

**FileStorage** - сервер для хранения произвольных файлов. Поддерживает функционал:

- Загрузка файла
- Получение метаданных о файлах (id, размер, тип)
- Скачивание файла
- Удаление файла

Для хранения метаданных и фильтрации файлов предлагается использовать SQLite

#### **API:**

##### 1. **POST** /api/upload

###### **Description:**

- Загружает файл на сервер.
- Требуется корректное указание заголовка Content-Length.
- Если имени файла нет в параметрах, имя файла будет таким же как id.
- Заголовок Content-Type сохраняется и выставляется обратно при попытке загрузки файла
- Если произойдет загрузка файла с уже существующим id - файл будет перезаписан

###### **Params:**

**id** - идентификатор файла (если не нужна автоматическая генерация)

**name** - имя файла

**tag** - тэг файла (можно оставить пустым)

**Payload:** содержимое файла

**Status code:** 201 Created

и json объект, содержащий все атрибуты загруженного файла

- **id** - идентификатор файла
- **name** - имя файла
- **tag** - тэг файла
- **size** - размер файла
- **mimeType** - заголовок Content-Type переданный при загрузке
- **modificationTime** - время последней загрузки файла

##### 2. **GET** /api/get

###### **Description:**

Возвращает метаданные файлов удовлетворяющих условию. Если условие пустое - возвращает метаданные всех файлов на сервере

###### **Params:**

- Все параметры указанные в этом запросе являются параметрами фильтрации. Параметры можно указывать по несколько штук за раз. Повторно указанные параметры будут использоваться в фильтрации через логическое ИЛИ, в то время как разные параметры - через И.
- Например `id=1&id=2&name=photo.jpg` значит (`id = 1` ИЛИ `id = 2`) И `name = photo.jpg`

**Status code:** 200 OK

json массив объектов. в каждом объекте есть те же поля, что можно использовать для фильтрации

- **id** - идентификатор файла
- **name** - имя файла
- **tag** - тэг файла
- **size** - размер файла
- **mimeType** - заголовок Content-Type переданный при загрузке
- **modificationTime** - время последней загрузки файла

### 3. DELETE /api/delete

**Description:**

- Удаляет файлы по условию
- Условия задаются аналогично `/api/get`
- Не допускается отсутствие условий, чтобы избежать удаления всех файлов

**Status code:**

- 200 X files deleted где X - количество удалённых файлов
- 400 - отсутствуют условия

### 4. GET /api/download

**Description:**

- Загружает указанный файл с сервера
- Оригинальное имя файла лежит в Content-Disposition как `filename="<имя файла>"`
- Content-Type выставляется таким же как был при загрузке файла на сервер

**Params:**

- **id** - идентификатор файла

**Status code:**

- 200 OK + тело файла
- 404 Not found - файл не существует

Дополнительный цели.

- 1) Limit / offset
- 2) Auto detect **mimeType**
- 3) Configurable meta attributes