IFS Aviation Maintenance

Material Planner API Guide

ICAM 26.2



Document Revision: 26.2-1

Publication Date: November 02, 2026

Contents

1 Introduction....4
About Maintenix APIs....4
Material Planner API Interface....5
How to use the Material Planner API....5
Integration architecture....6

2 Integration scenarios....7
Create shipment scenarios....7
Validate request against schema....7
Business validation for shipment header in incoming message....8
Business validation for shipment line in incoming message....8
Create a shipment....9
Create purchase order scenarios....9
Validate request against schema....9
Business validation for purchase order header in incoming message....10
Processing and validation for SPEC2K vendor fields....11
Business validation for purchase order lines in incoming message....11
Create a purchase order....12
Set stock level scenarios....12
Validate request for batch size....13
Validate request against schema....13
Business validation for stock level in incoming message....13
Process the stock level....14

3 Setting up the Material Planner API....16

4 Managing the integration....18
Security....18
Logging....18

5 Message response error codes....19

6 Set stock level....21
Default values....21

Batch processing....22
Batch message response error codes....22

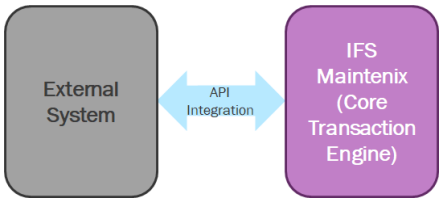
7 Sample messages and schemas....24
Create shipment sample message....24
Create shipment sample success response....24
Create shipment sample error response....25
Create purchase order sample message....25
Create purchase order sample success response....26
Create purchase order sample error response....26
Set stock level sample message....26
Set stock level sample success response....27
Set stock level sample error response....27

1 Introduction

About Maintenix APIs

Maintenix Application Programming Interfaces (APIs) support logical integrations with external business applications. You can also use Maintenix APIs for partial implementations, where your organization uses Maintenix in conjunction with an existing Enterprise Resource Planning (ERP) application, or for phased implementations, where workflows are supported by the existing ERP is transferred to Maintenix over time.

APIs simplify the task of integration, but do not deal with error handling between the integration layer of Maintenix and the external system. Messages are designed to describe the data in a flexible way, but message invocation is left open to customization. Timing considerations are particularly important when dealing with Web services that support a request-based integration as opposed to trigger-based integration.



Integration with Maintenix using APIs

There are three types of messaging scenarios available with the Maintenix APIs:

1. *Operational Service Messaging Scenario*: In this scenario, the external system sends an inbound message to Maintenix containing the information that Maintenix needs to function correctly. Upon successful processing of a message, Maintenix sends an outbound response message for confirmation.

2. *Querying Service Messaging Scenario:* In this scenario, the external system sends an inbound message to query information from Maintenix. Maintenix returns an outbound message containing the requested information.
3. *Notification Service Messaging Scenario*: In this scenario, certain Maintenix work flows create predefined events. These events are asynchonously pushed to an external system.

Note Not all APIs support all messaging scenarios.

Material Planner API Interface

The Material Planner APIs are a set of APIs for the Maintenix system to interact with external material planning systems.

While Maintenix has the ability to maintain minimum stock levels automatically through the use of the stock low action, it does not make long range predictions for part demand. External material planner systems can forecast the need for parts and use the Material Planner APIs to get the parts to the location where they are needed.

The Material Planner APIs are intended to be used for daily updates to Maintenix, at set intervals throughout the day. The Material Planner APIs are not a real time API, or used for initial data loading. Other solutions should be considered for those scenarios.

Some of the Material Planner APIs support batch processing, where requests can be batched and sent together in one API call.

How to use the Material Planner API

The following high-level scenarios describe the supported situations where you can use the Material Planner API to integrate an external system with Maintenix:

Create Shipment

Endpoint Address: `http://<server-url>/mxapiext/supplychain/materialplanning/create-shipment`. The Create Shipment Material Planner API can be used to create a shipment in Maintenix.

Create Purchase Order

Endpoint Address: `http://<server-url>/mxapiext/supplychain/materialplanning/create-purchase-order`. The Create Purchase Order Material Planner API can be used to create a purchase order in Maintenix.

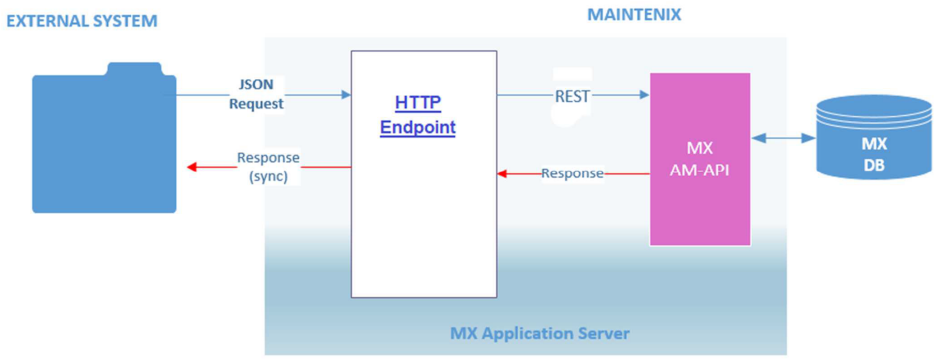
Set Stock Levels

Endpoint Address: `http://<server-url>/mxapiext/supplychain/materialplanning/set-stock-level`. The Set Stock Levels Material Planner API can be used to create, update, and delete stock levels in Maintenix.

Integration architecture

The APIs are accessible via an HTTP-based end. In Maintenix the standard is that all the built APIs will support JSON. The HTTP Endpoint is contained within the WebLogic deployment.

The following diagram shows the high level architecture:



High-level architecture

Components

The external system maps and transforms the incoming messages into data that complies with the Maintenix schema.

The HTTP Endpoint: This layer exposes a public HTTP Endpoint, and it receives incoming requests with a JSON message. This message is first validated and then based on action operation, invokes Maintenix AM-API.

Maintenix Core AM-API is the layer responsible for processing actions in Maintenix.

Interactions

All the communication between the external system andMaintenix is synchronous. It is mandatory to have synchronous responses, to avoid usage of other integrated systems (such as PubSub). The messages used in this integration are JSON, not XML.

2

Integration scenarios

Create shipment scenarios

One Create Shipment message is sent to the Material Planner Interface– this message contains information about the shipment itself, including the origin and destination locations as well as the shipment date. In addition, the message must contain at least one shipment line to be added to the shipment.

Validate request against schema

Goal

To validate an incoming message from an external system against the schema.

Summary

The incoming message should be compliant with schema provided.

Preconditions

- The message has passed authentication.

Messaging

The external system sends an inbound message.

Result

If the schema validation passes, proceed with next step: business validation. If there is an error, then an error message will be returned with a 400 HTTP code instead.

Business validation for shipment header in incoming message

Goal

To validate an incoming Create Shipment message from an external system against the defined business rules.

Summary

The incoming message should respect the business validation rules:

- Validate that the origin and the destination locations of the shipment exist in Maintenix.
- Validate the destination for the shipment is of type DOCK.
- Validate that the two locations are different.
- Validate that the two locations do not share the same supply location.
- Validate the priority, if provided, can be found in Maintenix.
- Validate the date given is valid.

Preconditions

- The message has passed authentication.
- The message has passed the schema validation.

Messaging

The external system sends an inbound message.

Result

If the business validation passes, then the validation will proceed to the shipment lines. If there is an error, then an error message will be returned with a 400 HTTP code.

Business validation for shipment line in incoming message

Goal

To validate the shipment line of an incoming Create Shipment message from an external system against the defined business rules.

Summary

The incoming shipment line should respect the business validation rules:

- Validate that the part can be found.
- Validate that the quantity provided is the correct form for the unit of measure.
- Validate that the provided unit of measure is the standard for the part.
- Validate that a shipment line with the same material planning external reference does not currently exist in Maintenix (if it has been provided in the message).

Preconditions

- The shipment header has passed validation.

Messaging

The external system sends an inbound message.

Result

If the business validation passes, then a message will be sent to the core Maintenix API requesting the creation of a shipment. If there is an error, then an error message will be returned with a 400 HTTP code. In almost all cases, if there is a duplicate material planning external reference, then a 409 HTTP code will be returned instead.

Create a shipment

Goal

To create a shipment using the information provided in the message.

Summary

Create a new shipment header, then add a shipment line to the new shipment using information provided in the message.

Preconditions

- The message passed authentication and is validated successfully.

Messaging

The external system sends an inbound message. Maintenix returns a response message.

Result

The shipment is successfully created with an attached shipment line. If there are no errors, then an HTTP success code (201) is returned along with the barcode of the newly-created shipment. If there is an error, then an HTTP error code will be returned with a message detailing the error encountered.

Create purchase order scenarios

One Create Purchase Order message will be sent to the Material Planner Interface– this message will contain information about the purchase order, such as the destination of the order and vendor information, as well as a single purchase order part line. If the vendor is SPEC2K compliant, then additional information will be included in the message for processing.

Validate request against schema

Goal

To validate an incoming message from an external system against the schema.

Summary

The incoming message should be compliant with schema provided.

Preconditions

- The message has passed authentication.

Messaging

The external system sends an inbound message.

Result

If the schema validation passes, proceed with next step: business validation. If there is an error, then an error message will be returned with a 400 HTTP code instead.

Business validation for purchase order header in incoming message

Goal

To validate an incoming Create Purchase Order message from an external system against the business rules.

Summary

The incoming message should respect the business validation rules:

- Validate that the purchase contact is valid, if given.
- Validate that the receipt organization is valid, if given.
- Validate that the vendor exists and is approved for the receipt organization, if given.
- Validate that the priority is valid, if given.

The following default values will be used for fields that are left blank:

- the purchase contact will be defaulted to the API user
- the receipt organization will be defaulted to the company organization of the HR organization

  that the API user is assigned to
- the vendor will be defaulted to the preferred vendor for a part, in the case where a single purchase order line is provided.
  - If there are multiple purchase order lines, or a preferred vendor can not be found, then an error will be returned.
- the priority will be defaulted to NORMAL

Preconditions

- The message has passed authentication.
- The message has passed the schema validation.

Messaging

The external system sends an inbound message.

Result

If the business validation passes, then the purchase orders lines in the message will be validated. If there is an error, then an error message will be returned with a 400 HTTP code.

Processing and validation for SPEC2K vendor fields

Goal

To validate fields in the message that are specific to SPEC2K-enabled vendors.

Summary

If the vendor given in the message is SPEC2K-enabled, and the following fields are not blank, then additional validation is performed:

- The ship to code (shipToCode, not shipTo) is assigned to the purchase order.
- The spec2000 customer code, if given and valid, will be assigned to the purchase order.
  - If this field is not provided, then the purchase order is assigned the default spec 2000 customer code.

Preconditions

- General validation for the Create Purchase Order header was performed.

Messaging

The external system sends an inbound message.

Result

If the business validation passes and no errors occur, then the validation will continue to the purchase order line. If there is an error, then an error message is returned with a 400 HTTP code.

Business validation for purchase order lines in incoming message

Goal

To validate the purchase order part line of an incoming Create Purchase Order message from an external system against the business rules defined.

Summary

The incoming shipment line should respect the business validation rules:

- Validate that the part can be found.
- Validate that the quantity provided is in the correct form for the unit of measure.
- Validate that the provided unit of measure is the standard for the part.
- Validate that a shipment line with the same material planning external reference does not currently exist in Maintenix (if it has been provided in the message).

Preconditions

- The shipment header passed validation.

Messaging

The external system sends an inbound message.

Result

If the business validation passes, then a message will be sent to the core Maintenix API requesting the creation of a shipment. If there is an error, then an error message will be returned with a 400 HTTP code, in almost all cases – if there is a duplicate material planning external reference – a 409 HTTP code will be returned instead.

Create a purchase order

Goal

To create a purchase order using the information provided in the message.

Summary

Create a new purchase order header, then add a part line to the new purchase order using information provided in the message.

Preconditions

- The message passed authentication and is validated successfully.

Messaging

The external system sends an inbound message. Maintenix returns a response message.

Result

The purchase order is successfully created with an attached part line. If there are no errors, then an HTTP success code (201) is returned with the barcode of the newly-created purchase order. If there is an error, then an HTTP error code is returned with a message detailing the error encountered.

Set stock level scenarios

The Set stock level message contains an array of set stock level commands, each of which refers to a stock level in IFS Maintenix. Each command contains information to either create, update, or delete a stock level.

The stock level command are processed in individual transactions, so the result of one command doesn't affect another. For more information, see *Batch processing*.

Validate request for batch size

Goal

To validate the incoming message does not exceed the maximum batch size for the number of set stock level commands in the message payload.

Summary

The incoming message should be compliant with the batch size.

Preconditions

- The message has passed authentication

Messaging

The external system sends an inbound message.

Result

If the batch validation passes, proceed with next step: schema validation. If there is an error, then an error message will be returned with a 400 HTTP code instead.

Validate request against schema

Goal

To validate an incoming message from an external system against the schema.

Summary

The incoming message should be compliant with schema provided.

Preconditions

- The message has passed authentication.
- The message has passed batch size validation.

Messaging

The external system sends an inbound message.

Result

If the schema validation passes, proceed with next step: business validation. If there is an error, then an error message will be returned with a 400 HTTP code instead.

Business validation for stock level in incoming message

At this point in the processing, each stock level command in the message is processed separately.

Goal

To validate a set stock level command from an external system against the defined business rules.

Summary

The set stock level command should respect the business validation rules:

- Validate the stock can be found in Maintenix.
- Validate the location is a warehouse or supply location in Maintenix.
- If given, validate the owner can be found in Maintenix.
- If given, validate the safety level must be less than or equal to the restock level.
- If given, validate the maximum level must be greater than or equal to the restock level.
- If given, validate the stock low action must be one of the following:
  - [MANUAL, POREQ, SHIPREQ] for a supply stock level
  - [MANUAL, DISTREQ] for a warehouse stock level

Preconditions

- The message has passed authentication.
- The message has passed batch size validation.
- The message has passed schema validation.

Messaging

The external system sends an inbound message.

Result

If the business validation passes, then a message will be sent to the core Maintenix API requesting the stock level command be processed (either created, updated, or deleted). If there is an error, then this set stock level command will fail with a 400 HTTP code and the next command will be processed.

Process the stock level

Goal

To create, update, or delete a stock level in Maintenix using the information provided in the set stock level command.

Summary

The action taken depends on whether the stock level can be found in Maintenix and given restock levels. Note that a stock level is identified by the combination of its stock, location, and owner.

- UPDATE: If a match is found for the stock level in Maintenix and restockLevel>0, the stock level will be updated.
- DELETE: If a match is found for the stock level in Maintenix and restockLevel=0, the stock level will be deleted.
- CREATE: If no match is found for the stock level in Maintenix and restockLevel>0, the stock level will be created.

- No action: If no match is found for the stock level in Maintenix and restockLevel=0, no action will be taken (already in desired state).

Preconditions

- The message has passed authentication.
- The message has passed batch size validation.
- The message has passed schema validation.
- The message has passed business validation.

Messaging

The external system sends an inbound message.

Result

The stock level command is processed. Once all the set stock level commands have been completed, then a response will be returned with an HTTP status code of 207. The response body will indicate if each set stock level command was processed successfully or not.

3

Setting up the Material Planner API

The Material Planner API is installed with Maintenix. See the Maintenix Installation and Configuration Guide.

License requirement

The following license is required to use the Material Planner Integration:

APIs – MATERIAL_PLANNER_API

Configuration parameters

The following configuration parameter is required for the Set stock level API:

SET_STOCK_LEVEL_API_MAX_BATCH_SIZE

Value: Maximum batch size

For more information, see *Batch processing*.

API permission parameters

The following pre-configured user and role are provided for use with the Material Planner Integration. You can create a different user, but, if you do, you must assign this user to the role to send messages into the integration.

Material Planner API user and role

User	Role code	Role name
mpintegration	MPINTEG	Material Planner Integration

By default, the following permission sets are enabled for the MPINTEG role:

- Send Spec 2000 Message

The following three permissions are assigned directly to the MPINTEG role:

- API_CREATE_PURCHASE_ORDER_PARM
- API_CREATE_SHIPMENT_PARM
- API_SET_STOCK_LEVEL_PARM

4

Managing the integration

Security

The following aspects are taken into consideration for security:

- All incoming messages will use Basic Authentication, with credentials (username and password) present in the HTTP header. The password is encrypted.
- Every connection is encrypted with SSL. The standard is to use TLS 1.2.

For more information on security, see the *Maintenix Installation and Configuration Guide*.

Logging

Integration events, exceptions, and responses are logged in the Maintenix log files. The log files use standard formats that you can configure to work with your own log processing and monitoring tools.

The primary log file is `maintenix.log`. This file uses Log4j format.

HTTP requests are also logged in `weblogic-access.log` which you have the option to monitor in real time.

For information on configuring logging, see the *Maintenix Installation and Configuration Guide*.

5

Message response error codes

The following response codes will be returned when sending Create shipment or Create purchase order messages. Note that these are "non-batch" APIs, unlike the "batch" Set stock level API.

For more information on batch API error codes, see the related concept below.

Message Response error codes for non-batch APIs

HTTP response code	Scenario(s)	Error code
201	Successful creation of the shipment or purchase order.	N/A (a barcode will be returned instead)
400	The message fails schema validation.	SCHEMA_VALIDATION_ERROR
400	The message has failed business validation (one or more fields contain incorrect or non-present data).	MAINTENIX_VALIDATION_ERROR
401	The request has failed an authentication check (the user does not exist within Maintenix).	N/A
403	The user is valid, but does not have the authorization to access on or more of the APIs.	N/A
409	The request has a part line with a material planning external reference that is already present within Maintenix.	DUPLICATION_ERROR

HTTP response code	Scenario(s)	Error code
>= 500	There has been an internal server error.	INTERNAL_SERVER_ERROR

Related concepts

Batch message response error codes

6

Set stock level

Default values

The following default values for Set stock level will be used if fields are left blank.

Owner ID

If the owner ID is not given and the stock level is for a Supply location, then the owner will be defaulted to the default local owner.

Safety Level

If the safety level is not given, then it will be defaulted to 0 in a create/update scenario.

Max Level

If the maximum level is not given, then the maximum level will be defaulted to the restock level in a create/update scenario.

Stock Low Action

- If the stock low action is not given for a Supply stock level, then it will be defaulted to MANUAL in a create/update scenario.
- If the stock low action is not given for a Warehouse stock level, then it will be defaulted to DISTREQ in a create/update scenario. Note that when the stock low action is DISTREQ, the Warehouse location must have a specified supplier location or an error will be returned.

Quantity

If the quantity is not given, it will be defaulted to 1 in a create/update scenario.

Batch processing

Requests are sent with a JSON payload and it contains an array of set stock level commands to process. Each set stock level command is processed in its own transaction.

The response payload contains an array of responses, one for each of the set stock level commands. The order of responses correlates to the order of the commands. As each set stock level command is processed in its own transaction, it is possible that there will be both successes and failures in one API call.

Since there is no known industry standard schema for these kinds of messages, it is likely that a middleware piece will transform and pass on messages to Maintenix using the Material Planning Interface (MPI).

Maximum batch size

The Set stock level API supports batch processing, so you can use a configuration parameter to restrict the maximum batch size of set stock level commands allowed in one API call.

The parameter value is configurable so you can fine-tune it for your environment.

Configuration parameter: SET_STOCK_LEVEL_API_MAX_BATCH_SIZE

Value: Maximum batch size

Batch message response error codes

The Set stock level API is a batch API, so the message response codes are different from the "non-batch" APIs (for example, Create shipment and Create purchase order).

Request processing logic

A 207 response means the message was received and processed by Maintenix. The response body has an array of inner responses to indicate whether each set stock level command was successful or not. The array of responses is correlated to the array of set stock level commands by order (first response correlates to the first command, and so on).

Each inner response entity has its own status code to indicate if the processing of that set stock level command was successful or not.

In the case of an error which applies to the entire HTTP request, for example a schema validation failure (400), then the response body will have the same format (array of responses), but the array only contains one response item since it applies to the entire request.

Response-level HTTP status codes

HTTP response code	Scenario(s)	Response body
207	The message was received and processed.	Array with responses (one for each stock level command)
400	The message fails schema validation (for example, one mandatory field is missing), or the number of set stock level commands in one batch was greater than the allowed maximum batch size.	Array with one response (400)
401	The request failed authentication checks.	N/A
403	The authenticated user does not have permission to access the API.	N/A
500	Internal server error.	Array with one response (500)

If a 207 HTTP response code is returned, then the response body will contain an array of responses. Each inner response will have one of the following statuses: 200, 400 or 500.

Inner HTTP status codes in the response body

Inner HTTP status codes for 207 response	Scenario(s)	Response fields
200	The stock level was successfully created/updated/deleted.	N/A
400	There was a business error while processing the set stock level command.	Array of errors
500	Internal server error.	Array of errors

7

Sample messages and schemas

This section includes instructions regarding sample messages for use with both Maintenix and the external system.

You can find schema files in the following Maintenix folder:

```
\\. . .\Documentation\Extensions\Integrators and Developers\API Schemas
```

Create shipment sample message

```
{
   "shipFrom": "YYZ/DOCK",
   "shipTo": "YOW/DOCK",
   "shipByDate": "2018-12-31T00:00:00Z",
   "priority": "AOG",
   "note": "Here is a note",
   "shipmentLines": [
      {
         "partNumber": "1806PN",
         "manufacturer": "BOEING",
         "quantity": "10",
         "unitOfMeasure": "EA",
         "materialPlanningExternalReference": "EXT-001"
      }
   ]
}
```

Create shipment sample success response

The following is a sample outbound message returned from a Create Shipment message:

```
{
```

```
   "status": 201,
   "body": {
      "shipmentId": "S48U00025JD"
   }
}
```

Create shipment sample error response

The following are sample outbound messages returned from a Create Shipment message:

```
{
   "status": 400,
   "body": {
      "errors": [
         {
            "code": "MAINTENIX_VALIDATION_ERROR",
            "message": "The part '1806PN' with manufacturer 'BOEING'
 could not
            be found in Maintenix."
         }
      ]
   }
}
```

Create purchase order sample message

```
{
   "shipTo": "YOW/DOCK",
   "purchaseContact": "jsmith",
   "receiptOrganization": "MXI",
   "vendor": "BOE",
   "priority": "AOG",
   "shipToCode": "12345",
   "spec2000CustomerCode": "CTIRE",
   "noteToVendor": "This is a note to the vendor.",
   "noteToReceiver": "This is a note to the receiver",
   "orderExternalReference": "EXT-001"
   "authorized": false,
   "issued": false,
   "purchaseLines": [
      {
         "partNumber": "1806PN",
         "manufacturer": "BOEING",
         "quantity": "10",
         "unitOfMeasure": "EA",
         "materialPlanningExternalReference": "EXT-002"
      }
   ]
```

```
}
```

Create purchase order sample success response

The following are sample outbound messages returned from a Create Purchase Order message:

```
{
   "status": 201,
   "body": {
      "orderNumber": "S48U00025JD"
   }
}
```

Create purchase order sample error response

The following are sample outbound messages returned from a Create Purchase Order message:

```
{
   "status": 400,
   "body": {
      "errors": [
         {
            "code": "MAINTENIX_VALIDATION_ERROR",
            "message": "The part '1806PN' with manufacturer 'BOEING'
 could not
            be found in Maintenix."
         }
      ]
   }
}
```

Set stock level sample message

The first set stock level command shows the minimum required fields, and the second shows all fields (note that the "quantity" field is only applicable to supply stock levels).

```
[
  {
    "stockId": "3DBE0A529DCA11E99FB100155D928C0E",
    "locationId": "DC9D1F589DCA11E99FB100155D928C0E",
    "restockLevel": 2
  },
  {
    "stockId": "603ED7F018D611E993C2FBDA0000122A",
    "locationId": "8C12D8F022F811E9973295AE0000122A",
```

```
    "ownerId": "93DC1A6122F811E9973295AE0000122A",
    "safetyLevel": 3,
    "restockLevel": 4,
    "maxLevel": 5,
    "stockLowAction": "POREQ",
    "quantity": 2
  }
]
```

Set stock level sample success response

HTTP satus code: 207

Body:

```
[
  {
    "status": "200",
    "body": null
  },
  {
    "status": "200",
    "body": null
  }
]
```

Set stock level sample error response

Example #1: Schema validation error

HTTP Status code: 400

Body:

```
[
  {
    "status": "400",
    "body": {
      "errors": [
        {
          "_message": "$[0].locationId: must be at least 32 characters
 long"
        },
        {
          "_message": "$[0].locationId: does not match the regex
 pattern ^[0-9A-F]{32}$"
        }
      ]
    }
```

```
  }
]
```

Example #2: Business error for an individual stock level command

HTTP Status code: 207

Body:

```
[
  {
    "status": "200",
    "body": null
  },
  {
    "status": "400",
    "body": {
      "errors": [
        {
          "_message": "The stock level with ID
 '0B1D5A6276AC401C9CC3C501CF1D501E' could not be updated because
 of the following error: The Safety Level value is greater than the
 Restock Level value."
        }
      ]
    }
  }
]
```

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

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.