

Fleet File Sets API

Table of Contents

1. Intro	1
1.1. Version	1
1.2. About File Sets	2
1.3. File Types	2
1.4. Directories on the Controller	2
1.5. Slots	2
1.6. Flow	3
1.7. API General information	3
1.7.1. Authentication	3
2. API reference	4
2.1. API: File Sets - Get List	4
2.2. API: File Sets - Get by ID	6
2.3. API: File Sets - Create	7
2.4. API: File Sets - Update	8
2.5. API: File Sets - Delete	9
2.6. API: Files - Get All from File Set	9
2.7. API: Files - Add	10
2.8. API: Files - Update	12
2.9. API: Files - Delete	13
2.10. API: F-Controllers - Get Assigned File Sets	14
2.11. API: F-Controllers - Assign File Set	16
3. API Types	17
3.1. FileSet	17
3.2. FFile	17
3.3. FFileType	17

1. Intro

This documentation covers Fleet File Sets management API for F controllers. Before using the API, make sure the Fleet is set up and running.

1.1. Version

Fleet version **2.5.3** or higher is needed.

1.2. About File Sets

File Sets are the way to delivery files to F controllers. Each File Set ([FileSet](#)) contains multiple files ([FFile](#)) with one or different types (see [FFileType](#)). Multiple file sets can be assigned to the controller. All files from all assigned File Sets will be written to the controller during USB configuration or running of Update Plan.

1.3. File Types

File Set can contain only files of known types:

- **uiScriptBin** - binary Aspe script to be executed on the controller
- **uiScriptData** - data file that contains JKV - binary analog of JSON
- **screenImage** - image file like regular png, bmp or jpeg
- **screenFont** - screen font file in a special **fnt** format

1.4. Directories on the Controller

Each File Type lives in a separate directory on the controller:

File Type	Base Directory
uiScriptBin	/lfs1/asp/ui/scripts
uiScriptData	/lfs1/asp/ui/data
screenImage	/lfs1/screen/img
screenFont	/lfs1/screen/fonts

So if FFile has **type=screenImage** and **path=some_dir/image.jpg**, then real destination on the controller will be **/lfs1/screen/img/some_dir/image.jpg**

1.5. Slots

When File Set assigns to the controller, unique slot should be specified (just a string label). This is useful when controller has multiple File Sets (for example, one set with images, one with fonts and one with UI script). When you want to replace images file set with the new one, slot helps to define which file set should be replaced. So if you have:

Slot	File Set
my-images	file set #1
my-fonts	file set #2
any-unique-string	file set #3

You can call

```
curl -XPUT /f-controllers/123/file-sets/slots/my-images \
```

```
--data='{ "fileSet": 4 }'
```

to replace file set #1 to file set #4 for slot **my-images** for controller id=123

1.6. Flow

File Sets can be created, assigned to controllers, updated and removed via UI or API.

Let's imagine, you need to upload couple scripts, data files, and images to 3 controllers. Common steps that should be done:

- Create necessary File Sets. It's possible to have only one File Set and upload all file types there (scripts, data, images). But it's better to create separate sets, it can reduce file duplicates and make update process smoother.
- Upload files no necessary File Sets
- Assign File Sets to necessary controllers
- Configure controllers via USB or schedule remote updates via Update Plans
- Done. All files are uploaded to the controllers

If you want to update some data (for example, replace one of the images to the new one), you can do it in two ways:

1. Directly edit file in the File Set. The changes will be uploaded to the controllers on nearest USB configuration or scheduled remote Updates
2. Create new File Set with images (for example, name it "Images v2") that contain all correct images. Then replace the old File Set with the new one on controllers. The benefit is that you can replace images version not for all controllers at once, but only for some test group at the beginning, also it's easy to rollback to the old version.

1.7. API General information

Fleet has REST API for Lockers and Reservations management.

API supports JSON format for request bodies and responses for all the endpoints.

2XX http codes mean success, 4XX - problem in request, 5XX - problem on server.

1.7.1. Authentication

Each request should have an authentication token, passed via **api-key** header:

```
curl --header 'api-key: abc123' \  
https://prod.jetbeep.com/fleet/api/pub/lockers/active
```

To obtain the api key, you need to create a user with role=API on the Fleet UI and then create the

api key on the user's page.

2. API reference

2.1. API: File Sets - Get List

Return paginated File Sets list.

Request

GET /f-file-sets

Query Parameters:

- **perPage** (number, optional, default=100) - number of items per page
- **page** (number, optional, default=1) - page started from 1
- **orderBy** (string, optional, default="id") - field to sort by
- **direction** (enum, optional, default="desc") - sort direction. Possible values: "asc", "desc"
- **filter** (json-string, optional) - filter results. Format `encodeURIComponent({ "group": "images", "customField1": "some-value" })`

Request examples:

```
# basic
curl 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets' \
  --header 'api-key: abc123'

# sorting and pagination
curl 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets?orderBy=updatedAt&direction=asc&perPage=30' \
  --header 'api-key: abc123'

# filter by { "updatedAt": { "operator": ">=", "value": "2024-10-22T11:31:54.400Z" } }
curl 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets?filter=%7B%22updatedAt%22%3A%7B%22operator%22%3A%22%3E%3D%22%2C%22value%22%3A%222024-10-22T11%3A31%3A54.400Z%22%7D%7D' \
  --header 'api-key: abc123'
```

Response

Http-code: 200

Response fields:

- **rows** ([FileSet\[\]](#)) - list of File Sets
- **pagination** (object) - pagination parameters
 - **totalCount** (number) - items count for all pages
 - **currentCount** (number) - items received in response

- **page** (number) - current page number (starts from 1)
- **perPage** (number) - items count per page
- **sorting** (object) - sorting parameters
 - **field** (string) - current sorting field
 - **direction** (enum) - sorting direction. Possible values: "asc", "desc"

Response example:

```
{
  "rows": [
    {
      "id": 122,
      "projectId": 19,
      "name": "Auto upload files",
      "crc32": 3090283660,
      "createdBy": 1,
      "createdAt": "2024-12-11T13:19:06.000Z",
      "updatedAt": "2024-12-11T16:45:53.000Z",
      "group": "custom",
      "customField1": "",
      "customField2": "",
      "readonly": false,
      "screenMetadata": null
    },
    {
      "id": 117,
      "projectId": 19,
      "name": "Images v2",
      "crc32": 569086670,
      "createdBy": 71,
      "createdAt": "2024-12-05T12:25:07.000Z",
      "updatedAt": "2024-12-05T12:25:08.000Z",
      "group": "images",
      "customField1": "844a3c0d",
      "customField2": "",
      "readonly": true,
      "screenMetadata": {
        "bitmaps": "844a3c0d"
      }
    },
    {
      "id": 116,
      "projectId": 19,
      "name": "Images v1",
      "crc32": 140309667,
      "createdBy": 71,
      "createdAt": "2024-07-22T12:36:57.000Z",
      "updatedAt": "2024-07-30T10:20:25.000Z",
      "group": "images",
```

```

        "customField1": "3566dc4a",
        "customField2": "",
        "readonly": true,
        "screenMetadata": {
            "bitmaps": "3566dc4a"
        }
    },
    ],
    "pagination": {
        "totalCount": 25,
        "currentCount": 3,
        "page": 1,
        "perPage": 3
    },
    "sorting": {
        "field": "createdAt",
        "direction": "asc"
    }
}

```

2.2. API: File Sets - Get by ID

Request

GET /f-file-sets/:id

Path parameters:

- **:id** - File Set ID

Request examples:

```

# basic
curl 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets/127' \
  --header 'api-key: abc123'

```

Response

Http-code: 200

Response fields:

[FileSet](#) model

Response example:

```

{
  "id": 127,
  "projectId": 19,
  "name": "Images v2",
  "crc32": 1308551952,

```

```

    "createdBy": 80,
    "createdAt": "2025-03-10T07:23:25.742Z",
    "updatedAt": "2025-03-10T08:27:44.931Z",
    "group": "images",
    "customField1": "844a3c0d",
    "customField2": "test",
    "readonly": false,
    "screenMetadata": {
      "bitmaps": "844a3c0d"
    }
  }
}

```

2.3. API: File Sets - Create

Request

POST /f-file-sets

Request fields:

- **name** (string, required) - File Set name
- **group** (string, required) - File Set group, can be any string
- **readonly** (boolean, required) - if true, file set and files will not be editable
- **customField1** (string, optional) - any user data
- **customField2** (string, optional) - any user data
- **screenMetadata** (object, optional) - key-value metadata for controller screen

Request example:

```

curl -X POST 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets' \
  --header 'api-key: abc123' \
  --data-raw '{
    "name": "Images v2",
    "group": "images",
    "customField1": "844a3c0d",
    "readonly": false,
    "screenMetadata": {
      "bitmaps": "844a3c0d"
    }
  }'

```

Response

Http-code: 201

Response fields:

- **FileSet** model

Response example:

```
{
  "id": 127,
  "projectId": 19,
  "name": "Images v2",
  "crc32": null,
  "createdBy": 80,
  "createdAt": "2025-03-10T07:23:25.742Z",
  "updatedAt": "2025-03-10T07:23:25.742Z",
  "group": "images",
  "customField1": "844a3c0d",
  "customField2": null,
  "readonly": false,
  "screenMetadata": {
    "bitmaps": "844a3c0d"
  }
}
```

2.4. API: File Sets - Update

Request

`PATCH /f-file-sets/:id`

Request fields:

The same as for [API: File Sets - Create](#), but all fields are optional

Request example:

```
curl -X PATCH 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets/127' \
  --header 'api-key: abc123' \
  --data-raw '{
    "customField2": "test"
  }'
```

Response

Http-code: 200

Response fields:

- [FileSet](#) model

Response example:

```
{
  "id": 127,
  "projectId": 19,
  "name": "Images v2",
```



```
"crc32": null,  
"createdBy": 80,  
"createdAt": "2025-03-10T07:23:25.742Z",  
"updatedAt": "2025-03-10T07:29:02.180Z",  
"group": "images",  
"customField1": "844a3c0d",  
"customField2": "test",  
"readonly": false,  
"screenMetadata": {  
  "bitmaps": "844a3c0d"  
}  
}
```

2.5. API: File Sets - Delete

Request

`DELETE /f-file-sets/:id`

Request fields:

No request fields

Request example:

```
curl -X DELETE 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets/128' \  
--header 'api-key: abc123'
```

Response

Http-code: 200

Response fields:

No response body

2.6. API: Files - Get All from File Set

Get list of all files from File Set with or without file content

Request

`GET /f-file-sets/127/files`

Request examples:

```
# basic  
curl 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets/127/files' \  
--header 'api-key: abc123'
```

Response

Http-code: 200

Response fields:

List of [FFile](#)

Response example:

Files without content:

```
[
  {
    "id": 1474,
    "setId": 127,
    "type": "uiScriptBin",
    "path": "main.aspe",
    "crc32": 63513495,
    "createdBy": 80,
    "createdAt": "2025-03-10T08:50:52.052Z",
    "updatedAt": "2025-03-10T08:50:52.052Z"
  },
  {
    "id": 1471,
    "setId": 127,
    "type": "uiScriptData",
    "path": "my-dir/data.jkv",
    "crc32": 263154844,
    "createdBy": 80,
    "createdAt": "2025-03-10T07:59:21.775Z",
    "updatedAt": "2025-03-10T07:59:21.775Z"
  }
]
```

2.7. API: Files - Add

Add file to the File Set

Request

POST /f-file-sets/files

Request fields:

- **setId** (number, required) - File Set ID
- **type** ([FFileType](#), required) - File Type
- **path** (string, required) - File path relative to [Directories on the Controller](#)
- **dataText** (string, conditional) - Text file content, relevant for [uiScriptData](#) file type
- **dataBinStr** (string, conditional) - Binary and serialized file content. "hex" or "base64" serialization can be used (see [binSerializer](#) request parameter). relevant for [uiScriptBin](#), [screenImage](#), [screenFont](#) file types.

- `binSerializer` (string, conditional) - Defines the format of `dataBinStr` content. Available values are `hex`, `base64`.

Request examples:

```
# Create uiScriptData file from JSON string.
# On the server JSON will be converted to JKV - format supported by controller
curl -X POST 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets/files' \
  --header 'api-key: abc123' \
  --data-raw '{
    "setId": 127,
    "type": "uiScriptData",
    "path": "my-dir/data.jkv",
    "dataText": "{\"key\":\"value\",\"hello\":true}"
  }'

# Create screenImage file from base64 serialized image file
curl -X POST 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets/files' \
  --header 'api-key: abc123' \
  --data-raw '{
    "setId": 127,
    "type": "screenImage",
    "path": "picture1.jpg",
    "dataBinStr":
    "iVBORw0KGgoAAAANSUhEUgAAB3UAAAA0CAIAAA9AfvkAAACXBIWXMAABJ...",
    "binSerializer": "base64"
  }'

# Create uiScriptBin file from HEX serialized script file
curl -X POST 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets/files' \
  --header 'api-key: abc123' \
  --data-raw '{
    "setId": 127,
    "type": "uiScriptBin",
    "path": "main.aspe",
    "dataBinStr":
    "c2431644243dbee5a7d6973154a6bd3c76dc42369196698a28b9013327391155...",
    "binSerializer": "hex"
  }'
```

Response

Http-code: 201

Response fields:

- [FFile](#) model

Response example:

```
{
  "id": 1472,
  "setId": 127,
  "type": "screenImage",
  "path": "picture1.jpg",
  "crc32": 3152520141,
  "createdBy": 80,
  "createdAt": "2025-03-10T08:04:20.803Z",
  "updatedAt": "2025-03-10T08:04:20.803Z"
}
```

2.8. API: Files - Update

Update existing file

Request

PATCH /f-file-sets/files/:id

Request fields:

- **path** (string, optional)
- **dataText** (string, conditional)
- **dataBinStr** (string, conditional)
- **binSerializer** (string, conditional)

Meaning of fields is the same as for [API: Files - Add](#).

Request examples:

```
curl -X PATCH 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets/files/1473' \
  --header 'api-key: abc123' \
  --data-raw '{
    "path": "main2.aspe",
    "dataBinStr":
"c2431644243dbee5a7d6973154a6bd3c76dc42369196698a28b9013327391155...",
    "binSerializer": "hex"
  }'
```

Response

Http-code: 200

Response fields:

- **file** ([FFile](#), required)
- **set** ([FileSet](#), required)

Response example:

```
{
  "set": {
    "id": 127,
    "projectId": 19,
    "name": "Images v2",
    "crc32": 111222333,
    "createdBy": 80,
    "createdAt": "2025-03-10T07:23:25.742Z",
    "updatedAt": "2025-03-10T08:09:58.539Z",
    "group": "images",
    "customField1": "844a3c0d",
    "customField2": "test",
    "readonly": false,
    "screenMetadata": {
      "bitmaps": "844a3c0d"
    }
  },
  "file": {
    "id": 1473,
    "setId": 127,
    "type": "uiScriptBin",
    "path": "main2.aspe",
    "crc32": 3806588089,
    "createdBy": 80,
    "createdAt": "2025-03-10T08:09:58.526Z",
    "updatedAt": "2025-03-10T08:25:28.414Z"
  }
}
```

2.9. API: Files - Delete

Request

`DELETE /f-file-sets/files/:id`

Request fields:

No request fields

Request example:

```
curl -X DELETE 'https://dev.jetbeep.com/fleet/api/pub/f-file-sets/files/1473' \
  --header 'api-key: abc123'
```

Response

Http-code: 200

Response fields:

- **set** (**FileSet**, required)

Response example:

```
{
  "set": {
    "id": 127,
    "projectId": 19,
    "name": "Images v2",
    "crc32": 1234567800,
    "createdBy": 80,
    "createdAt": "2025-03-10T07:23:25.742Z",
    "updatedAt": "2025-03-10T08:27:44.931Z",
    "group": "images",
    "customField1": "844a3c0d",
    "customField2": "test",
    "readonly": false,
    "screenMetadata": {
      "bitmaps": "844a3c0d"
    }
  }
}
```

2.10. API: F-Controllers - Get Assigned File Sets

Get all File Sets assigned to the controller

Request

GET /f-controllers/:controllerId/file-sets

Path parameters:

- **:controllerId** - Controller ID

Request examples:

```
# basic
curl 'https://dev.jetbeep.com/fleet/api/pub/f-controllers/2/file-sets' \
  --header 'api-key: abc123'
```

Response

Http-code: 200

Response fields:

Object where key is slot name, value is **FileSet** with all associated files in **files** field.

Response example:

```
{
```

```

"app-images": {
  "id": 87,
  "projectId": 19,
  "name": "Images #2ea2f26d",
  "crc32": 4132554010,
  "createdBy": 71,
  "createdAt": "2024-08-19T08:26:56.000Z",
  "updatedAt": "2024-09-03T15:28:57.000Z",
  "group": "images",
  "customField1": "2ea2f26d",
  "customField2": "",
  "readonly": true,
  "screenMetadata": {
    "bitmaps": "2ea2f26d"
  },
  "files": [
    {
      "id": 726,
      "setId": 87,
      "type": "screenImage",
      "path": "img1.jpg",
      "crc32": 509567301,
      "createdBy": 71,
      "createdAt": "2024-08-19T10:26:56+02:00",
      "updatedAt": "2024-08-19T10:26:56+02:00"
    },
    {
      "id": 728,
      "setId": 87,
      "type": "screenImage",
      "path": "img2.jpg",
      "crc32": 2063647239,
      "createdBy": 71,
      "createdAt": "2024-08-19T10:26:56+02:00",
      "updatedAt": "2024-08-19T10:26:56+02:00"
    }
  ]
},
"app-scriptData": {
  "id": 86,
  "projectId": 19,
  "name": "Script Data #a0ae550d",
  "crc32": 173236107,
  "createdBy": 71,
  "createdAt": "2024-08-19T08:11:16.000Z",
  "updatedAt": "2024-08-19T08:11:19.000Z",
  "group": "script",
  "customField1": "a0ae550d",
  "customField2": null,
  "readonly": true,
  "screenMetadata": null,

```

```

    "files": [
      {
        "id": 694,
        "setId": 86,
        "type": "uiScriptData",
        "path": "data1.jkv",
        "crc32": 1416766020,
        "createdBy": 71,
        "createdAt": "2024-08-19T10:11:18+02:00",
        "updatedAt": "2024-08-19T10:11:18+02:00"
      },
      {
        "id": 690,
        "setId": 86,
        "type": "uiScriptData",
        "path": "data2.jkv",
        "crc32": 3616177068,
        "createdBy": 71,
        "createdAt": "2024-08-19T10:11:18+02:00",
        "updatedAt": "2024-08-19T10:11:18+02:00"
      }
    ]
  }
}

```

2.11. API: F-Controllers - Assign File Set

Assign file set to the F-Controller to the specific slot. If another File Set already assigned to this slot, it will be replaced with the new File Set.

Request

PUT /f-controllers/:id/file-sets/slots/:slot

Path parameters:

- **:id** - Controller ID
- **:slot** - Slot for file set

Request fields:

- **fileSet** (number, required) - File Set ID

Request example:

```

curl -X PUT 'https://dev.jetbeep.com/fleet/api/f-controllers/2/file-sets/slots/app-images' \
  --header 'api-key: abc123' \
  --data-raw '{
    "fileSet": 127
  }'

```


Response

Http-code: 200

Response fields:

No response body

3. API Types

3.1. FileSet

```
export interface FileSet {  
  id: number;  
  projectId: number;  
  name: string;  
  crc32: number;  
  createdBy: number;  
  createdAt: string;  
  updatedAt: string;  
  group: string;  
  customField1?: string;  
  customField2?: string;  
  readonly: boolean;  
  screenMetadata?: AnyObject;  
}
```

3.2. FFile

```
export interface FFile {  
  id: number;  
  setId: number;  
  type: FFileType;  
  path: string;  
  crc32: number;  
  createdBy: number;  
  createdAt: string;  
  updatedAt: string;  
}
```

3.3. FFileType

```
export enum FFileType {  
  uiScriptBin = 'uiScriptBin',  
  uiScriptData = 'uiScriptData',  
}
```

```
screenImage = 'screenImage',  
screenFont = 'screenFont',  
}
```