

Fleet Update Plans API

Table of Contents

1. Intro	1
1.1. Version	1
1.2. Update flow	1
1.3. Update Plan Status Transition Diagram	2
1.4. API General information	2
1.4.1. Authentication	2
2. API reference	3
2.1. API: Get Update Plans List	3
2.2. API: Create Update Plan	7
2.3. API: Activate Update Plan	9
3. API Types	10
3.1. FUpdatePlan	10
3.2. FUpdateStatus	10
3.3. FUpdateConfig	10
3.4. FUpdateType	11
3.5. FUpdateFwTargets	11
3.6. Other	11

1. Intro

This documentation covers Fleet Update Plans API for F-controllers that allows applying settings, files, and firmware updates to the controller. Before using the API, make sure the Fleet is set up and running.

1.1. Version

The documentation covers Fleet version **2.4.4** or higher.

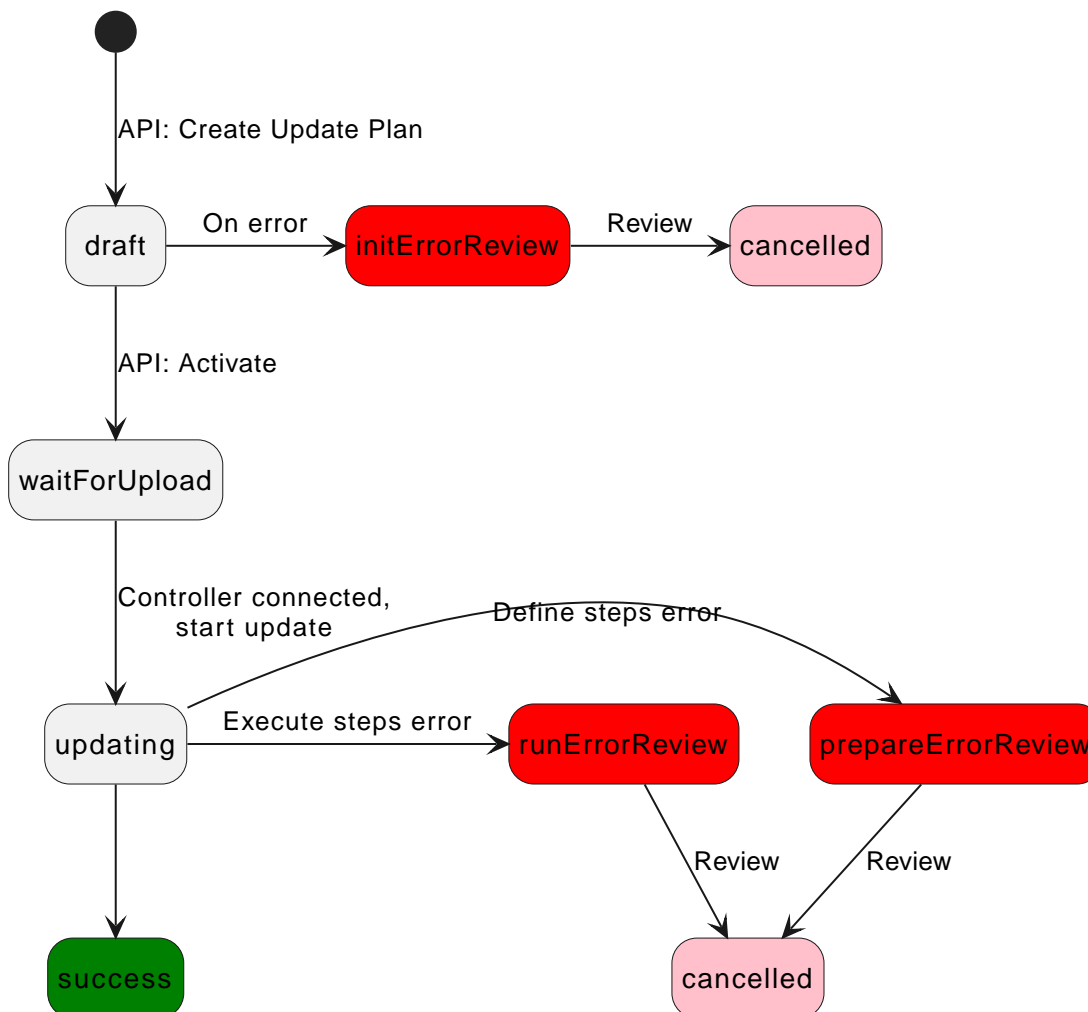
1.2. Update flow

- Make needed changes in the controller settings or assigned file sets via UI or API
- Create update plan via **POST /f-updates** with necessary parameters. It will be created in status "draft"
- Activate update plan via **POST /f-updates/:id/activate**. It will set "waitForUpload" status, so update will start when controller will be connected to the Fleet (according to Update Type)
- When update starts, Fleet defines steps to run, including device settings sync, files sync, screen

sync, firmware update, etc.

- Then Fleet reboots controllers in maintenance mode and run steps one by one. Detailed logs can be found in UI
- When update finished, status becomes "success" and controller reboots again to leave maintenance mode
- If problems occur, update plan will receive status "prepareErrorReview" or "runErrorReview". It means, that user should review what's wrong and then cancel the update

1.3. Update Plan Status Transition Diagram



1.4. 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.

1.4.1. 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: Get Update Plans List

Return paginated update plans list.

Request

GET /f-updates

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({ "updatedAt": {"operator": ">=", "value": "2024-10-22T11:31:54.400Z" } })`

Request examples:

```
# basic  
curl 'https://dev.jetbeep.com/fleet/api/pub/f-updates' \  
  --header 'api-key: abc123'  
  
# sorting and pagination  
curl 'https://dev.jetbeep.com/fleet/api/pub/f-updates?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-updates?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%22%7D' \  
  --header 'api-key: abc123'
```

Response

Http-code: 200

Response fields:

- **rows** (**FU**updatePlan[]) - list of update plans
- **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": 31,
      "projectId": 19,
      "controllerId": 1,
      "comment": "",
      "status": "draft",
      "fwUpdates": {
        "mainFw": "1.1.3+0"
      },
      "customField1": "",
      "customField2": "",
      "config": {
        "type": "sleep",
        "syncDeviceSettings": "auto",
        "syncFiles": "auto",
        "syncScreen": "auto"
      },
      "createdBy": 1,
      "createdAt": "2025-01-16T09:37:24.000Z",
      "updatedAt": "2025-01-16T09:37:24.000Z",
      "steps": null
    },
    {
      "id": 30,
      "projectId": 19,
      "controllerId": 1,
      "comment": "",
      "status": "runErrorReview",
      "fwUpdates": {},
      "customField1": "",
      "customField2": "",
      "config": {
```

```

    "type": "urgent",
    "syncDeviceSettings": "auto",
    "syncFiles": "auto",
    "syncScreen": "auto"
  },
  "createdBy": 1,
  "createdAt": "2024-12-19T14:45:52.000Z",
  "updatedAt": "2024-12-19T14:47:04.000Z",
  "steps": [
    {
      "id": 1,
      "type": "deviceSettings",
      "runIndex": 0,
      "status": "skip",
      "oldHash": 1521692067,
      "newHash": 1521692067,
      "syncMode": "auto",
      "msg": "Skip due to meta match"
    },
    {
      "id": 2,
      "type": "filesSync",
      "runIndex": 1,
      "status": "successWithWarnings",
      "syncMode": "auto",
      "startedAt": "2024-12-19T14:46:02.314Z",
      "stats": {
        "alreadyPresent": 61,
        "added": 1,
        "replaced": 19,
        "deleted": 0,
        "errors": 0
      },
      "endedAt": "2024-12-19T14:46:52.680Z",
      "warning": "Found 16 file duplicates"
    },
    {
      "id": 3,
      "type": "screenWrite",
      "runIndex": 5,
      "status": "error",
      "pluggedScreenMeta": {
        "all": 593713086,
        "fnt": "1.0.9",
        "bitmaps": "a119de5b"
      },
      "fleetScreenMeta": {
        "all": 3445825369,
        "fnt": "1.0.9",
        "bitmaps": "444111"
      }
    },
  ],

```

```

        "syncMode": "auto",
        "startedAt": "2024-12-19T14:46:52.693Z",
        "error": "[Error]: [RETURN_CODE_INTERNAL_ERROR] Error occurred",
        "endedAt": "2024-12-19T15:47:04+01:00"
    }
}
],
{
    "id": 29,
    "projectId": 19,
    "controllerId": 1,
    "comment": "",
    "status": "success",
    "fwUpdates": {},
    "customField1": "",
    "customField2": "",
    "config": {
        "type": "urgent",
        "syncDeviceSettings": "auto",
        "syncFiles": "auto",
        "syncScreen": "auto"
    },
    "createdBy": 1,
    "createdAt": "2024-12-19T14:43:21.000Z",
    "updatedAt": "2024-12-19T14:43:50.000Z",
    "steps": [
        {
            "id": 1,
            "type": "deviceSettings",
            "runIndex": 0,
            "status": "skip",
            "oldHash": 1521692067,
            "newHash": 1521692067,
            "syncMode": "auto",
            "msg": "Skip due to meta match"
        },
        {
            "id": 2,
            "type": "filesSync",
            "runIndex": 1,
            "status": "success",
            "syncMode": "auto",
            "startedAt": "2024-12-19T14:43:32.144Z",
            "stats": {
                "alreadyPresent": 64,
                "added": 1,
                "replaced": 0,
                "deleted": 0,
                "errors": 0
            },
            "endedAt": "2024-12-19T14:43:48.637Z"
        }
    ]
}

```

```

    },
    {
      "id": 3,
      "type": "screenWrite",
      "runIndex": 1,
      "status": "success",
      "pluggedScreenMeta": {
        "all": 593713086,
        "fnt": "1.0.9",
        "bitmaps": "a119de5b"
      },
      "fleetScreenMeta": {
        "all": 593713086,
        "fnt": "1.0.9",
        "bitmaps": "a119de5b"
      },
      "syncMode": "auto",
      "startedAt": "2024-12-19T14:43:48.647Z",
      "endedAt": "2024-12-19T14:43:50.122Z"
    }
  ]
},
"pagination": {
  "totalCount": 25,
  "currentCount": 3,
  "page": 1,
  "perPage": 3
},
"sorting": {
  "field": "createdAt",
  "direction": "asc"
}
}

```

2.2. API: Create Update Plan

Update Plan will be created in "draft" status, so it will not be run until activation.

Request

POST /f-updates

Request fields:

- **controllerId** (number, required) - controller to update
- **customField1** (string, optional) - any user data
- **customField2** (string, optional) - any user data
- **comment** (string, optional) - comment for update plan

- `config` (`FUpdateConfig`, optional) - update config
- `fwUpdates` (`FUpdateFwTargets`, optional) - firmware updates to apply

Request example:

```
curl -X POST 'https://dev.jetbeep.com/fleet/api/pub/f-updates' \  
  --header 'api-key: abc123' \  
  --data-raw '{  
    "controllerId": 1,  
    "customField1": "test",  
    "comment": "API demo",  
    "config": {  
      "type": "sleep"  
    },  
    "fwUpdates": {  
      "mainFw": "1.2.3+0"  
    }  
  }'
```

Response

Http-code: 201

Response fields:

- `FUpdatePlan` model

Response example:

```
{  
  "id": 32,  
  "projectId": 19,  
  "controllerId": 1,  
  "comment": "API demo",  
  "status": "draft",  
  "fwUpdates": {  
    "mainFw": "1.2.3+0"  
  },  
  "customField1": "test",  
  "customField2": null,  
  "config": {  
    "type": "sleep"  
  },  
  "createdBy": 80,  
  "createdAt": "2025-01-16T11:57:39.000Z",  
  "updatedAt": "2025-01-16T12:07:40.000Z",  
  "steps": null  
}
```


2.3. API: Activate Update Plan

Set status "waitForUpload" for "draft" update plan, so it will run on nearest controller connection.

NOTE Update plan should have `config.type` before activation

Request

POST /f-updates/:id/activate

Request fields:

No request fields

Request example:

```
curl -X POST 'https://dev.jetbeep.com/fleet/api/pub/f-updates/32/activate' \
  --header 'api-key: abc123'
```

Response

Http-code: 200

Response fields:

- [FUpdatePlan](#) model

Response example:

```
{
  "id": 32,
  "projectId": 19,
  "controllerId": 1,
  "comment": "API demo",
  "status": "waitForUpload",
  "fwUpdates": {
    "mainFw": "1.2.3+0"
  },
  "customField1": "test",
  "customField2": null,
  "config": {
    "type": "sleep"
  },
  "createdBy": 80,
  "createdAt": "2025-01-16T11:57:39.000Z",
  "updatedAt": "2025-01-16T12:07:40.000Z",
  "steps": null
}
```

3. API Types

3.1. FUpdatePlan

```
export interface FUpdatePlan {  
  id: number;  
  projectId: number;  
  controllerId: number;  
  comment: string | null;  
  status: FUpdateStatus;  
  fwUpdates: FUpdateFwTargets | null;  
  customField1: string | null;  
  customField2: string | null;  
  config: FUpdateConfig | null;  
  steps: any[] | null;  
  createdBy: number | null;  
  createdAt: string;  
  updatedAt: string;  
}
```

3.2. FUpdateStatus

```
export enum FUpdateStatus {  
  draft = 'draft',  
  initErrorReview = 'initErrorReview',  
  waitForUpload = 'waitForUpload',  
  
  prepareErrorReview = 'prepareErrorReview',  
  
  updating = 'updating',  
  
  success = 'success',  
  runErrorReview = 'runErrorReview',  
  cancelled = 'cancelled',  
}
```

3.3. FUpdateConfig

```
export interface FUpdateConfig {  
  type?: FUpdateType;  
  
  // time window to start update, actual for type="time"  
  time?: {  
    start: [number, number]; // hours, minutes  
    end: [number, number]; // hours, minutes  
  }  
}
```

```
};

syncFiles?: FSyncMode;
syncScreen?: FSyncMode;
syncDeviceSettings?: FSyncMode;
formatFs?: boolean;
}
```

3.4. FUpdateType

```
export enum FUpdateType {
  // run update immediately after controller connect
  urgent = 'urgent',

  // run update when controller connected and after UI script finish
  sleep = 'sleep',

  // run update on particular time. Controller connect will be scheduled on that time
  time = 'time',
}
```

3.5. FUpdateFwTargets

```
export interface FUpdateFwTargets {
  // main board version to update
  mainFw?: string;
}
```

3.6. Other

```
export enum FSyncMode {
  // do sync if needed (if some changes found)
  auto = 'auto',

  // do sync in any case
  force = 'force',

  // skip sync in any case
  skipSync = 'skipSync',
}
```