

Upload File

Endpoint

URL: `/:projectId/storage`

Method: `POST`

Authentication Required: Yes

Description

Upload a file to project storage. The file is saved using a path structure defined by `pathParts`. The filename is sanitized using transliteration.

Request

Headers

| Header | Type | Required | Description |
|----------------------------|---------------------|----------|--|
| <code>Authorization</code> | <code>string</code> | Yes | Bearer token for authentication. |
| <code>Content-Type</code> | <code>string</code> | Yes | Must be <code>multipart/form-data</code> . |

Body Parameters (multipart/form-data)

| Field | Type | Required | Description |
|-------------------|---------------|----------|---|
| <code>file</code> | File (binary) | Yes | The file to upload. Must not exceed 15MB. |

| Field | Type | Required | Description |
|------------------------|-------------------------------|----------|--|
| <code>pathParts</code> | <code>string</code> (JSON) | Yes | A JSON stringified array of folder path segments |

Example Request

```
POST /proj1234/storage
Authorization: Bearer <ACCESS_TOKEN>
Content-Type: multipart/form-data

Body:
- file: (binary file)
- pathParts: ["profile", "images"]
```

Response

Success Response (201 Created)

```
{
  "fileId": "abc123",
  "relativePath": "profile/images/photo.jpg",
  "publicPath": "https://cdn.example.com/profile/images/photo.jpg"
}
```

Error Responses

Missing File (400 Bad Request)

```
{
  "error": "No file provided",
  "code": "storage/no-file"
}
```

Invalid Path Parts (400 Bad Request)

```
{
```

```
"error": "pathParts must be an array of strings",
"code": "storage/invalid-path-parts"
}
```

File Too Large (413 Payload Too Large)

```
{
  "error": "File size exceeds the allowed limit of 15MB",
  "code": "storage/file-too-large"
}
```

Server Error (500 Internal Server Error)

```
{
  "error": "Internal server error",
  "code": "storage/server-error",
  "details": "<Error message>"
}
```

Notes

- The uploaded file is stored at the location specified by `pathParts`, with the filename sanitized.
- Returns a public URL to access the file.

Last updated on May 11, 2025