  
    "id": "string",  
    "is_deleted": true,  
    "lang": "string",  
    "last_name": "string",  
    "middle_name": "string",  
    "password": "string",  
    "position": "string",  
    "quota": 0,  
    "recovery_email": "string",  
    "roles": [  
      {  
        "clientRole": true,  
        "composite": true,  
        "containerId": "string",  
        "description": "string",  
        "id": "string",  
        "name": "string"  
      }  
    ],  
    "username": "string"  
  }  
],  
"users_count": 0  
}
```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```
{  
  "message": "string"  
}
```

## 4.11.7 Чтение запретов на загрузку медиатипов для пользователя

### Метод и конечная точка

GET/tenants/{tenant\_name}/ctypes\_prohibition/users/{user\_id}

### Описание

Позволяет получить список запретов медиатипов для определенного пользователя.

### Параметры

Параметры для метода указаны в таблице 82.

Таблица 82 — Перечень доступных параметров для просмотра запретов медиатипов определенного пользователя

Название	Обязательный	Тип данных	Описание
user_id	да	string	Пользователь, для которого необходимо найти запреты на загрузку медиатипов
tenant_name	да	string	Имя тенанта

## Запрос

```
curl -X 'GET' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/  
\tenants/<TENANT_NAME>/ctypes_prohibition/users/<USER_ID>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <USER\_ID> — идентификатор искомой группы.

## Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{  
  "count": 0,  
  "users": [  
    {  
      "ctime": "string",  
      "domain": "string",  
      "enabled": true,  
      "except_ctype_prohibition": true,  
      "first_name": "string",  
      "id": "string",  
      "is_deleted": true,  
      "lang": "string",  
      "last_name": "string",  
      "middle_name": "string",  
      "password": "string",  
      "position": "string",  
      "quota": 0,  
      "recovery_email": "string",  
      "roles": [  
        {  
          "clientRole": true,  
          "composite": true,  
          "containerId": "string",  
          "description": "string",
```

```

        "id": "string",
        "name": "string"
    }
],
"username": "string"
}
],
"users_count": 0
}

```

– 404 — Tenant not found — Тенант не найден. Пример вывода:

```

{
    "message": "string"
}

```

## 4.11.8 Изменение запретов на загрузку медиатипов для пользователя

### Метод и конечная точка

PUT/tenants/{tenant\_name}/ctypes\_prohibition/users/{user\_id}

### Описание

Позволяет изменить список запретов медиатипов для определенного пользователя.

### Параметры

Параметры для метода указаны в таблице 83.

*Таблица 83 — Перечень доступных параметров для изменения запретов медиатипов определенного пользователя*

Название	Обязательный	Тип данных	Описание
except_ctype_prohibition	да	string	Изменяемые запреты для медиатипов
user_id	да	string	Группа, для которой необходимо изменить запреты на загрузку медиатипов
tenant_name	да	string	Имя тенанта

### Запрос

```

curl -X 'PUT' \
"https://admin[<ENV>].<DEFAULT_DOMAIN>/adminapi/\
tenants/<TENANT_NAME>/ctypes_prohibition/users/<USER_ID>?\

```

```
except_ctype_prohibition=<CTYPES>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <USER\_ID> — идентификатор искомого пользователя;
- <CTYPES> — изменяемые запреты для медиатипов.

## Ответ

Возможные варианты ответов:

- 200 — Success — Успешно. Пример вывода:

```
{  
  "count": 0,  
  "users": [  
    {  
      "ctime": "string",  
      "domain": "string",  
      "enabled": true,  
      "except_ctype_prohibition": true,  
      "first_name": "string",  
      "id": "string",  
      "is_deleted": true,  
      "lang": "string",  
      "last_name": "string",  
      "middle_name": "string",  
      "password": "string",  
      "position": "string",  
      "quota": 0,  
      "recovery_email": "string",  
      "roles": [  
        {  
          "clientRole": true,  
          "composite": true,  
          "containerId": "string",  
          "description": "string",  
          "id": "string",  
          "name": "string"  
        }  
      ],  
      "username": "string"  
    }  
  ],  
  "users_count": 0  
}
```

- 404 — Tenant or group not found — Тенант или группа не найдены. Пример вывода:

```
{  
  "message": "string"  
}
```



## 4.11.9 Удаление запретов на загрузку медиатипов для пользователя

### Метод и конечная точка

DELETE/tenants/{tenant\_name}/ctypes\_prohibition/users/{user\_id}

### Описание

Позволяет удалить список запретов медиатипов для определенного пользователя.

### Параметры

Параметры для метода указаны в таблице 84.

Таблица 84 — Перечень доступных параметров для удаления запретов медиатипов определенного пользователя

Название	Обязательный	Тип данных	Описание
user_id	да	string	Пользователь, для которого необходимо удалить запреты на загрузку медиатипов
tenant_name	да	string	Имя тенанта

### Запрос

```
curl -X 'DELETE' \  
"https://admin[-<ENV>].<DEFAULT_DOMAIN>/adminapi/\  
tenants/<TENANT_NAME>/ctypes_prohibition/users/<USER_ID>" \  
-H 'accept: application/json' \  
-H 'Authorization: <TOKEN>'
```

где:

- <TENANT\_NAME> — имя тенанта;
- <USER\_ID> — идентификатор искомого пользователя.

### Ответ

Возможные варианты ответов:

– 200 — Success — Успешно. Пример вывода:

```
{  
  "count": 0,  
  "users": [  
    {  
      "ctime": "string",  
      "domain": "string",
```

```
"enabled": true,
"except_ctype_prohibition": true,
"first_name": "string",
"id": "string",
"is_deleted": true,
"lang": "string",
"last_name": "string",
"middle_name": "string",
"password": "string",
"position": "string",
"quota": 0,
"recovery_email": "string",
"roles": [
  {
    "clientRole": true,
    "composite": true,
    "containerId": "string",
    "description": "string",
    "id": "string",
    "name": "string"
  }
],
"username": "string"
}
],
"users_count": 0
}
```

– 404 — Tenant or group not found — Тенант или группа не найдены. Пример вывода:

```
{
  "message": "string"
}
```