

**Руководство пользователя ПО  
«Объектное хранилище»**

# 1. Введение

## 1.1. Область применения

Настоящий документ предназначен для сотрудников эксплуатирующей организации и отражает основные функциональные возможности и порядок взаимодействия с программным и интерфейсом программного обеспечения «Объектное хранилище» (далее - «Система»)

## 1.3. Уровень подготовки оператора

Оператор Системы должен понимать общие принципы работы HTTP-протокола, уметь делать REST HTTP-вызовы с помощью системных инструментов или любого языка программирования.

Рекомендуемая численность персонала для эксплуатации Системы — 1 штатная единица.

Операторы Системы должны пройти обязательную общую подготовку для работы с Системой.

Общая подготовка должна включать в себя получение знаний и навыков работы с Системой в качестве пользователя.

# 2. Описание API

## 2.1. Общие сведения

Система предоставляет возможность взаимодействия с собой посредством REST API.

Часть методов API требует авторизации посредством Bearer Token. Для начала взаимодействия, требуется получить у Администратора Системы текущее значение токена, установленное им при установке Системы.

Bearer-токен — это тип авторизационного токена, который предоставляется пользователю для доступа к защищенным ресурсам. Этот токен обычно представляет собой строку, которую пользователь должен передавать в заголовке HTTP-запроса для каждого запроса к защищенному сервису или приложению.

Bearer-токен используется для предоставления доступа к защищенным ресурсам. Он должен храниться в безопасном месте и не передаваться третьим лицам.

Bearer-токен обеспечивает простой механизм аутентификации без необходимости использования cookies или сессий. Он является частью стандарта аутентификации OAuth 2.0 и широко используется во многих современных веб-приложениях и API.

Объекты в хранилище представляют собой файлы и взаимодействие с ними организовано как взаимодействие с файлами.

## 2.2. Загрузка файла

Для загрузки файла в Систему должен использоваться endpoint `/files`

Для доступа к endpointу требуется аутентификация посредством Bearer token.

Обращение производится посредством POST-запроса. Тело запроса представляет собой стандартную http-форму в формате form-data. Параметры формы:

- **file** (file): Файл, который нужно загрузить.
- **uncompressed** (text): Опциональный параметр. true или 1, если требуется не сжимать файл после загрузки

При успешном выполнении, Система вернет статус: 200. Content-Type: application/json.

В теле ответа будет json-объект с параметрами:

- **id** (string): Идентификатор загруженного файла. Строка, содержащая UID
- **message** (string): Текстовое сообщение, указывающее статус загрузки файла.

## 2.3. Загрузка файла с указанием id

Для загрузки файла в Систему должен использоваться endpoint `/files/<идентификатор_файла>`

В качестве идентификатора файла должен быть UID

Для доступа к endpointу требуется аутентификация посредством Bearer token.

Обращение производится посредством PUT-запроса. Тело запроса представляет собой стандартную http-форму в формате form-data. Параметры формы:

- **file** (file): Файл, который нужно загрузить.
- **uncompressed** (text): Опциональный параметр. true или 1, если требуется не сжимать файл после загрузки

При успешном выполнении, Система вернет статус: 200. Content-Type: application/json.

В теле ответа будет json-объект с параметрами:

- **id** (string): Идентификатор загруженного файла.
- **message** (string): Текстовое сообщение, указывающее статус загрузки файла.

## 2.4. Получение файла

Для загрузки файла из Системы должен использоваться endpoint `/files/<идентификатор_файла>`

В качестве идентификатора файла должен быть UID

Для доступа к endpointу требуется аутентификация посредством Bearer token.

Обращение производится посредством GET-запроса.

При успешном выполнении, Система вернет статус: 200. В теле запроса будет содержимое файла.

## 2.5. Удаление файла

Для удаления файла в Системе должен использоваться эндпоинт `/files/<идентификатор_файла>`

В качестве идентификатора файла должен быть UID

Для доступа к эндпоинту требуется аутентификация посредством Bearer token.

Обращение производится посредством DELETE-запроса.

При успешном выполнении, Система вернет статус: 200. Content-Type: application/json.

В теле ответа будет json-объект с параметрами:

- **id** (string): Идентификатор удаленного файла.
- **message** (string): Текстовое сообщение, указывающее статус удаления файла.

## 2.6. Получение списка идентификаторов файлов

Для загрузки файла из Системы должен использоваться эндпоинт `/files`

Для доступа к эндпоинту требуется аутентификация посредством Bearer token.

Обращение производится посредством GET-запроса.

При успешном выполнении, Система вернет статус: 200. Content-Type: application/json.

В теле ответа будет json-объект с параметрами:

- **count** (integer): Количество файлов в Системе.
- **keys** (array[string]): Список идентификаторов (ключей) файлов

## 2.7. Получение метаданных файла

Для загрузки файла из Системы должен использоваться эндпоинт `/files/stats/<идентификатор_файла>`

Для доступа к эндпоинту требуется аутентификация посредством Bearer token.

Обращение производится посредством GET-запроса.

При успешном выполнении, Система вернет статус: 200. Content-Type: application/json.

В теле ответа будет json-объект с параметрами:

- **file\_name** (string): Имя файла.
- **id** (string): Уникальный идентификатор файла.
- **mime\_type** (string): MIME-тип файла.

- **storage\_used** (number): Объем использованного хранилища для файла.

## 2.8. Создание открытой ссылки для файла

Для загрузки файла из Системы должен использоваться эндпоинт `/files/share/<идентификатор_файла>`

Для доступа к эндпоинту требуется аутентификация посредством Bearer token.

Обращение производится посредством POST-запроса.

При успешном выполнении, Система вернет статус: 200. Content-Type: application/json.

В теле ответа будет json-объект с параметрами:

- **id** (string): Уникальный идентификатор ссылки на файл.
- **message** (string): Текстовое сообщение, указывающее статус создания ссылки.

## 2.9. Удаление открытой ссылки для файла

Для загрузки файла из Системы должен использоваться эндпоинт `/files/share/<идентификатор_ссылки_на_файл>`

Для доступа к эндпоинту требуется аутентификация посредством Bearer token.

Обращение производится посредством DELETE-запроса.

При успешном выполнении, Система вернет статус: 200. Content-Type: application/json.

В теле ответа будет json-объект с параметрами:

- **id** (string): Уникальный идентификатор ссылки на файл.
- **message** (string): Текстовое сообщение, указывающее статус удаления ссылки.

## 2.10. Загрузка файла без авторизации

Для загрузки файла из Системы должен использоваться эндпоинт `/<идентификатор_ссылки_на_файл>`

Для доступа к эндпоинту не требуется аутентификация.

Обращение производится посредством GET-запроса.

При успешном выполнении, Система вернет статус: 200. В теле ответа будет содержимое файла.

## 2.11. Получение статистики о работе сервиса

Для загрузки файла из Системы должен использоваться эндпоинт `/stats`

Для доступа к эндпоинту не требуется аутентификация.

Обращение производится посредством GET-запроса.

При успешном выполнении, Система вернет статус: 200. Content-Type: application/json.

В теле ответа будет json-объект с параметрами:

- **files\_count** (number): Количество объектов, сохраненных в Системе
- **storage\_used** (number): Занимаемое объектами место в хранилище в байтах