

File storage

- [Описание](#)
- [REST API](#)
- [Host: fsms.iqoption.com](#)
 - [\[POST\] /api/v1/file](#)
 - [\[POST\] /api/v1/file/from-multipart](#)
 - [\[DELETE\] /api/v1/file/{fileUri}](#)
 - [\[PUT\] /api/v1/file/{fileUri}](#)
 - [\[PUT\] /api/v1/file/from-multipart/{fileUri}](#)
 - [\[POST\] /api/v1/image/from-multipart](#)
- [Использование имени в ссылках](#)

Описание

Микросервис хранения файлов предоставляет возможность клиенту средствами REST загружать файлы в хранилище, чтобы в дальнейшем предоставить пользователю доступ к ним. Этот доступ на усмотрение клиента может быть защищенным или публичным. В случае защищенного доступа при запросе ресурса хранилища происходит проверка сессии и ролей пользователя. В случае публичного доступа - никаких дополнительных проверок не происходит, ресурсы с публичным доступом распространяются через CDN.

- Репозиторий приложения: <https://gitlab.mobbbtech.com/fsms/fsms-app>
- Репозиторий PHP-клиента: <https://gitlab.mobbbtech.com/fsms/fsms-client>

REST API

Host: fsms.iqoption.com

[POST] /api/v1/file

Создает файл в хранилище из переданного через параметр содержимого.

header:

| name | value |
|--------------|---------------------|
| Content-Type | multipart/form-data |

body: JSON

| name | type | optional | description | example |
|---------------|---------|-------------|---|--|
| filename | string | обязательно | Исходное имя файла. | my.ext |
| body | string | обязательно | Тело файла BLOB | |
| access | json | - | Список допустимых идентификаторов пользователей или ресурсов admin-platform | { "uids: [id1, ..., idN], resources: [{"path": "/tickets", "client_name": "admin"}, {"path": "...}]] |
| optimize_file | boolean | - | нужно ли оптимизировать файл (картинку), используется image_optim | |

| | | | | |
|-----|-----|---|--|--|
| ttl | int | - | Время жизни файла в секундах. Если не передается или передается 0, то TTL не задается. | |
|-----|-----|---|--|--|

returns: JSON

| Return: |
|---|
| <pre>{ "server": "http://server-base-url.com/", "uri": "path/to/file.ext", "ttl": 0 }</pre> |

[POST] /api/v1/file/from-multipart

Создает файл в хранилище из данных в формате multipart/form-data (см. RFC 2388)

header:

| name | value |
|--------------|---------------------|
| Content-Type | multipart/form-data |

body: JSON - Запрос должен содержать ровно один файл, прикрепленный средствами multipart/form-data

| name | type | optional | description | example |
|---------------|---------|-------------|---|---|
| filename | string | - | Исходное имя файла. Если не указано, используется оригинальное имя прикрепленного файла | my.ext |
| access | json | - | Список допустимых идентификаторов или ресурсов admin-platform | {uids: [id1, ..., idN], resources: [{"path": "/tickets", "client_name": "admin"}, {"path": "..."}]} |
| optimize_file | boolean | - | нужно ли оптимизировать файл (картинку), используется image_optim | |
| ttl | int | - | Время жизни файла в секундах. Если не передается или передается 0, то TTL не задается. | |
| file | file | обязательно | файл, прикрепленный средствами multipart/form-data | |

returns: JSON

Returns

```
{
  "server": "http://server-base-url.com/",
  "uri": "path/to/file.ext",
  "ttl": 0
}
```

[DELETE] /api/v1/file/{fileUri}

Удаляет файл из хранилища по его URI, параметр возвращаемый в ответе методов [POST, PUT] /v1/file, [POST, PUT] /v1/file/from-multipart

пример:DELETE

```
http://server-base-url.com/api/v1/file/path/to/file.ext
```

returns:JSON

```
{
  "result": "OK"
}
```

[PUT] /api/v1/file/{fileUri}

Замена файла по его URI. Данные необходимо отправлять в формате **"application/x-www-form-urlencoded"**

header:

| name | value |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

body:JSON

| name | type | optional | description | example |
|----------|--------|-------------|--------------------|---------|
| filename | string | обязательно | Исходное имя файла | |
| body | json | обязательно | Тело файла BLOB | |

| | | | | |
|---------------|---------|---|---|--|
| access | json | - | Список допустимых идентификаторов пользователей или ресурсов admin-platform Если файл закрыт параметром "access", то его невозможно заменить на публичный файл (без параметра "access") | {uids: [id1, ..., idN], resources: [{"path": "/tickets", "client_name" "admin"}, {"path": "...}]]} |
| optimize_file | boolean | - | нужно ли оптимизировать файл (картинку), используется image_optim | |
| ttl | int | - | Время жизни файла в секундах. Если не передается или передается 0, то TTL не задается. | |

returns:JSON

```
{
  "result": "OK"
}
```

[PUT] /api/v1/file/from-multipart/{fileUri}

Заменяет файл в хранилище на файл переданный в формате "multipart/form-data".

header:

| name | value |
|--------------|---------------------|
| Content-Type | multipart/form-data |

body:

| name | type | optional | description | example |
|---------------|---------|----------|---|--|
| filename | string | - | Исходное имя файла. Если не указано, используется оригинальное имя прикрепленного файла | |
| access | json | - | Список допустимых идентификаторов или ресурсов admin-platform Если файл закрыт параметром "access", то его невозможно заменить на публичный файл (без параметра "access") | {uids: [id1, ..., idN], resources: [{"path": "/tickets", "client_name" "admin"}, {"path": "...}]]} |
| optimize_file | boolean | - | нужно ли оптимизировать файл (картинку), используется image_optim | |

| | | | | |
|------|-------------|---|---|--|
| ttl | <i>int</i> | - | Время жизни файла в секундах. Если не передается или передается 0, то TTL не задается. | |
| file | <i>file</i> | - | файл, прикрепленный средствами multipart/form-data | |

returns:JSON

```
{
  "result": "OK"
}
```

[POST] /api/v1/image/from-multipart

Загружает изображение в хранилище с возможностью изменения размеров (resize) поворотом (rotate) и обрезания определенной области изображения (crop)

Данные необходимо отправлять в формате "multipart/form-data", в том числе параметры передаются в body-запроса

header:

| name | value |
|--------------|---------------------|
| Content-Type | multipart/form-data |

body:

| name | type | optional | description | example |
|--|---------------|-------------|---|---|
| filename | <i>string</i> | - | Исходное имя файла. Если не указано, используется оригинальное имя прикрепленного файла | |
| access | <i>json</i> | - | Список допустимых идентификаторов или ресурсов admin-platform | {uids: [id1, ..., idN], resources: [{"path": "/tickets", "client_name": "admin"}, {"path": "..."}]} |
| thumbs[] | <i>string</i> | - | Формат: (ширина)X(Длина) Изменение размеров изображения (resize) | thumbs[]=100x100 |
| rotate_angle | <i>int</i> | - | Угол наклона изображения по часовой стрелке. Crop выполняется после. | 30 |
| rotate_color | <i>string</i> | - | Цвет фона при повороте изображения, по умолчанию белый. | #000000 или green |
| file | <i>file</i> | обязательно | файл, прикрепленный средствами multipart/form-data | |
| Параметры для обрезки изображения (работает только при указании всех параметров): | | | | |
| crop_width | <i>int</i> | - | Ширина обрезания | |
| crop_height | <i>int</i> | - | Высота обрезания | |

| | | | | |
|--------|------------|---|---|--|
| crop_x | <i>int</i> | - | Координата X верхнего левого угла области обрезания | |
| crop_y | <i>int</i> | - | Координата Y верхнего левого угла области обрезания | |

returns:JSON

```
{
  "server": "http://server-base-url.com/",
  "uri": "path/to/file.ext",
  "crop_uri": "path/to/crop_file.ext",
  "thumbs": {
    "100x100": "path/to/thumbs_file.ext"
  }
}
```

Использование имени в ссылках

Можно передавать имя файла параметром, чтобы к нему обращаться, например, в ссылке изменить окончание:

server: "http://server-base-url.com/", uri: "path/to/file/**my**.ext".

Или использовать как параметр ?original_name=my.ext :

http://server-base-url.com/path/to/file.ext?original_name=my.ext