IFS Aviation Maintenance

Line Planner API Guide

ICAM 26.2



Limited Availability Release

Document Revision: 26.2-1

Publication Date: November 02, 2026

Contents

1 Introduction....4
Line Planner API overview....4

2 Setting up Line Planner API....5
Licensing and subscription....5
Permissions and authentication....5
Assign the Line Planner integration role to a user....5
Give Line Planner integration permission to a user role....6
Update the Maintenix server properties....6
Activate alerts....7
Configure batch upload jobs for the line planner application....8

3 Line Planner API....12
Inbound messages....12
Inbound message samples....16
Schedule tasks to a new work package....16
Schedule tasks to an existing work package....19
Unschedule tasks....21
Override work package name....22
Prioritize tasks....23
Commit work packages....25
Request parts for work packages....26
Adjust work package schedules....28
Batch upload processes....29
Synchronization processes....32
Batch upload samples....34
Clean task data....35
Flight data....36
Generate availability data....39
Generate capability data....40
Generate reprocess task batch data....43
Generate task batch data....50

Appendix A Line Planner Authentication....62

Limited Availability Release

Appendix B Error Handling..... 64

Limited Availability Release

1 Introduction

Line Planner API overview

The Line Planner APIs enable a third-party line planning application to interact with Maintenix.

The Line Planner APIs allow third-party line planning applications to facilitate management of maintenance tasks, including:

- creating and updating work packages.
- assigning tasks.
- committing work packages.
- requesting parts for work packages.
- adjusting work package schedules.

The Line Planner APIs also provide batch upload job processes to ensure tasks, associated work packages, aircraft, and flight data are always up-to-date in the line planner application. This data is necessary for line planners to see ground time and where to plan maintenance. Line execution uses this information for the production team, so that they are ready to meet the aircraft when it lands.

Maintenix batch jobs are configured to run at set intervals or they can be triggered manually. These trigger a synchronous process where new data is queried from the Maintenix database and pushed to the line planning application via HTTP REST endpoints.

Limited Availability Release

2

Setting up Line Planner API

The following section provides setup and configuration information for the line planner APIs.

Licensing and subscription

To use the Line Planner APIs, your organization must hold a valid license for it.

That means the `LINE_PLANNER_API` feature must have been added to your Maintenix `license.xml`. If your current license does not include this feature, contact your Maintenix representative.

Note `license.xml`, resides in the `<install-dir>/domain/config/maintenix` folder, on the Maintenix application server host.

Permissions and authentication

The Line Planner Integration role handles your line planner application permissions.

The line planner installation automatically creates a default user called `lpintegration` with the LPINTEG role, which gives the user the permissions to authenticate and authorize inbound messages from the line planner application to Maintenix. Any other user created for this purpose should be assigned the LPINTEG role. No further permissions configuration is required.

Assign the Line Planner integration role to a user

You can grant specific users permission to use a line planner application.

Steps

1. Go to the **Role Search** page.

Limited Availability Release

2. Enter LPINTEG in the **Role Code** field and click **Search**.
3. On the **Role Details** page, **Users** tab, click **Assign User**.

> **Tip** If the user already appears on the **Role Details** page, select the user from the list and click **Assign User**. Otherwise, follow the steps below.

4. On the **User Search** page, do any of the following:
   - Click **Search** to see a list of all users in the system.
   - Enter the complete or partial name of the user in any of the user name fields and click **Search**.
5. Select a user from the list.
6. Click **Assign User**.

Give Line Planner integration permission to a user role

You can grant users, with a specific role, permission to use a line planner application.

Steps

1. Go to the **Role Search** page.
2. On the **Role Search** page, do any of the following:
   - Click **Search** to see a list of all roles in the system.
   - Enter the code in the **Role Code** field and click **Search**.
   - Enter the complete or partial name of the role in the **Name** field and click **Search**.
3. Select a role from the list.
4. On the **Role Details** page, click the **Permission Sets** tab.
5. Under the **Maintenix API Extensions** setting, select the **Line Planner API** check box.
6. Click **OK**.

Update the Maintenix server properties

Third-party line planner integration requires update of the `mx-servers.properties` file with your account information.

Steps

1. Navigate to the `/mxi/apps/assetmanagement/resource/property/` folder where your instance of Maintenix is installed.
2. Open the `mx-servers.properties` file in a text editor.

Limited Availability Release

3. Update the following properties with your account information. Note that the following are examples that will need to be modified:

```
linePlanner.url={URL of the line planner application}
mxSaasAuth.url={URL of the authorization application}
mxSaasAuth.retryPolicy.delay={The delay in between retry attempts,
 measured in seconds}
mxSaasAuth.retryPolicy.maxRetries={The maximum number of retries
 before returning a failure}
linePlanner.retryPolicy.delay={The delay in between retry attempts,
 measured in seconds}
linePlanner.retryPolicy.maxRetries={The maximum number of retries
 before returning a failure}
linePlanner.integration.username={user name associated with the line
 plannner application}
maintenix.tenant=maintenix-integration
```

4. Type the password associated with the user account for logging onto your line planner application.

```
linePlanner.integration.password={password associated with the line
 planner application}
```

Note When setting the password in the text editor, the password will be encrypted when Maintenix is restarted. To have the password automatically encrypted as you enter it, skip this step and set the password by using the password encryption utility. See the *Maintenix Operator Installation and Configuration Guide* for more details.

5. Save and close the file.
6. Restart the server.

Important You must restart your server for changes to properties files to take effect.

Activate alerts

To identify when a job fails, the Job Failure alert must be activated.

Steps

1. Go to the **Alerts Setup** page.
2. Click the **Category** drop-down list and select "JOB".
3. Select the check box beside the Job Failure alert.

4. Click **Activate Alert**.

> **Tip** You can edit the roles assigned to alerts by selecting the check box beside the Job Failure alert and clicking **Edit Role Assignments**. From this page you can select or clear roles as required.

Configure batch upload jobs for the line planner application

You can configure the batch upload jobs by setting an initial date for extracting of flight information, by activating the jobs that you want to run automatically, and by specifying relevant parameters. When a job is active, it runs at set intervals which can be configured by adjusting the REPEAT_INTERVAL attribute.

About this task

All tables that are referenced in the following steps are located in the Maintenix database.

Steps

1. Go to the EXT_BATCH_UPLOAD table.
2. Set an initial value for the LAST_SUCCESS_END_DT parameter of the EXT_BATCH_UPLOAD_GENERATE_TASK_BATCH_DATA job.

   The LAST_SUCCESS_END_DT parameter is the date that will be used to extract all flight information on the first run of the job. After successful run of the job, the LAST_SUCCESS_END_DT is updated in the EXT_BATCH_UPLOAD table. It becomes the timestamp of the data that was last uploaded successfully. When the job runs next time, it will look only for new data or data that changed from that date onwards.

   > **Important** This step is mandatory. If you don't set up an initial value for LAST_SUCCESS_END_DT, the job returns an empty result, and nothing is sent for the first run.

3. Go to the UTL_JOB table.
4. If you want any of the following Batch Upload jobs to run automatically, enable them by setting their ACTIVE_BOOL attribute to 1.
   - EXT_BATCH_UPLOAD_FLIGHT_DATA
   - EXT_BATCH_UPLOAD_CLEAN_TASK_DATA
   - EXT_BATCH_UPLOAD_GENERATE_AVAILABILITY_DATA
   - EXT_BATCH_UPLOAD_GENERATE_CAPABILITY_DATA
   - EXT_BATCH_UPLOAD_GENERATE_TASK_BATCH_DATA

Limited Availability Release

- EXT_BATCH_UPLOAD_GENERATE_REPROCESS_TASK_BATCH_DATA
- EXT_BATCH_UPLOAD_GENERATE_REPROCESS_UNCOMPLETE_TASK_BATCH_DATA

> **Tip** You can also enable the jobs through the **Jobs** page in the Maintenix UI. This page is typically accessible through the **Job** item on the **Administrator** menu .

5. Set a value for the REPEAT_INTERVAL attribute.

   REPEAT_INTERVAL specifies the time (in seconds) between consecutive runs of a job, after the first run.

   If START_TIME is provided, Maintenix applies the interval from the start of the last run. If a job runs past the time that the next run should start, Maintenix starts the next run immediately after the job completes. You should avoid such situations if possible.

   If START_TIME is not provided, the interval is the time between the end of one run and the start of the next run.

   For information about the attributes of Maintenix jobs, see the *Operator Administrator Guide*.
6. Go to the UTL_CONFIG_PARM table.
7. Set any of the following configuration parameters, as required.

BATCH_UPLOAD_HTTP_COMPRESSION_ENABLED

When set to TRUE the payload is sent out as compressed.

Applicable job(s):

- GENERATE_TASK_BATCH_DATA
- GENERATE_REPROCESS_TASK_BATCH_DATA
- REPROCESS_UNCOMPLETE_TASK_BATCH_DATA

EXT_KEY_COLD_START_CATCH_UP_HOURS

When performing a cold start, the system will send X days of hours at a time using EXT_KEY_COLD_START_CATCH_UP_HOURS until it catches up to the current date, in order to shorten the period the system holds a transaction open and avoiding timeouts for transferring large data sets.

Applicable job(s):

- GENERATE_TASK_BATCH_DATA

EXT_KEY_REPROCESS_TASK_HOURS

The number of hours a task needs to be in the line planner application before it can be picked up by the reprocess job and before the check for any updates to the task in Maintenix.

Applicable job(s):

- GENERATE_REPROCESS_TASK_BATCH_DATA

Limited Availability Release

- REPROCESS_UNCOMPLETE_TASK_BATCH_DATA

EXT_KEY_LENGTH_MAX_LISTS

The maximum number of tasks that are sent to the line planner application during a single connection request. The default value is 1000.

Applicable job(s):

- GENERATE_TASK_BATCH_DATA
- GENERATE_REPROCESS_TASK_BATCH_DATA
- REPROCESS_UNCOMPLETE_TASK_BATCH_DATA

EXT_KEY_REPROCESS_TASK_STATUS

The statuses of the tasks in the line planner application which will be processed by the reprocess job.

Applicable job(s):

- GENERATE_REPROCESS_TASK_BATCH_DATA

EXT_KEY_NUM_DAY_ACTION

The number of days from today's date that define a range. Only tasks with deadlines that fall within this range will be sent to the line planner application.

Applicable job(s):

- GENERATE_TASK_BATCH_DATA

EXT_KEY_REPROCESS_UNCOMPLETE_WP_STATUS

The statuses of the incomplete work packages to be picked up by the process.

Applicable job(s):

- REPROCESS_UNCOMPLETE_TASK_BATCH_DATA

EXT_KEY_REPROCESS_UNCOMPLETE_TASK_STATUS

The statuses of tasks against the incomplete work package to be picked up by the process.

Applicable job(s):

- REPROCESS_UNCOMPLETE_TASK_BATCH_DATA

EXT_REST_TASKS_DELETE_TASK_STATUS

The statuses of the tasks which will be included when the CLEAN_TASK_DATA process runs.

Applicable job(s):

- CLEAN_TASK_DATA

EXT_REST_TASKS_RULES_ENGINE

Calls the rules engine, which analyzes data and raises alerts in the line planner application. By default, this parameter is set to True. If you want the rules engine be bypassed, set this parameter to False.

Applicable job(s):

Limited Availability Release

- GENERATE_TASK_BATCH_DATA
- GENERATE_REPROCESS_TASK_BATCH_DATA
- REPROCESS_UNCOMPLETE_TASK_BATCH_DATA

EXT_REST_TASKS_DUEDATE_DAYS

Specifies the value in days for the tasks' range due date.

Applicable job(s):

- CLEAN_TASK_DATA

EXT_WP_DATE_RANGE_DAYS

The work package schedule date must occur within a configurable date range—for example, 180 days— to be sent over to the line planning application.

Applicable job(s):

- GENERATE_TASK_BATCH_DATA

Limited Availability Release

3

Line Planner API

The Line Planner API consists of different integrations used to support line planning processes, manage the synchronization of data, and update the plan in Maintenix.

The integrations consist of the following:

- Inbound messages.
- Outbound batch upload processes.

Inbound messages

The inbound Work Package Management messages ensure that actions performed in the line planner application are represented in the core engine and mapped to existing business logic.

The following messages are mapped to update Maintenix when changes are made in the line planner application.

- Schedule Tasks to a New Work Package
- Schedule Tasks to An Existing Work Package
- Unschedule Tasks
- Override Work Package Name
- Prioritize Tasks
- Commit Work Packages
- Request Parts for Work Packages
- Adjust Work Package Schedules

Schedule Tasks to a New Work Package

Line planners using a line planner application review the flight plan for an upcoming aircraft, identify maintenance slots, and assign tasks to these slots. This planning action is triggered by the

Limited Availability Release

line planner dragging one or more tasks to the maintenance slot. These tasks are in an unplanned or planned state. Planned tasks have already been assigned a maintenance slot.

A line planner application user sees maintenance slots but might not see the corresponding work package. If a work package does not exist, with no corresponding maintenance slot, the work package is created and scheduled in Maintenix.

This work flow consists of the following interactions:

1. Creates a new work package, with a scheduled location and date.
2. Assigns tasks to the work package.
3. Cancels any work packages that are now empty as a result of this interaction.

> **Note** Committed work packages are uncommitted before cancellation.

The following data is required:

- Aircraft (REG_CODE).
- Scheduled location.
- Scheduled start date.
- Scheduled end date.
- Account.
- External reference.
- Name.
- List of:
  - Task (barcode).
  - Scheduled start date.

Schedule Tasks to An Existing Work Package

Line planners using a line planner application will review the flight plan for an upcoming aircraft, identify maintenance slots, and assign tasks to these slots. This planning action is triggered by the line planner dragging one or more tasks to the maintenance slot. These tasks are in a unplanned or planned state. Planned tasks have already been assigned a maintenance slot.

A line planner application user sees maintenance slots but might not see the corresponding work package. If a work package does exist, the line planning application will assign the unplanned tasks to the work package.

This work flow consists of the following interactions:

1. Assigns tasks to the work package.
2. Cancels any work packages that are now empty as a result of this interaction

Limited Availability Release

> **Note** Committed work packages are uncommitted before cancellation.

The following data is required:

- Work package (barcode).
- List of:
  - Task (barcode).
  - Scheduled start date.

Unschedule Tasks

When an end user sets the task, within a maintenance slot, into the unplanned task state in a line planner application, an action is triggered to remove tasks from the work package. This often occurs when a deadline for a task is changed to fall outside of a line maintenance planning time horizon.

If all the tasks in a work package are removed, Maintenix cancels the work package.

This work flow consists of the following interactions:

1. Unassigns tasks from the work package.
2. Cancels any work packages that are now empty as a result of this interaction.

> **Note** Committed work packages will are uncommitted before cancellation.

The following data is required:

- List of:
  - Task (barcode).

Override Work Package Name

Line planners may choose to provide a more appropriate name for a work package, based on the information they have. Work package names are generally named according to a corporate policy, to be quickly assessed from a glance; however, this name may not reflect the complexity of the work package.

This work flow consists of the following interaction:

1. Rename a work package.

The following data is required:

- Work package (barcode).
- Name.

Limited Availability Release

Prioritize Tasks

Line planners set task priorities in the line planner application, and the corresponding tasks' priority in Maintenix is updated.

This work flow consists of the following interaction:

1. Sets task priorities.

The following data is required:

- List of:
  - Task (barcode).
  - Priorities (HIGH, LOW).

Commit Work Packages

Line planners review near-term planned tasks and decide if their plans can be finalized and committed. To give the maintenance team early notice, line planners can manually commit the maintenance slot in their line planner application, and the work package is committed in Maintenix.

This work flow consists of the following interaction:

1. Commits a list of work packages.

The following data is required:

- List of:
  - Work package (barcode).

Request Parts for Work Packages

Line planners can request parts for complex maintenance activities when tasks are nearing the scheduled due date.

This work flow consists of the following interaction:

1. Requests parts for the work package.

The following data is required:

- List of:
  - Work package (barcode).

Adjust Work Package Schedules

Line planners are often confronted with circumstances which affect their plans. For example, if a flight is canceled or delayed, the maintenance slot for a location and time will change to match the aircraft's flight schedule.

This work flow consists of the following interaction:

Limited Availability Release

1. Adjusts the work package schedules.

The following data is required:

- List of:
  - Work package (barcode).
  - Scheduled location.
  - Scheduled start date.
  - Scheduled end date.

Inbound message samples

This section contains descriptions, end points, and samples of the inbound messages.

The following information is provided for each event:

- Description of the event.
- End points that are accessed.
- HTTP operation (PUT or POST).
- Request example (if applicable).
- Response example (if applicable).

Schedule tasks to a new work package

This message is used to schedule tasks to a new work package.

This POST method requires a list of the tasks and work package information. Each task must contain the barcode information. The work package information must contain the identifier (barcode) or the work package details.

Request

Endpoint: `http://<server:port>/mxapiext/short-term-planning/schedule-tasks-to-a-new-work-package`

POST - Schedule tasks to a new work package.

```
{
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "required": true,
      "minLength": 1,
      "maxLength": 240
    },
    "aircraft": {
      "type": "string",
      "required": true,
```

Limited Availability Release

```
      "minLength": 1,
      "maxLength": 80
    },
    "scheduledStartDate": {
      "type": "string",
      "required": true,
      "format" : "date-time",
      "description": "ISO8601 Date time format.",
      "pattern": "yyyy-mm-ddTHH:mm:ss.SSS+-HH:mm",
      "example": "2017-07-30T17:28:54.000+00:00"
    },
    "scheduledEndDate": {
      "type": "string",
      "required": true,
      "format" : "date-time",
      "description": "ISO8601 Date time format.",
      "pattern": "yyyy-mm-ddTHH:mm:ss.SSS+-HH:mm",
      "example": "2017-07-30T17:28:54.000+00:00"
    },
    "scheduledLocation": {
      "type": "string",
      "required": true,
      "minLength": 1,
      "maxLength": 2000
    },
    "externalReference": {
      "type": "string",
      "required": true,
      "minLength": 1,
      "maxLength": 80
    },
    "account": {
      "type": "string",
      "required": true,
      "minLength": 1,
      "maxLength": 80,
    },
    "subclass": {
      "type": "string",
      "required": false,
      "minLength": 1,
      "maxLength": 8
    },
    "workTypes": {
      "type": "array",
      "required": false,
      "items": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 8
      }
    },
      "tasks": {
      "type": "array",
```

```
      "required": true,
      "items": {
        "type": "object",
        "properties": {
          "task": {
            "type": "string",
            "required": true,
            "minLength": 1,
            "maxLength": 80
          },
          "scheduledStartDate": {
              "type": "string",
              "required": false,
              "format" : "date-time",
              "description": "ISO8601 Date time format.",
              "pattern": "yyyy-mm-ddTHH:mm:ss.SSS+-HH:mm",
              "example": "2017-07-30T17:28:54.000+00:00"
          }
        }
      }
    }
  }
}
```

Response

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "status": {
        "type": "string",
        "required": true,
        "enum" : ["SUCCESS", "FAIL"]        ]
      },
      "workPackage": {
        "type": "string",
        "required": false,
        "minLength": 1,
        "maxLength": 80
      },
      "task": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
      },
      "errors": {
        "type": "array",
        "required": false,
        "items": {
          "code": {
```

```
            "type": "string",
            "required": true,
            "minLength": 1,
            "maxLength": 80
          },
          "data": {
            "type": "object",
            "required": false,
            "additonalProperties": {
              "type": "object"
            }
          }
        }
      }
    }
  }
}
```

Schedule tasks to an existing work package

This message is used to schedule tasks, that are not assigned, to an existing work package.

This POST method requires a list of the tasks and work package information. Each task must contain the barcode information. The work package information must contain the identifier (barcode) or the work package details).

Request

Endpoint: `http://<server:port>/mxapiext/short-term-planning/schedule-tasks-to-an-existing-work-package`

POST - Assign task(s) to an existing work package

```
"type": "object",
  "properties": {
    "workPackage": {
      "type": "string",
      "required": true,
      "minLength": 1,
      "maxLength": 80
    },
    "tasks": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "task": {
            "type": "string",
            "required": true,
            "minLength": 1,
            "maxLength": 80
          },
```

```
            "scheduledStartDate": {
                "type": "string",
                "required": false,
                "format" : "date-time",
                "description": "ISO8601 Date time format.",
                "pattern": "yyyy-mm-ddTHH:mm:ss.sss+-HH:mm",
                "example": "2017-07-30T17:28:54.000+00:00"
            }
          }
        }
      }
    }
}
```

Response

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "status": {
        "type": "string",
        "required": true,
        "enum" : ["SUCCESS", "FAIL"]
      },
      "workPackage": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
      },
      "task": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
      },
      "errors": {
        "type": "array",
        "required": false,
        "items": {
          "code": {
            "type": "string",
            "minLength": 1,
            "maxLength": 80
          },
          "data": {
            "type": "object",
            "required": false,
            "additonalProperties": {
              "type": "object"
            }
```

```
          }
        }
      }
    }
  }
```

Unschedule tasks

This message is used to unassign a task.

This POST method requires a list of the tasks. Each task must contain the barcode information.

Request

Endpoint: `http://<server:port>/mxapiext/short-term-planning/unschedule-tasks`

POST - Unschedule a task.

```
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "task": {
        "type": "string",
        "minLength": 1,
        "maxLength": 80,
        "required": true
      }
    }
  }
}
```

Response

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "status": {
        "type": "string",
        "required": true,
        "enum" : ["SUCCESS", "FAIL"]
      },
     "task": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
```

```
      },
      "errors": {
        "type": "array",
        "required": false,
        "items": {
          "code": {
            "type": "string",
            "minLength": 1,
            "maxLength": 80
          },
          "data": {
            "type": "object",
            "required": false,
            "additonalProperties": {
              "type": "object"
            }
          }
        }
      }
    }
  }
}
```

Override work package name

This message is used to rename a work package.

This POST method requires a list of work packages, and the proposed new name.

Request

Endpoint: `http://<server:port>/mxapiext/short-term-planning/override-work-package-name`

POST - Rename a work package in active, canceled, or committed state.

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "workPackage": {
        "type": "string",
        "minLength": 1,
        "maxLength": 80,
        "required": true
      },
      "name": {
        "type": "string",
        "minLength": 1,
        "maxLength": 240,
        "required": true
      }
```

```
    }
  }
}
```

Response

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "status": {
        "type": "string",
        "required": true,
        "enum" : ["SUCCESS", "FAIL"]
      },
      "workPackage": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
      },
      "errors": {
        "type": "array",
        "required": false,
        "items": {
          "code": {
            "type": "string",
            "minLength": 1,
            "maxLength": 80
          },
          "data": {
            "type": "object",
            "required": false,
            "additonalProperties": {
              "type": "object"
            }
          }
        }
      }
    }
  }
}
```

Prioritize tasks

This message is used to set task priorities.

This POST method requires a list of the tasks. Each task must contain a barcode and the priority value.

Request

Endpoint: `http://<server:port>/mxapiext/short-term-planning/prioritize-tasks`

POST - Prioritize a task.

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "task": {
        "type": "string",
        "minLength": 1,
        "maxLength": 80,
     "required": true,
      },
      "priority": {
        "type": "string",
        "minLength": 1,
        "maxLength": 8,
     "required": true,
        "example": [
          "HIGH",
          "LOW"
        ]
      }
    }
  }
}
```

Response

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "status": {
        "type": "string",
        "required": true,
        "enum" : ["SUCCESS", "FAIL"]
      },
      "task": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
      },
      "errors": {
        "type": "array",
        "required": false,
```

```
        "items": {
          "code": {
            "type": "string",
            "minLength": 1,
            "maxLength": 80
          },
          "data": {
            "type": "object",
            "required": false,
            "additonalProperties": {
              "type": "object"
            }
          }
        }
      }
    }
  }
}
```

Commit work packages

This message is used to commit work packages.

This POST method requires a list of the work packages. Each work package must contain a barcode.

Request

Endpoint: `http://<server:port>/mxapiext/short-term-planning/commit-work-packages`

POST - Commit a work package.

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "workPackage": {
        "type": "string",
        "minLength": 1,
        "maxLength": 80,
        "required": true
      }
    }
  }
}
```

Response

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "status": {
        "type": "string",
        "required": true,
        "enum" : ["SUCCESS", "FAIL"]
      },
      "workPackage": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
      },
      "errors": {
        "type": "array",
        "required": false,
        "items": {
          "code": {
            "type": "string",
            "minLength": 1,
            "maxLength": 80
          },
          "data": {
            "type": "object",
            "required": false,
            "additonalProperties": {
              "type": "object"
            }
          }
        }
      }
    }
  }
}
```

Request parts for work packages

This message is used to request parts for work packages.

This POST method requires a list of the work packages. Each work package must contain a barcode.

Request

Endpoint: `http://<server:port>/mxapiext/short-term-planning/request-parts-for-work-packages`

POST - Request parts for a work package.

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "workPackage": {
        "type": "string",
        "minLength": 1,
        "maxLength": 80,
        "required": true
      }
    }
  }
}
```

Response

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "status": {
        "type": "string",
        "required": true,
        "enum" : ["SUCCESS", "FAIL"]
      },
      "workPackage": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
      },
      "errors": {
        "type": "array",
        "required": false,
        "items": {
          "code": {
            "type": "string",
            "minLength": 1,
            "maxLength": 80
          },
          "data": {
            "type": "object",
            "required": false,
            "additonalProperties": {
              "type": "object"
            }
          }
        }
```

```
      }
    }
  }
}
```

Adjust work package schedules

This message is used to adjust work package schedules.

This POST method requires a list of the work packages, along with their location, start date, and end date. Each work package must contain a barcode.

Request

Endpoint: `https://<environment:port>/mxapiext/short-term-planning/adjust-work-package-schedules`

POST - Schedule a work package with a valid location, start date, and end date.

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "workPackage": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
      },
      "scheduledStartDate": {
        "type": "string",
        "required": true,
        "format" : "date-time",
        "description": "ISO8601 Date time format.",
        "pattern": "yyyy-mm-ddTHH:mm:ss.sss+-HH:mm",
        "example": "2017-07-30T17:28:54.000+00:00"
      },
      "scheduledEndDate": {
        "type": "string",
        "required": true,
        "format" : "date-time",
        "description": "ISO8601 Date time format.",
        "pattern": "yyyy-mm-ddTHH:mm:ss.sss+-HH:mm",
        "example": "2017-07-30T17:28:54.000+00:00"
      },
      "scheduledLocation": {
        "type": "string",
        "required": false,
        "minLength": 1,
        "maxLength": 2000
      }
```

```
    }
  }
}
```

Response

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
     "status": {
        "type": "string",
        "required": true,
        "enum" : ["SUCCESS", "FAIL"]
      },
      "workPackage": {
        "type": "string",
        "required": true,
        "minLength": 1,
        "maxLength": 80
      },
      "errors": {
        "type": "array",
        "required": false,
        "items": {
          "code": {
            "type": "string",
            "minLength": 1,
            "maxLength": 80
          },
          "data": {
            "type": "object",
            "required": false,
            "additonalProperties": {
              "type": "object"
            }
          }
        }
      }
    }
  }
}
```

Batch upload processes

The batch upload jobs upload data to line planning applications. Note that these jobs are disabled by default when the batch upload extension is installed.

The batch uploads consist of the following job processes:

- EXT_BATCH_UPLOAD_FLIGHT_DATA
- EXT_BATCH_UPLOAD_GENERATE_AVAILABLITY_DATA
- EXT_BATCH_UPLOAD_GENERATE_CAPABILITY_DATA
- EXT_BATCH_UPLOAD_GENERATE_TASK_BATCH_DATA
- EXT_BATCH_UPLOAD_CLEAN_TASK_DATA
- EXT_BATCH_UPLOAD_GENERATE_REPROCESS_TASK_BATCH_DATA
- EXT_BATCH_UPLOAD_GENERATE_REPROCESS_UNCOMPLETE_TASK_BATCH_DATA

The Clean Task Data and Reprocess jobs—unlike the other processes that query data from the Maintenix database and upload it to the line planner application—are synchronization processes that ensure data between Maintenix database. and the line planner application are synchronized.

Note When you perform a batch upload job, a unique identifier called the correlation ID is generated that corresponds to that particular batch. All requests made under the same batch job must contain the correlation ID in the HTTP header.

EXT_BATCH_UPLOAD_FLIGHT_DATA

This job process uploads flight data to the line planner application. The following job flow is executed in sequence:

1. Flight data is queried from the Maintenix database.
2. Flight data is uploaded to the line planner application.
3. If the job is completed successfully: last_success_dt in the EXT_BATCH_UPLOAD table is updated.

EXT_BATCH_UPLOAD_GENERATE_AVAILABILITY_DATA

This job process uploads inventory availability information to the line planner application. The following job flow is executed in sequence:

1. Inventory availability information is queried from the Maintenix database.
2. Inventory availability information is uploaded to the line planner application.
3. If the job is completed successfully: last_success_dt in the EXT_BATCH_UPLOAD table is updated.

EXT_BATCH_UPLOAD_GENERATE_CAPABILITY_DATA

This job process uploads capable work locations and their configurations to the line planner application. The locations must meet the following criteria:

- The location type must be LINE or TRACK
- The location must have at least one capability (for example, assembly-worktype).

Limited Availability Release

The following job flow is executed in sequence:

1. Capable work locations, that meet the criteria listed above, are queried from the Maintenix database.
2. Work locations are uploaded to the line planner application.

EXT_BATCH_UPLOAD_GENERATE_TASK_BATCH_DATA

This job process uploads new tasks, aircraft, parts, and work packages to the line planner application. If any of the tasks are changed from the last time the process ran, they will be included with the new job run process and uploaded to the line planner application.

This job uses the following criteria to select tasks to upload to the line planner application:

- It must satisfy one of these three conditions:
  - The deadline date of the task must be set before the system date and a set number of days configured in the EXT_KEY_NUM_DAY_ACTION configuration parameter. Note that the default value is 90 days.
  - The driving task deadline date must be set before the system date and a set number of days configured in the EXT_KEY_NUM_DAY_ACTION configuration parameter. Note that the default value is 90 days.
  - It is a CORR task.
- If the task has no set deadlines it is picked up by the job process.
- If the aircraft linked to the task does not have a set operator it is not picked up by the job process.
- It uses the timestamp from the last successful run, stored in the EXT_BATCH_UPLOAD table, to filter out newly created or recently updated tasks that will be uploaded to the line planner application. Note that if the timestamp is greater than one day, then it is considered a cold start. Cold starts do not send any historic tasks to the line planning application.

  This job is a synchronous call that follows this execution sequence:

  1. New and updated tasks, that meet the criteria listed above, are queried from the Maintenix database.
  2. Parts, work packages, and aircraft are also queried from the Maintenix database.
  3. If the rules engine configuration parameter EXT_REST_TASK_RULES_ENGINE is enabled, tasks are sent over to the rules engine for the purpose of raising alerts; otherwise, this step is skipped.
  4. Tasks and their associated aircraft and work packages are sent to the line planner application.
  5. Parts are queried from the Maintenix database, and are then used to filter out from the list of tasks. The filtered list is used to query part requirements on tasks.

Limited Availability Release

6. If the rules engine configuration parameter EXT_REST_TASKS_RULE_ENGINE is enabled, tasks with part requirements are sent over to the rules engine to raise alerts; otherwise, this step is skipped.
7. Tasks with part requirements are sent to the line planner application.
8. Work packages are queried from the Maintenix database.
9. Work package information is sent to the line planner application.
10. If the job is completed successfully, last_success_dt in the EXT_BATCH_UPLOAD table is updated.

Synchronization processes

The Clean Task Data and Reprocess batch upload jobs ensure that data between Maintenix and the line planner application are in synch.

EXT_BATCH_UPLOAD_CLEAN_TASK_DATA

This job requests tasks in the line planner application that have the following status: ACTV, FORECAST, PAUSE, IN WORK, or COMPLETED.

Returned tasks are validated in Maintenix based on the following conditions:

- The task (and the corrective task of the fault) is assigned to a component work package.
- The task is not against an aircraft, due over 180 days.
  - The loose task's driving deadline is over 180 days (excluding the corrective task of the fault)

    Note If the driving task's deadline is over 180 days, the task's deadline must be over 180 days or it must not have a deadline.

    .
- The task is against a component repair work package.
- The task is assigned to a work package scheduled to start over 180 days that has a driving deadline due date (180 days as the default) past the SYSDATE. This can be configured by modifying the configuration parameter EXT_REST_TASKS_RANGE_DUEDATE_DAYS.
  - The same conditions apply to an unscheduled work package.
  - If the task has driving tasks, the deadline or driving deadline of each must be over 180 days
- The task is assigned to an unscheduled work package starting over 180 days that has a driving deadline due date (180 days as the default) past the SYSDATE. This can be configured by modifying the configuration parameter EXT_REST_TASKS_RANGE_DUEDATES_DAYS.
  - The same conditions apply to an unscheduled work package.
  - If the task has driving tasks, the deadline or driving deadline of each must be over 180 days
- The fault has a minor severity.

Limited Availability Release

- The aircraft is not the highest inventory.
- The task is against inventory with the following conditions:
  - An aircraft is archived.
  - An aircraft is scrapped.
- The task is the child of a block.

If any one of these conditions are met, Maintenix deletes the task barcode(s) from the line planner application. The following job flow is executed in sequence:

1. A request is sent to the line planner application to return a list of tasks.
2. The list of tasks is validated in Maintenix. If any of the above criteria is met, they are added to another list.
3. The list of tasks is sent to the line planner application for deletion

EXT_BATCH_UPLOAD_GENERATE_REPROCESS_TASK_BATCH_DATA

This job requests tasks have been in the line planner application, for a default value of 24 hours or longer, with a status of ACTV, FORECAST, PAUSE, or IN WORK. The tasks returned by the line planner application are queried in the Maintenix database and sent back to the line planner application to be reprocessed for any additional changes.

Returned tasks are validated in Maintenix based on the following conditions:

- The task's main inventory is currently installed on an aircraft.
- The aircraft is not archived or scrapped.
- The task is not in an RO work package.
- The task is not a minor fault.
- The task is a top task; meaning, it does not have a parent task, or its parent task is a work package.
- The task is not a migration task marked with OFFWING/OFFPARENT.

If any one of these conditions are met, Maintenix deletes the task barcode(s) from the line planner application. The following job flow is executed in sequence:

The following job flow is executed in sequence:

1. A request is sent to the line planner application to return all tasks that have been in the application for 24 hours (default value) or longer (EXT_KEY_REPROCESS_TASK_HOURS and EXT_KEY_REPROCESS_TASK_STATUS).
2. The line planner application returns a list of tasks, that meet the criteria listed above, that are then queried in the Maintenix database (EXT_KEY_LENGTH_MAX_LISTS).
3. IF EXT_REST_TASK_RULES_ENGINE is enabled, the filtered task list is returned to us.
4. The list of tasks is sent to the line planner application for deletion.
5. If the job is completed successfully: last_success_dt in the EXT_BATCH_UPLOAD table is updated.

Limited Availability Release

EXT_BATCH_UPLOAD_GENERATE_REPROCESS_UNCOMPLETE_TASK_BATCH_DATA

This job requests tasks in the line planner application—that are a part of an ACTV, IN WORK, or COMMIT work package— with a status of CANCEL, COMPLETE, ERROR, or TERMINATE. The tasks returned by the line planner application are queried in the Maintenix database and sent back to the application to be reprocessed for any changes. The following job flow is executed in sequence:

Returned tasks are validated in Maintenix based on the following conditions:

- The task's main inventory is currently installed on an aircraft.
- The aircraft is not archived or scrapped.
- The task is not in an RO work package.
- The task is not a minor fault.
- The task is a top task; meaning, it does not have a parent task, or its parent task is a work package.
- The task is not a migration task marked with OFFWING/OFFPARENT.

If any one of these conditions are met, Maintenix deletes the task barcode(s) from the line planner application. The following job flow is executed in sequence:

1. A request is sent to the line planner application to return all tasks that meet the above criteria (EXT_KEY_REPROCESS_UNCOMPLETE_WP_STATUS and EXT_KEY_REPROCESS_UNCOMPLETE_TASK_STATUS).
2. The line planner application returns a list of tasks which are then queried in the Maintenix database (EXT_KEY_LENGTH_MAX_LISTS).
3. IF EXT_REST_TASK_RULES_ENGINE is enabled, the filtered task list is returned to us.
4. The list of tasks is sent to the line planner application for deletion
5. If the job is completed successfully: last_success_dt in the EXT_BATCH_UPLOAD table is updated

Batch upload samples

This section contains descriptions, end points, and samples of the batch upload job processes.

The following information is provided for each job process:

- Frequency of the process.
- Description of the process.
- End points that are accessed.
- PUT body example (if applicable).
- POST body example (if applicable).

Limited Availability Release

Clean task data

This process cleans task data from the line planner application.

> **Note** When you perform a EXT_BATCH_UPLOAD_CLEAN_TASK_DATA job, a unique identifier called the correlation ID is generated that corresponds to that particular batch. All requests made under the same batch job must contain the correlation ID in the HTTP header.

Frequency: Runs every four hours.

Endpoints

`http(s)://<line_planner_instance>/api/integration/task/tasks/status`

POST - Send a list of statuses for tasks that should be validated for completion

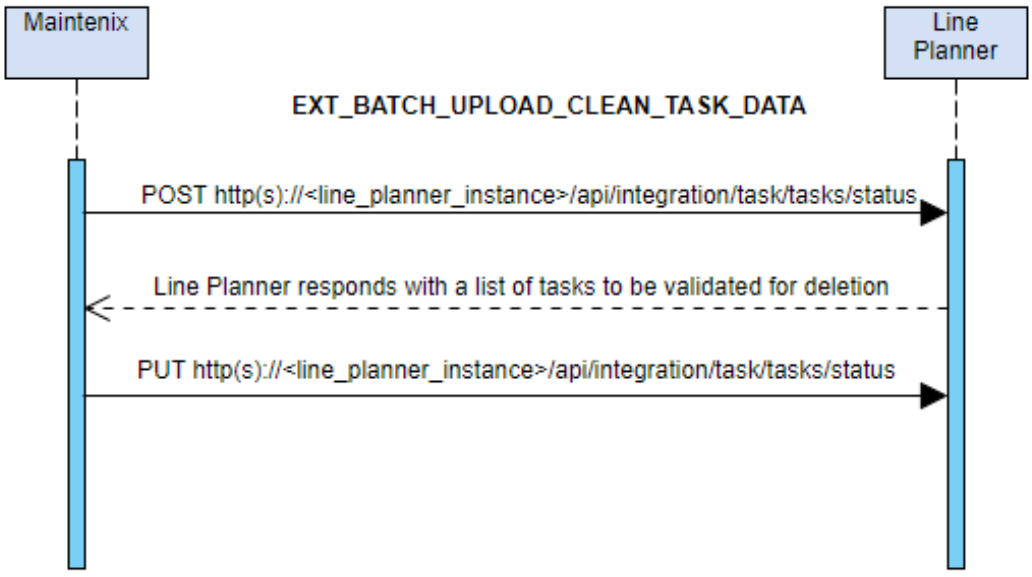
```
["ACTV","FORECAST","PAUSE","IN WORK","COMPLETED"]
```

`http(s)://<line_planner_instance>/api/integration/task/tasks/status`

PUT - Send a list of task barcodes to be deleted from the line planner application.

```
["TTG8Q000PKKE","TTG8Q000PLAP"]
```

Sequence Diagram



Flight data

This process uploads flight data to the line planner application.

> **Note** When you perform a EXT_BATCH_UPLOAD_FLIGHT_DATA job, a unique identifier called the correlation ID is generated that corresponds to that particular batch. All requests made under the same batch job must contain the correlation ID in the HTTP header.

Frequency: Runs every five minutes.

Endpoints

`http(s)://<line_planner_instance>/api/integration/flight/flightv2`

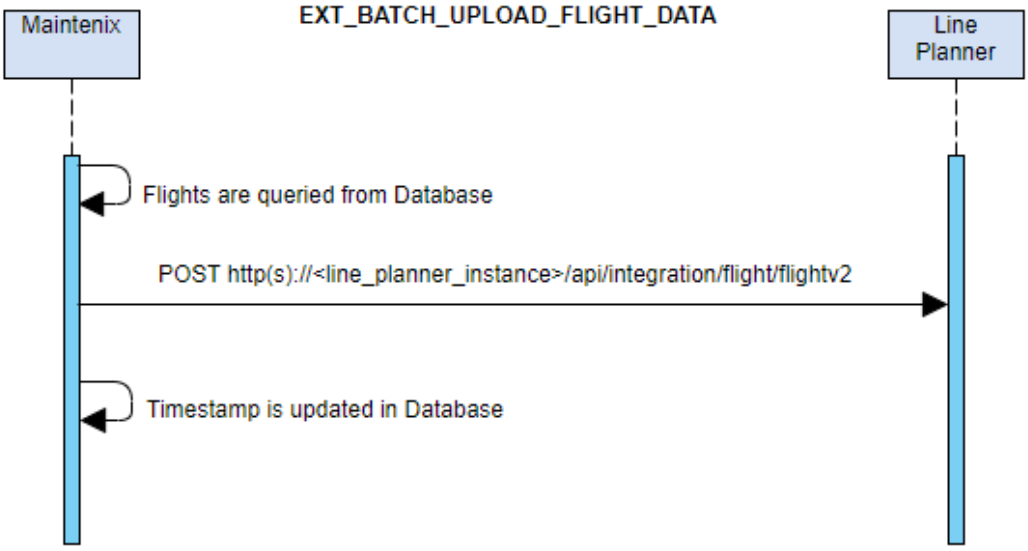
POST - Send an array of the flight data to the line planning system.

```
        "id": {
          "type": "string",
          "description": "Flight Id as supplied by the Maintenix DB",
          "minLength": 1,
          "maxLength": 255
        },
        "flightNumber": {
          "type": "string",
          "description": "The Flight Leg Number from Maintenix",
          "minLength": 1,
          "maxLength": 500
        },
        "aircraft": {
          "type": "string",
          "description": "Aircraft ID as supplied by the Maintenix
 DB",
          "minLength": 1,
          "maxLength": 255
        },
        "aircraftRegistrationCode": {
          "type": "string",
          "description": "Aircraft Registration Code",
          "maxLength": 10
        },
        "departure": {
          "type": "object",
          "description": "Flight Leg Departure Information",
          "required": [
            "scheduledDate",
            "airport"
          ],
           "properties": {
            "scheduledDate": {
              "type": "string",
              "description": "Scheduled Flight Departure Date"
```

```
        },
        "estimatedDate": {
          "type": "string",
          "description": "Estimated Flight Departure Date"
        },
        "actualDate": {
          "type": "string",
          "description": "Actual Flight Departure Date"
        },
        "upDate": {
          "type": "string",
          "description": "Date Flight took off"
        },
        "airport": {
          "type": "string",
          "minLength": 3,
          "maxLength": 3,
          "description": "Departure Airport"
        },
        "gate": {
          "type": "string",
          "minLength": 1,
          "maxLength": 8,
          "description": "Departure Gate"
        }
      }
    },
    "arrival": {
      "type": "object",
      "description": "Flight Leg Arrival Information",
      "required": [
        "scheduledDate",
        "airport"
      ],
      "properties": {
        "scheduledDate": {
          "type": "string",
          "description": "Scheduled Flight Arrival Date"
        },
        "estimatedDate": {
          "type": "string",
          "description": "Estimated Flight Arrival Date"
        },
        "actualDate": {
          "type": "string",
          "description": "Actual Flight Arrival Date"
        },
        "downDate": {
          "type": "string",
          "description": "Date Flight landed"
        },
        "airport": {
          "type": "string",
          "minLength": 3,
          "maxLength": 3,
```

```
            "description": "Arrival Airport"
          },
          "gate": {
            "type": "string",
            "minLength": 1,
            "maxLength": 8,
            "description": "Arrival Gate"
          }
        }
      },
      "reason": {
        "type": "string",
        "maxLength": 8
      },
      "status": {
        "type": "string",
        "maxLength": 16
      },
      "operationalType": {
        "type": "string",
        "maxLength": 8
      },
      "description": {
        "type": "string",
        "maxLength": 4000
      }
    }
  }
}
```


Sequence Diagram



Generate availability data

This process uploads inventory availability information to the line planner application.

> **Note** When you perform a EXT_BATCH_UPLOAD_GENERATE_AVAILABILITY_DATA job, a unique identifier called the correlation ID is generated that corresponds to that particular batch. All requests made under the same batch job must contain the correlation ID in the HTTP header.

Frequency: Runs every five minutes.

Endpoints

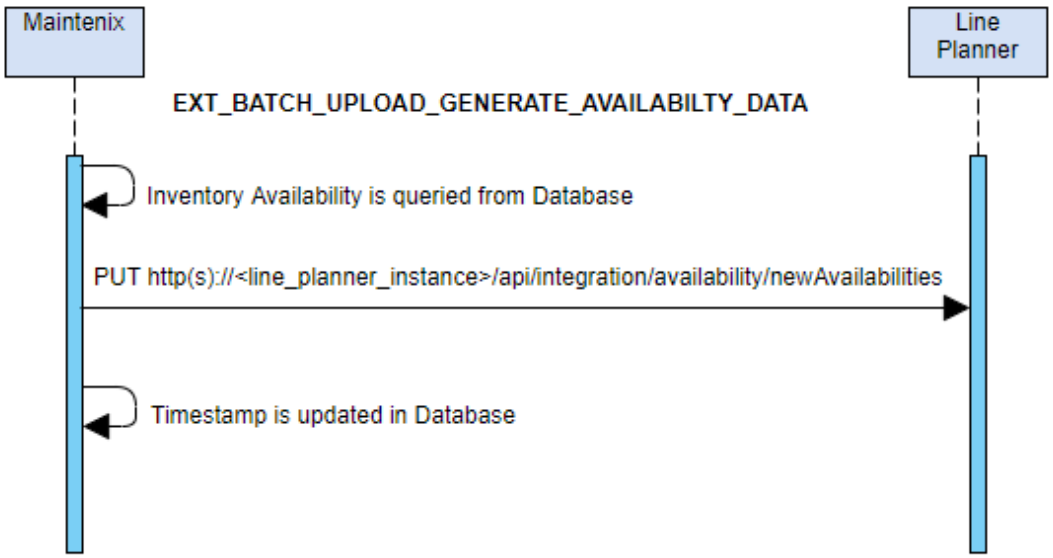
`http(s)://<line_planner_instance>/api/integration/availability/newAvailabilities`

PUT - Send a list of inventories with their availabilities.

```
{
    "inventory": [{
        "partNumber": "12004",
        "flagGroup": "N",
        "manufactureCode": "K3996",
        "systemSource": "MXI",
        "availability": [{
            "inventoryId": null,
            "id": null,
            "supplyLocation": "DAL-MTX",
            "financialOwner": "IFSMX",
            "quantity": 1.0,
            "unit": "EA"
        }]
    }, {
        "partNumber": "12004",
        "flagGroup": "Y",
        "manufactureCode": "K3996",
        "systemSource": "MXI",
        "availability": [{
            "inventoryId": null,
            "id": null,
            "supplyLocation": "DAL-MTX",
            "financialOwner": "IFSMX",
            "quantity": 1.0,
            "unit": "EA"
        }]
    }, {
        "partNumber": "1871671",
        "flagGroup": "Y",
        "manufactureCode": "05257",
        "systemSource": "MXI",
        "availability": [{
```

```
            "inventoryId": null,
            "id": null,
            "supplyLocation": "DAL",
            "financialOwner": "IFSMX",
            "quantity": 1.0,
            "unit": "EA"
        }]
    }]
}
```

Sequence Diagram



Generate capability data

This process uploads capable work locations, and their configurations, to the line planner application.

Note When you perform a EXT_BATCH_UPLOAD_GENERATE_CAPABILITY_DATA job, a unique identifier called the correlation ID is generated that corresponds to that particular batch. All requests made under the same batch job must contain the correlation ID in the HTTP header.

Frequency: Runs every five minutes.

Endpoints

`http(s)://<line_planner_instance>/api/integration/capability/capabilities`

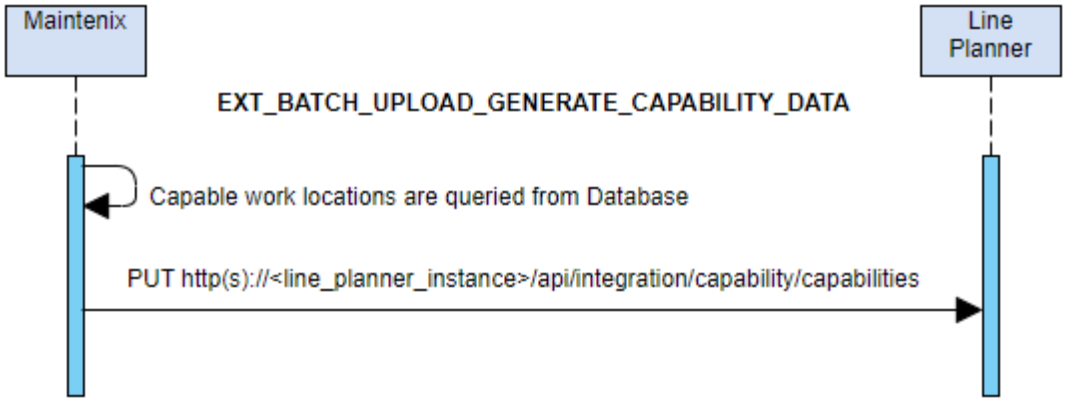
PUT - Send a list of supply locations, with included sub-locations and configurations.


```
[{
    "code": "DAL",
    "description": "Love Field",
    "available": "Y",
    "timezone": "Etc/UTC",
    "subLocationList": [{
        "supplyLocation": "DAL",
        "code": "DAL/LINE",
        "description": "Dallas Line",
        "available": "Y",
        "minGroundTime": 0.0,
        "durationBuffer": 0.0,
        "trackCapacity": 0.0,
        "configurationList": [{
            "code": "DAL/LINE",
            "type": "LINE",
            "workType": "APPEARNC",
            "assembly": "737NG",
            "available": "Y"
        }, {
            "code": "DAL/LINE",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "737NG",
            "available": "Y"
        }]
    }, {
        "supplyLocation": "DAL",
        "code": "DAL/THRU",
        "description": "Thru Maintenance",
        "available": "Y",
        "minGroundTime": 0.0,
        "durationBuffer": 0.0,
        "trackCapacity": 0.0,
        "configurationList": [{
            "code": "DAL/THRU",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "737NG",
            "available": "Y"
        }]
    }]
}, {
    "code": "DEN",
    "description": "Denver International",
    "available": "Y",
    "timezone": "Etc/UTC",
```

```
    "subLocationList": [{
        "supplyLocation": "DEN",
        "code": "DEN/LINE",
        "description": "Denver International LINE",
        "available": "Y",
        "minGroundTime": 0.0,
        "durationBuffer": 0.0,
        "trackCapacity": 0.0,
        "configurationList": [{
            "code": "DEN/LINE",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "737NG",
            "available": "Y"
        }, {
            "code": "DEN/LINE",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "131-9B",
            "available": "Y"
        }, {
            "code": "DEN/LINE",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "AGSE",
            "available": "Y"
        }, {
            "code": "DEN/LINE",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "TSE",
            "available": "Y"
        }, {
            "code": "DEN/LINE",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "CFM56-7B",
            "available": "Y"
        }, {
            "code": "DEN/LINE",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "COMHW",
            "available": "Y"
        }, {
            "code": "DEN/LINE",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "CFM56-7",
            "available": "Y"
        }]
    }, {
        "supplyLocation": "DEN",
        "code": "DEN/THRU",
        "description": "Line Maintenance",
```

```
        "available": "Y",
        "minGroundTime": 0.0,
        "durationBuffer": 0.0,
        "trackCapacity": 0.0,
        "configurationList": [{
            "code": "DEN/THRU",
            "type": "LINE",
            "workType": "LINE",
            "assembly": "737NG",
            "available": "Y"
        }]
    }]
}]
```


Sequence Diagram



Generate reprocess task batch data

These job processes—EXT_BATCH_UPLOAD_GENERATE_REPROCESS_TASK_BATCH_DATA and EXT_BATCH_UPLOAD_GENERATE_REPROCESS_UNCOMPLETE_TASK_BATCH_DATA request tasks from the line planner application to be processed.

Note When you perform a EXT_BATCH_UPLOAD_GENERATE_REPROCESS job, a unique identifier called the correlation ID is generated that corresponds to that particular batch. All requests made under the same batch job must contain the correlation ID in the HTTP header.

Frequency: REPROCESS_TASK_BATCH_DATA runs every four hours; UNCOMPLETE_TASK_BATCH_DATA runs every 15 minutes.

Endpoints

`http(s)://<line_planner_instance>/api/integration/task/tasks/reprocess/completed`

POST - Send a request to the line planner application to return valid tasks to reprocess.

```
{
    "hours": 24,
    "status": ["ACTV", "FORECAST", "PAUSE", "IN WORK"],
    "action": "REPROCESS"
}
```

`http(s)://<line_planner_instance>/api/integration/task/tasks/reprocess/uncompleted`

POST - Send a request to the line planner application to return valid tasks to reprocess.

```
{
    "statusTask": ["CANCEL", "COMPLETE", "ERROR", "TERMINATE"],
    "statusWP": ["IN WORK", "COMMIT", "ACTV"]
}
```

`http(s)://<line_planner_instance>/api/rules/engine/tasks`

POST - Send a list of tasks to the line planner application rule engine for filtering.

```
[{
    "update": null,
    "barcode": "TTG8Q000QKLT",
    "codeMxi": 2795338,
    "name": "ABC-Fault LP2",
    "status": "ACTV",
    "type": "CORR",
    "dueDate": 1533427199000,
    "dueDateExtended": 1533513599000,
    "dueDateLimit": 1533513599000,
    "yield": 0.0,
    "yieldReal": null,
    "yieldRealDouble": "NaN",
    "duration": 0.0,
    "scheduleStatus": null,
    "workPackage": {
        "barcode": "TTG8Q000QLHX",
        "name": "WP-LM-SWA90007AUG2018",
        "status": "ACTV",
        "location": "DAL/LINE",
        "beginDate": 1533668700000,
        "endDate": 1533709800000,
        "userId": null
    },
    "aircraft": {
```

```
        "tail": "SWA900",
        "fleet": "737NG",
        "authority": "FIT",
        "operator": "WN",
        "userId": null
    },
    "ata": "00",
    "originator": null,
    "activation": 1533155203000,
    "taskFault": {
        "barcode": "TTG8Q000QKLT",
        "deferralClass": "MEL A",
        "failType": "F",
        "foundDate": 1533155100000,
        "contact": {
            "firstName": "MXI",
            "lastName": "TECHNOLOGIES",
            "location": "UNMANNED-SUPPLY"
        },
        "instructions": null,
        "description": null,
        "eval": false
    },
    "taskDrive": null,
    "listWorkTypes": [],
    "listParts": [],
    "labourList": [{
        "barcode": "TTG8Q000QKLT",
        "skill": "AMT",
        "manRequired": 1,
        "manHrs": 0.0,
        "userId": "BATCH-TASK-SHORT"
    }, {
        "barcode": "TTG8Q000QKLT",
        "skill": "RII",
        "manRequired": 1,
        "manHrs": 0.0,
        "userId": "BATCH-TASK "
    }],
    "listTools": [],
    "etops": false,
    "instructions": null,
    "description": null,
    "listSchedulings": [{
        "barcode": "TTG8Q000QKLT",
        "driver": 1,
        "usageRemQt": -3.4077314814814814,
        "interval": 3.0,
        "parameter": "CDY",
        "type": "CA",
        "startDt": 1533182399000,
        "startQt": 0.0,
        "dueDt": 1533427199000,
        "dueQt": 0.0,
        "extCalculated": 1.5,
```

```
        "ext": 1.5,
        "currentDt": null,
        "currentQt": 0.0,
        "forecastedDueDate": 1533513599000,
        "userId": "BATCH-TASK-REPROCESS"
    }],
    "actions": [],
    "steps": [],
    "revisionBlock": 0,
    "listAlerts": [],
    "maxEtaPart": null,
    "taskPriorityCd": null,
    "softDeadline": false,
    "userId": "BATCH-TASK-REPROCESS"
}]
```


`http(s)://<line_planner_instance>/api/integration/task/tasks`

PUT - Send a list of tasks to the line planner application.


```
[{
    "update": null,
    "barcode": "TTG8Q000QKLT",
    "codeMxi": 2795338,
    "name": "ABC-Fault LP2",
    "status": "ACTV",
    "type": "CORR",
    "dueDate": 1533427199000,
    "dueDateExtended": 1533513599000,
    "dueDateLimit": 1533513599000,
    "yield": 0.0,
    "duration": 0.0,
    "scheduleStatus": "PLANNED",
    "workPackage": {
        "barcode": "TTG8Q000QLHX",
        "name": "WP-LM-SWA90007AUG2018",
        "status": "ACTV",
        "location": "DAL/LINE",
        "beginDate": 1533668700000,
        "endDate": 1533709800000,
        "userId": null
    },
    "aircraft": {
        "tail": "SWA900",
        "fleet": "737NG",
        "authority": "FIT",
        "operator": "WN",
        "userId": null
    },
    "ata": "00",
    "originator": null,
    "activation": 1533155203000,
    "taskFault": {
        "barcode": "TTG8Q000QKLT",
```

```
        "deferralClass": "MEL A",
        "failType": "F",
        "foundDate": 1533155100000,
        "contact": {
            "firstName": "MXI",
            "lastName": "TECHNOLOGIES",
            "location": "UNMANNED-SUPPLY"
        },
        "instructions": null,
        "description": null
    },
    "taskDrive": {
        "barcode": null,
        "barcodeDrive": null,
        "name": null,
        "dueDate": null
    },
    "listWorkTypes": [],
    "listParts": [],
    "labourList": [{
        "barcode": "TTG8Q000QKLT",
        "skill": "AMT",
        "manRequired": 1,
        "manHrs": 0.0,
        "userId": "BATCH-TASK-REPROCESS"
    }, {
        "barcode": "TTG8Q000QKLT",
        "skill": "RII",
        "manRequired": 1,
        "manHrs": 0.0,
        "userId": "BATCH-TASK-REPROCESS"
    }],
    "listTools": [],
    "etops": false,
    "instructions": null,
    "description": null,
    "listSchedulings": [{
        "barcode": "TTG8Q000QKLT",
        "driver": 1,
        "usageRemQt": -3.4077314814814814,
        "interval": 3.0,
        "parameter": "CDY",
        "type": "CA",
        "startDt": 1533182399000,
        "startQt": 0.0,
        "dueDt": 1533427199000,
        "dueQt": 0.0,
        "extCalculated": 1.5,
        "ext": 1.5,
        "currentDt": null,
        "currentQt": 0.0,
        "forecastedDueDate": 1533513599000,
        "userId": "BATCH-TASK-REPROCESS"
    }],
    "actions": [],
```

```
    "steps": [],
    "revisionBlock": 0,
    "listAlerts": [{
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 2,
            "groupRelation": "ALERT",
            "name": "NO_ESTIMATED_TIME"
        },
        "status": {
            "seq": 3,
            "groupRelation": "STATUS",
            "name": "ACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 6,
            "groupRelation": "ALERT",
            "name": "NO_PART_AVAILABLE"
        },
        "status": {
            "seq": 5,
            "groupRelation": "STATUS",
            "name": "INACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 10,
            "groupRelation": "ALERT",
            "name": "PART_ETA"
        },
        "status": {
            "seq": 5,
            "groupRelation": "STATUS",
            "name": "INACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
```

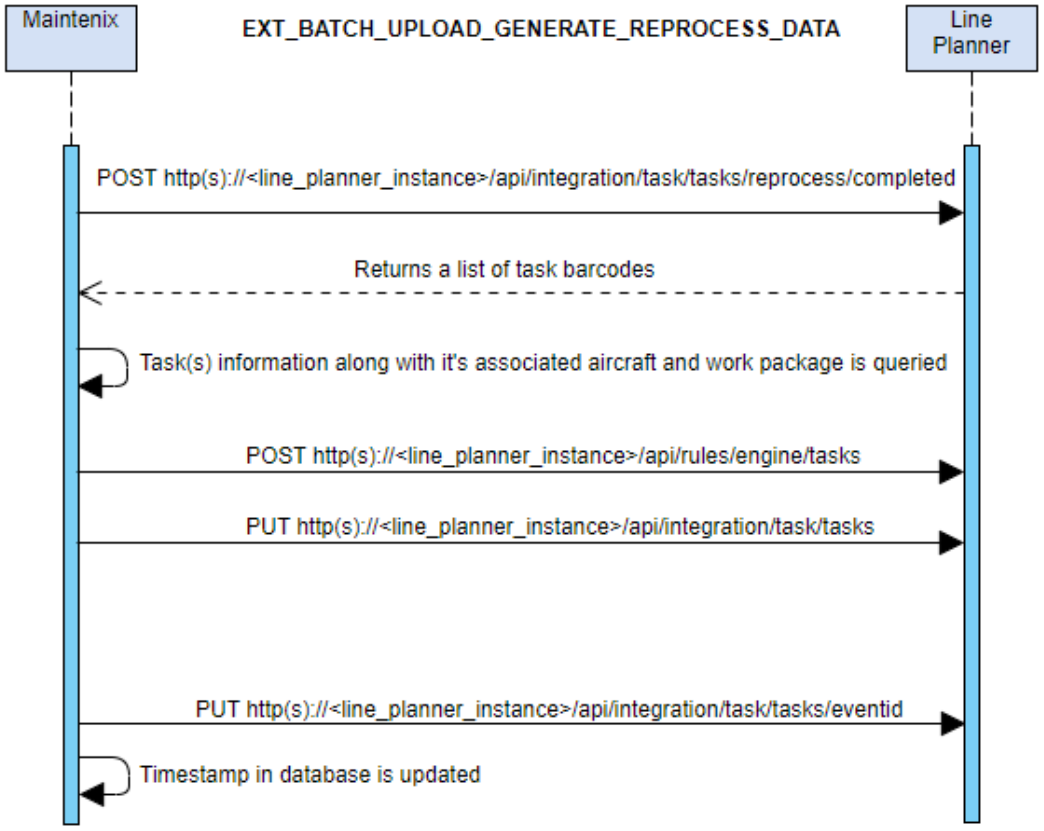
```
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 7,
            "groupRelation": "ALERT",
            "name": "NO_TOOL_AVAILABLE"
        },
        "status": {
            "seq": 5,
            "groupRelation": "STATUS",
            "name": "INACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 1,
            "groupRelation": "ALERT",
            "name": "NO_MAN_HOURS"
        },
        "status": {
            "seq": 3,
            "groupRelation": "STATUS",
            "name": "ACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 16,
            "groupRelation": "ALERT",
            "name": "SLOT_NOT_ENOUGH"
        },
        "status": {
            "seq": 5,
            "groupRelation": "STATUS",
            "name": "INACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }],
    "maxEtaPart": null,
    "userId": "BATCH-TASK-REPROCESS"
}]
```

`http(s)://<line_planner_instance>/api/integration/task/tasks/eventid`

PUT - Send a list of task IDs to delete from the line planner application.

```
["104718","108631","66101","80866" ]
```

Sequence Diagram



Generate task batch data

This job uploads tasks, aircraft, parts, and work packages to the line planner application.

> **Note** When you perform a EXT_BATCH_UPLOAD_GENERATE_TASK_BATCH_DATA job, a unique identifier called the correlation ID is generated that corresponds to that particular batch. All requests made under the same batch job must contain the correlation ID in the HTTP header.

Frequency: Runs every five minutes. Note that this value can be configured.

Endpoints

`http(s)://<line_planner_instance>/api/rules/engine/tasks`

POST - Send a list of tasks to the line planner application rule engine for populating alerts.


```
[{
    "update": null,
    "barcode": "TTG8Q000QKLT",
    "codeMxi": 2795338,
    "name": "ABC-Fault LP2",
    "status": "ACTV",
    "type": "CORR",
    "dueDate": 1533427199000,
    "dueDateExtended": 1533513599000,
    "dueDateLimit": 1533513599000,
    "yield": 0.0,
    "yieldReal": null,
    "yieldRealDouble": "NaN",
    "duration": 0.0,
    "scheduleStatus": null,
    "workPackage": {
        "barcode": "TTG8Q000QLHX",
        "name": "WP-LM-SWA90007AUG2018",
        "status": "ACTV",
        "location": "DAL/LINE",
        "beginDate": 1533668700000,
        "endDate": 1533709800000,
        "userId": null
    },
    "aircraft": {
        "tail": "SWA900",
        "fleet": "737NG",
        "authority": "FIT",
        "operator": "WN",
        "userId": null
    },
    "ata": "00",
    "originator": null,
    "activation": 1533155203000,
    "taskFault": {
        "barcode": "TTG8Q000QKLT",
        "deferralClass": "MEL A",
        "failType": "F",
        "foundDate": 1533155100000,
        "contact": {
            "firstName": "MXI",
            "lastName": "TECHNOLOGIES",
            "location": "UNMANNED-SUPPLY"
        },
        "instructions": null,
        "description": null,
        "eval": false
    },
    "taskDrive": null,
```

```
    "listWorkTypes": [],
    "listParts": [],
    "labourList": [{
        "barcode": "TTG8Q000QKLT",
        "skill": "AMT",
        "manRequired": 1,
        "manHrs": 0.0,
        "userId": "BATCH-TASK-SHORT"
    }, {
        "barcode": "TTG8Q000QKLT",
        "skill": "RII",
        "manRequired": 1,
        "manHrs": 0.0,
        "userId": "BATCH-TASK "
    }],
    "listTools": [],
    "etops": false,
    "instructions": null,
    "description": null,
    "listSchedulings": [{
        "barcode": "TTG8Q000QKLT",
        "driver": 1,
        "usageRemQt": -3.4077314814814814,
        "interval": 3.0,
        "parameter": "CDY",
        "type": "CA",
        "startDt": 1533182399000,
        "startQt": 0.0,
        "dueDt": 1533427199000,
        "dueQt": 0.0,
        "extCalculated": 1.5,
        "ext": 1.5,
        "currentDt": null,
        "currentQt": 0.0,
        "forecastedDueDate": 1533513599000,
        "userId": "BATCH-TASK "
    }],
    "actions": [],
    "steps": [],
    "revisionBlock": 0,
    "listAlerts": [],
    "maxEtaPart": null,
    "taskPriorityCd": null,
    "softDeadline": false,
    "userId": "BATCH-TASK"
}]
```


`http(s)://<line_planner_instance>/api/integration/task/tasks`

PUT - Send a list of tasks to the line planner application.


```
[{
    "update": null,
    "barcode": "TTG8Q000QKLT",
```

```
    "codeMxi": 2795338,
    "name": "ABC-Fault LP2",
    "status": "ACTV",
    "type": "CORR",
    "dueDate": 1533427199000,
    "dueDateExtended": 1533513599000,
    "dueDateLimit": 1533513599000,
    "yield": 0.0,
    "duration": 0.0,
    "scheduleStatus": "PLANNED",
    "workPackage": {
        "barcode": "TTG8Q000QLHX",
        "name": "WP-LM-SWA90007AUG2018",
        "status": "ACTV",
        "location": "DAL/LINE",
        "beginDate": 1533668700000,
        "endDate": 1533709800000,
        "userId": null
    },
    "aircraft": {
        "tail": "SWA900",
        "fleet": "737NG",
        "authority": "FIT",
        "operator": "WN",
        "userId": null
    },
    "ata": "00",
    "originator": null,
    "activation": 1533155203000,
    "taskFault": {
        "barcode": "TTG8Q000QKLT",
        "deferralClass": "MEL A",
        "failType": "F",
        "foundDate": 1533155100000,
        "contact": {
            "firstName": "MXI",
            "lastName": "TECHNOLOGIES",
            "location": "UNMANNED-SUPPLY"
        },
        "instructions": null,
        "description": null
    },
    "taskDrive": {
        "barcode": null,
        "barcodeDrive": null,
        "name": null,
        "dueDate": null
    },
    "listWorkTypes": [],
    "listParts": [],
    "labourList": [{
        "barcode": "TTG8Q000QKLT",
        "skill": "AMT",
        "manRequired": 1,
        "manHrs": 0.0,
```

```
        "userId": "BATCH-TASK"
    }, {
        "barcode": "TTG8Q000QKLT",
        "skill": "RII",
        "manRequired": 1,
        "manHrs": 0.0,
        "userId": "BATCH-TASK"
    }],
    "listTools": [],
    "etops": false,
    "instructions": null,
    "description": null,
    "listSchedulings": [{
        "barcode": "TTG8Q000QKLT",
        "driver": 1,
        "usageRemQt": -3.4077314814814814,
        "interval": 3.0,
        "parameter": "CDY",
        "type": "CA",
        "startDt": 1533182399000,
        "startQt": 0.0,
        "dueDt": 1533427199000,
        "dueQt": 0.0,
        "extCalculated": 1.5,
        "ext": 1.5,
        "currentDt": null,
        "currentQt": 0.0,
        "forecastedDueDate": 1533513599000,
        "userId": "BATCH-TASK "
    }],
    "actions": [],
    "steps": [],
    "revisionBlock": 0,
    "listAlerts": [{
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 2,
            "groupRelation": "ALERT",
            "name": "NO_ESTIMATED_TIME"
        },
        "status": {
            "seq": 3,
            "groupRelation": "STATUS",
            "name": "ACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 6,
```

```
            "groupRelation": "ALERT",
            "name": "NO_PART_AVAILABLE"
        },
        "status": {
            "seq": 5,
            "groupRelation": "STATUS",
            "name": "INACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 10,
            "groupRelation": "ALERT",
            "name": "PART_ETA"
        },
        "status": {
            "seq": 5,
            "groupRelation": "STATUS",
            "name": "INACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 7,
            "groupRelation": "ALERT",
            "name": "NO_TOOL_AVAILABLE"
        },
        "status": {
            "seq": 5,
            "groupRelation": "STATUS",
            "name": "INACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 1,
            "groupRelation": "ALERT",
            "name": "NO_MAN_HOURS"
        },
        "status": {
```

```
            "seq": 3,
            "groupRelation": "STATUS",
            "name": "ACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }, {
        "seq": null,
        "barcode": "TTG8Q000QKLT",
        "type": {
            "seq": 16,
            "groupRelation": "ALERT",
            "name": "SLOT_NOT_ENOUGH"
        },
        "status": {
            "seq": 5,
            "groupRelation": "STATUS",
            "name": "INACTIVE"
        },
        "user": null,
        "start": null,
        "end": null,
        "listAlertDetails": null
    }],
    "maxEtaPart": null,
    "userId": "BATCH-TASK"
}]
```


`http(s)://<line_planner_instance>/api/integration/task/tasks/aircrafts`

PUT - Send a list of aircraft to line planner application.


```
[{
    "tail": "SWA900",
    "fleet": "737NG",
    "authority": "FIT",
    "operator": "WN",
    "userId": null
}]
```


`http(s)://<line_planner_instance>/api/integration/task/tasks/workpackages`

PUT - Send a list of work packages to the line planner application.


```
[{
    "barcode": "TTG8Q000QLHX",
    "name": "WP-LM-SWA90007AUG2018",
    "status": "ACTV",
    "location": "DAL/LINE",
```

```
    "beginDate": 1533668700000,
    "endDate": 1533709800000,
    "userId": null
}]
```

`http(s)://<line_planner_instance>/api/rules/engine/parts`

POST - Send a list of parts to the line planner application rule engine for populating alerts.

```
{
   "update":null,
   "barcode":null,
   "codeMxi":107722,
   "name":null,
   "status":null,
   "type":null,
   "dueDate":null,
   "dueDateExtended":null,
   "dueDateLimit":null,
   "yield":null,
   "yieldReal":null,
   "yieldRealDouble":null,
   "duration":null,
   "scheduleStatus":null,
   "workPackage":null,
   "aircraft":null,
   "ata":null,
   "originator":null,
   "activation":null,
   "taskFault":null,
   "taskDrive":null,
   "listWorkTypes":[

   ],
   "listParts":[
      {
         "id":null,
         "uuid":"64D1FC614FC34017A4097A6C54D1B480",
         "systemOrigin":"MTX",
         "barcode":null,
         "part":{
            "id":null,
            "cdPart":"398774-8",
            "namePart":"MODULATION",
            "manufacturer":"1000",
            "unit":"EA"
         },
         "partGroup":{
            "id":null,
            "cdGroup":"211001845",
            "nameGroup":"MODULATION",
            "userId":"BATCH-PART"
         },
         "reqAction":"REQ",
```

```
            "quantity":1.0,
            "unit":"EA",
            "availability":null,
            "reqMaster":"R0002B1E",
            "eta":null,
            "status":"OPEN",
            "enabled":1
         }
      ],
      "labourList":[

      ],
      "listTools":[

      ],
      "etops":false,
      "instructions":null,
      "description":null,
      "listSchedulings":[

      ],
      "actions":[

      ],
      "steps":[

      ],
      "revisionBlock":null,
      "listAlerts":[

      ],
      "maxEtaPart":null,
      "userId":"BATCH-PART",
      "taskPriorityCd":null,
      "softDeadline":null
}
```


`http(s)://<line_planner_instance>/api/integration/task/tasks/parts`

PUT - Send a list of parts to the line planner application.


```
{
   "update":null,
   "barcode":null,
   "codeMxi":107722,
   "name":null,
   "status":null,
   "type":null,
   "dueDate":null,
   "dueDateExtended":null,
   "dueDateLimit":null,
   "yield":null,
   "yieldReal":null,
   "yieldRealDouble":null,
```

```
"duration":null,
"scheduleStatus":null,
"workPackage":null,
"aircraft":null,
"ata":null,
"originator":null,
"activation":null,
"taskFault":null,
"taskDrive":null,
"listWorkTypes":[

],
"listParts":[
   {
      "id":null,
      "uuid":"64D1FC614FC34017A4097A6C54D1B480",
      "systemOrigin":"MTX",
      "barcode":null,
      "part":{
         "id":null,
         "cdPart":"398774-8",
         "namePart":"MODULATION",
         "manufacturer":"1000",
         "unit":"EA"
      },
      "partGroup":{
         "id":null,
         "cdGroup":"211001845",
         "nameGroup":"MODULATION",
         "userId":"BATCH-PART"
      },
      "reqAction":"REQ",
      "quantity":1.0,
      "unit":"EA",
      "availability":null,
      "reqMaster":"R0002B1E",
      "eta":null,
      "status":"OPEN",
      "enabled":1
   }
],
"labourList":[

],
"listTools":[

],
"etops":false,
"instructions":null,
"description":null,
"listSchedulings":[

],
"actions":[
```

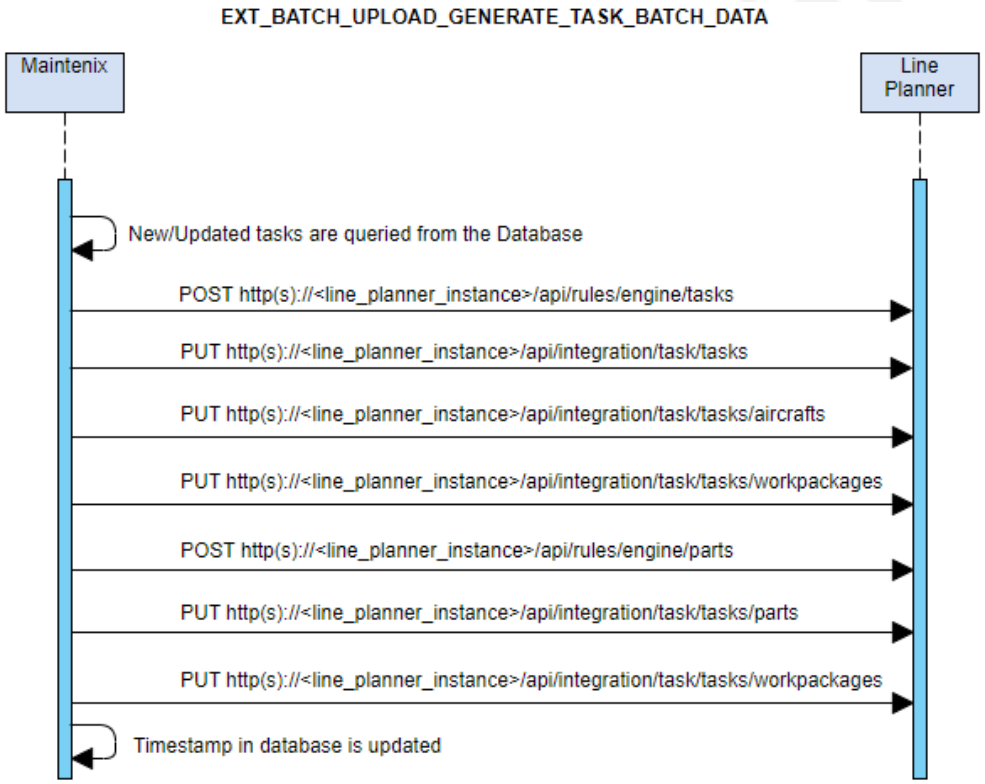
```
    ],
    "steps":[

    ],
    "revisionBlock":null,
    "listAlerts":[

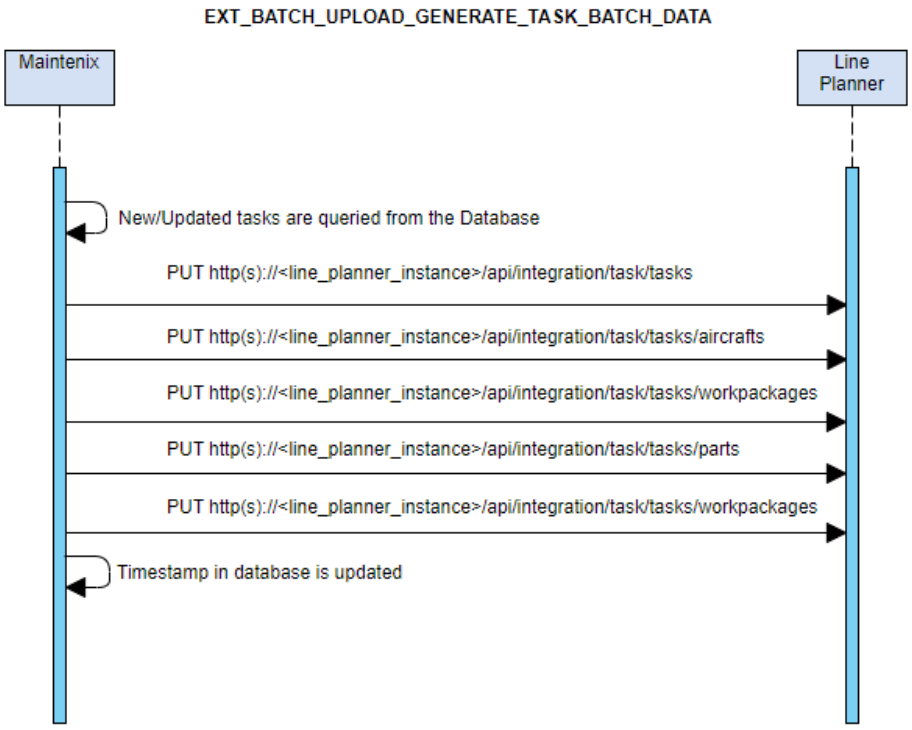
    ],
    "maxEtaPart":null,
    "userId":"BATCH-PART",
    "taskPriorityCd":null,
    "softDeadline":null
}
```


Sequence Diagram

The following diagram displays the process with the rules engine.



The following diagram displays the process without the rules engine.



Limited Availability Release

A

Line Planner Authentication

Batch uploads use JWT, a JSON web token, as a mechanism for securely transferring information between Maintenix and the line planner application.

Inbound authentication

An integration user must be created to support the line planner application. The user credentials should be securely stored within the line planner application and used for each API request to Maintenix via basic authentication

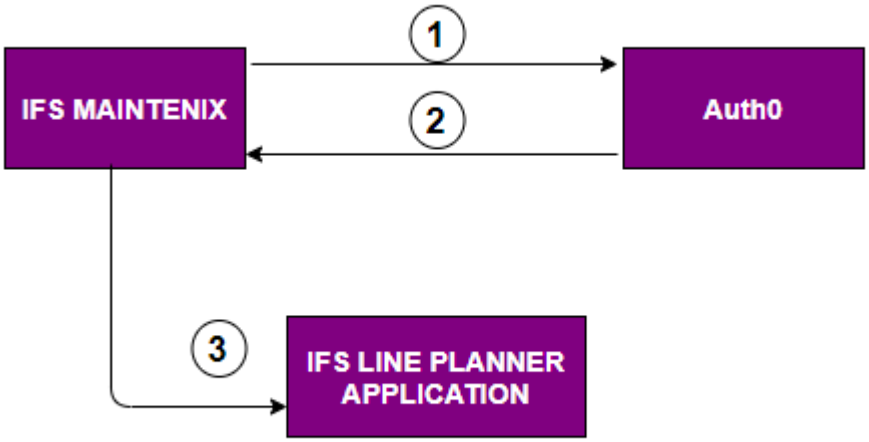
Information on creating the integration user can be found in the Maintenix *Operator Extensions SaaS Configuration Guide*.

Outbound authentication

Based on the credentials provided in the `mxservers.properties` file, at the start of each batch upload process, Maintenix retrieves the JWT token from the authorization server and places it on the JSON authorization header using the bearer schema. Maintenix does not store the token, a new token is requested for each batch upload execution. The token is not validated; validation occurs in the line planner application. This is referred to as a stateless authentication mechanism since the user state is never saved in server memory.

The line planner application always checks for a valid JWT in the authorization header, and it allows the user access if the JWT is present and valid. The following diagram shows how JWT is obtained and used to access the line planner application.

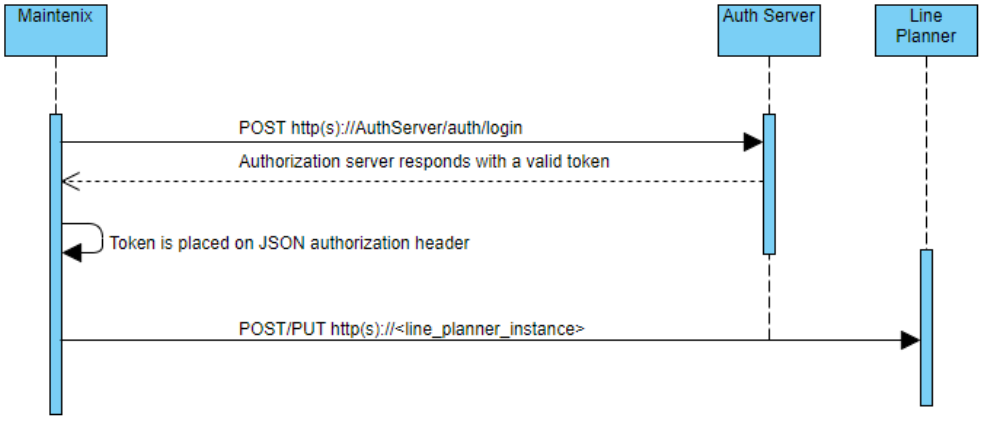
Limited Availability Release



Line Planner Outbound Authentication

1. Maintenix requests authorization from the authorization server.
2. When authorization is granted, the server returns an access token to the application.
3. Maintenix uses the token to access the line planner application.

Sequence Diagram



B

Error Handling

Status codes and messages are returned based on incoming messages and outbound jobs. Descriptions of the messages are provided below.

Note that batch jobs are a synchronous process; the upload will be retried when the job is next scheduled to run. If alerts are configured, an alert is raised when a job fails.

Additional information can be found in the Maintenix logs.

Inbound successful request

In the event of a successful inbound request, the line planner application returns one of the following status codes:

Status Code	Message	Description
207	Multi Status	Batch request was successfully processed.

Batch upload successful request

In the event of a successful batch request, the line planner application returns one of the following status codes:

Status Code	Message	Description
200	OK	Standard successful HTTP request.
201	Created	Request fulfilled, new resource created.
202	Accepted	Accepted for processing.

Error

In the event of a business error, the server logs indicate the following error code and a description:

Limited Availability Release

Status Code	Message	Description
400	Bad Request	The request does not meet the API specification.
401	Unauthorized	The user is not authenticated.
403	Forbidden	The authenticated user does not have the necessary permissions for the resource.
404	Not Found	The resource could not be found.

In the event of a technical error, the server logs indicate the following error code and a description:

Status Code	Message	Description
500	Internal Server Error	The request could not be processed due to an internal server error.

Limited Availability Release

ABOUT IFS

IFS develops and delivers enterprise software for customers around the world who manufacture and distribute goods, maintain assets, and manage service-focused operations. The industry expertise of our people and solutions, together with commitment to our customers, has made us a recognized leader and the most recommended supplier in our sector. Our team of 3,500 employees supports more than 10,000 customers world-wide from a network of local offices and through our growing ecosystem of partners.

#forthechallengers

ifs.com

WHERE WE ARE

AMERICAS

+1 888 437 4968

ASIA PACIFIC

+65 63 33 33 00

EUROPE EAST

+48 22 577 45 00

EUROPE CENTRAL

+49 9131 77 340

UK & IRELAND

+44 1494 428 900

FRANCE, BENELUX AND IBERICA

+33 3 89 50 72 72

MIDDLE EAST AND AFRICA

+971 4390 0888

NORDICS

+46 13 460 4000

Limited Availability Release

COPYRIGHT © 2026 INDUSTRIAL AND FINANCIAL SYSTEMS, IFS AB. IFS AND ALL IFS PRODUCTS AND SERVICES NAMES ARE TRADEMARKS OF IFS. ALL RIGHTS RESERVED. THIS DOCUMENT MAY CONTAIN STATEMENTS OF POSSIBLE FUTURE FUNCTIONALITY FOR IFS'S PRODUCTS AND TECHNOLOGY. SUCH STATEMENTS ARE FOR INFORMATION PURPOSES ONLY AND SHOULD NOT BE INTERPRETED AS ANY COMMITMENT OR REPRESENTATION. THE NAMES OF ACTUAL COMPANIES AND PRODUCTS MENTIONED HEREIN MAY BE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.