Recipe API

30/06/2025 5:43 pm BST

Tags: Recipe API, BoM API, Workflow API

Versions

Versions this documentation is relevant for:

- 8.7.0+

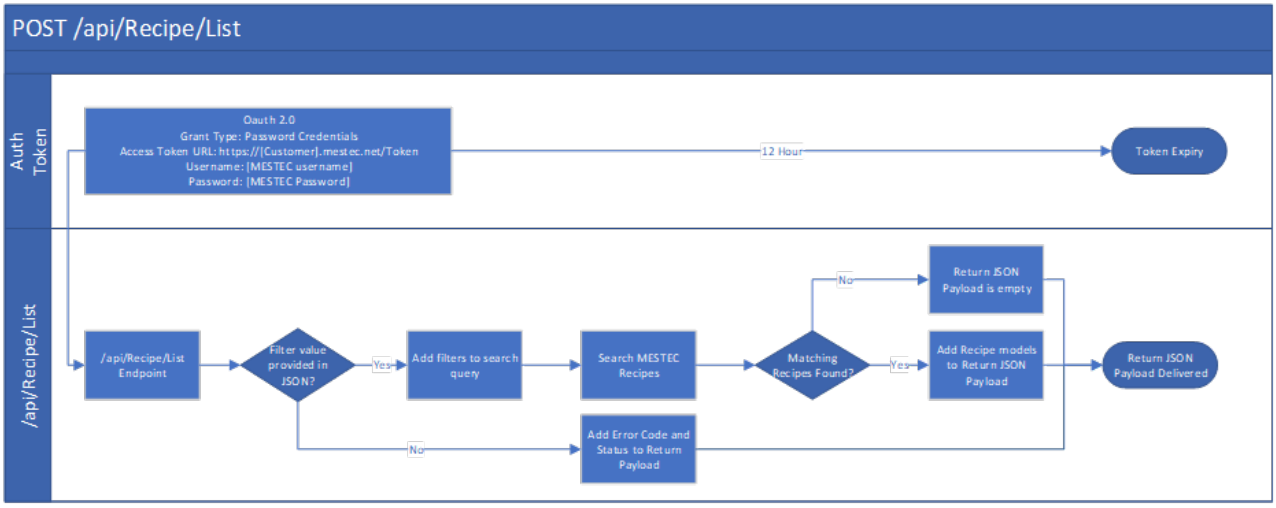
A recipe in Eyelit MES-M is the approved combination of Workflow and BoM used to create a product. A Scheduled Job is raised against a recipe.

List

The Recipe – List API call reads existing Recipes from Eyelit MES-M. The data can be filtered based on the parameters passed in the body/payload of the JSON packet giving the flexibility to search for one or multiple Recipes.

Using the Recipe – List API call has no impact on the data within the given Eyelit MES-M application, it is read-only.

Figure 1 - Logic within Recipe/List API Call



Data Prerequisites

There is no data required in Eyelit MES-M to act as a pre-requisite to make the API call to list Recipes. If parameters are passed in through the body that return no valid results, an empty payload will be returned.

Request

Table 1 shows the method and endpoint required to make the API call to list Scheduled Jobs.

Table 1 - Outbound Message Detail for Recipe - List

Method	URL Structure	Endpoint

POST	https://[environment].mestec.net	/api/Recipe/List

The body of the payload should follow the format below.

JSON Structure for Recipe List

```
{
  "id": 0,
  "suid": "string",
  "productName": "string",
  "name": "string",
  "workflow": {
    "name": "string",
    "suid": "string",
    "versionSUID": "string",
    "majorVersion": 0,
    "minorVersion": 0
  },
  "bom": {
    "name": "string",
    "suid": "string",
    "versionSUID": "string",
    "majorVersion": 0,
    "minorVersion": 0
  }
}
```

See Table 2 for information on which fields are optional, the appropriate data types and the mappings to fields in Eyelit MES-M.

Table 2 - Parameter Information for Schedule - List1

Parameter Name	Data Mapping	Data Type	Mandatory	Case Sensitive	Match Type
ID	recipe.id	NUMBER	No	N/A	Exact
suid	recipe.suid		No	No	Exact
productName	product.name	VARCHAR2(50 BYTE)	No	No	Exact
name	recipe.name	VARCHAR2(200 BYTE)	No	No	Exact
workflow	N/A	JSON Array	No	N/A	N/A
name	workflow.name	VARCHAR2(200 BYTE)	No	No	Exact
suid	workflow.suid	VARCHAR2(120 BYTE)	No	No	Exact
versionSUID	workflowVersion.suid	VARCHAR2(500 BYTE)	No	No	Exact
majorVersion	workflowVersion.majorVersion	NUMBER	No	N/A	N/A
minorVersion	workflowVersion.minorVersion	NUMBER	No	N/A	N/A
BOM	N/A	JSON Array	No	N/A	N/A
name	bom.name	VARCHAR2(100 BYTE)	No	No	Exact
suid	bom.suid	VARCHAR2(120 BYTE)	No	No	Exact
versionSUID	bomVersion.suid	VARCHAR2(120 BYTE)	No	No	Exact
majorVersion	bomVersion.majorVersion	NUMBER	No	N/A	N/A

minorVersion	bomVersion.minorVersion	NUMBER	No	N/A	N/A

Note: For any fields where the match type is 'Like', a percent symbol should be used as a wildcard character to indicate a number of characters within the given string.

Sample Request

See below for sample use cases with examples of the JSON payload format required.

To list Recipes where the name is 'ABC':

JSON Sample for Recipe List Request

```
{
  "name": "ABC"
}
```

Response

When using the Recipe – List API call, if any data has been found in Eyelit MES-M that meets the parameter values passed in the original payload, a JSON payload will be returned containing data in the following structure:

JSON Structure for Recipe List

```
[
  {
    "id": 0,
    "productName": "string",
    "workflow": "string",
    "worflowversion": "string",
    "bom": "string",
    "bomVersion": "string",
    "name": "string",
    "suid": "string",
    "description": "string",
    "changeReference": "string",
    "defaultItemVersion": "string",
    "approvalStatus": "string"
  }
]
```

Sample Response

See below for sample use cases with examples of the JSON payload format returned.

No results were found that matched given parameters:

JSON Sample for Recipe List Response A

```
{

}
```

Recipes found that matched given parameters:

JSON Sample for Recipe List Response B

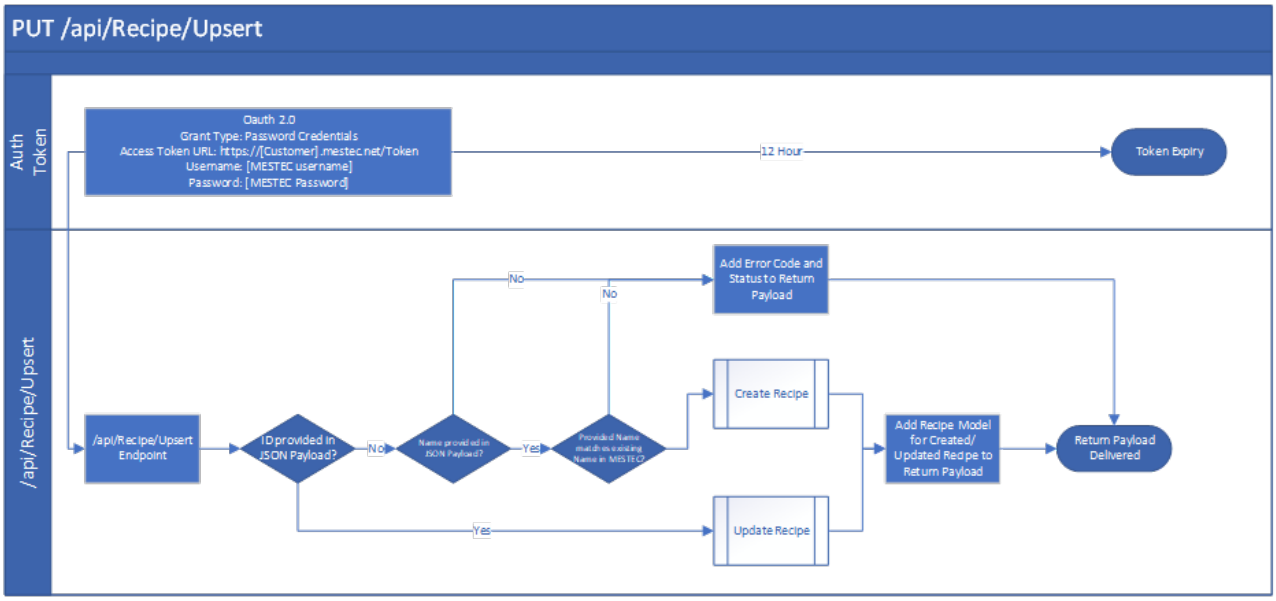
```
[
  {
    "id": 123,
    "productName": "ProductABC",
    "workflow": "WF0001",
    "worflowversion": "1.0",
    "bom": "BOM0001",
    "bomVersion": "1.1",
    "name": "ABC",
    "suid": "R-ABC",
    "description": "Default Recipe for Product ABC",
    "changeReference": "CR00001",
    "defaultItemVersion": "11",
    "approvalStatus": "Approved"
  }
]
```


Upsert

When Recipe ID is provided, The Recipe – Upsert API call checks if a Recipe with the given ID already exists. If it does, appropriate fields are updated in the matching Recipe in Eyelit MES-M.

When the Recipe ID is omitted, a new Recipe will be created.

Figure 2 - Logic within Recipe/Upsert API Call



Data Prerequisites

A recipe references a product, a BoM and a workflow, these must all exist in Eyelit MES-M prior to Recipe-Upsert.

Request

Table 3 shows the method and endpoint required to make the API call to Recipe-Upsert.

Table 3 - Outbound Message Detail for Recipe - Upsert

Method	URL Structure	Endpoint
PUT	https://[environment].mestec.net	/api/Recipe/Upsert

The body of the payload should follow the format below.

```
{
  "id": 0,
  "productName": "string",
  "workflow": "string",
  "worflowversion": "string",
  "bom": "string",
  "bomVersion": "string",
  "name": "string",
  "suid": "string",
  "description": "string",
  "changeReference": "string",
  "defaultItemVersion": "string",
  "approvalStatus": "string"
}
```

See Table 4 for information on which fields are optional, the appropriate data types and the mappings to fields in Eyelit MES-M.

Table 4 - Parameter Information for Product Group - Upsert

Parameter Name	Data Mapping	Data Type	Mandatory		Case Sensitive	Match Type
			Create	Update		
id	Recipe.id	NUMBER	N/A	Yes	N/A	Exact
productName	Product.name	VARCHAR2(200 BYTE)	Yes	No	No	Exact
workflow	Workflow.name	VARCHAR2(200 BYTE)	Yes	No	No	Exact
workflowVersion	Workflow.MajorVersion Workflow.MinorVersion	VARCHAR2(50 BYTE)	No	No	N/A	N/A
bom	Bom.name	VARCHAR2(100)	No	No	No	Exact
bomVersion	Bomversion.MajorVersion Bomversion.MinorVersion	VARCHAR2(50)	No	No	N/A	N/A
name	Recipe.name	VARCHAR2(200)	Yes	No	N/A	N/A
suid	Recipe.SUID	VARCHAR2(200)	No	No	N/A	N/A
description	Recipe.Comments	VARCHAR2(200 BYTE)	Yes	No	N/A	N/A
changeReference	Recipe.ChangeReference	NVARCHAR2(100 CHAR)	No	No	N/A	N/A
defaultItemVersion	Recipe.DefaultItemVersion	NVARCHAR2(100 CHAR)	Yes	No	N/A	N/A
approvalStatus	Approvalstatus.name	VARCHAR2(50)	No	No	No	Exact

Note: For any fields where the match type is 'Like', a percent symbol should be used as a wildcard character to indicate a number of characters within the given string.

Sample Request

See below for sample use cases with examples of the JSON payload format required.

To create a Recipe.

```
{
  "productName": "ProductABC",
  "workflow": "WF0001",
  "worflowversion": "1.0",
  "bom": "BOM0001",
  "bomVersion": "1.1",
  "name": "Recipe ABC",
  "suid": "R-ABC",
  "description": "Default Recipe for Product ABC",
  "changeReference": "CR00001",
  "defaultItemVersion": "11",
  "approvalStatus": "Approved"
}
```

To update description by ID.

```
{
  "id": 28,
  "description": "Updated through API Call"
}
```

Response

When using the Recipe – Upsert API call, if a Recipe has been Created or Updated a JSON payload will be returned containing data in the following structure:

```
[
  {
    "id": 0,
    "productName": "string",
    "workflow": "string",
    "worflowversion": "string",
    "bom": "string",
    "bomVersion": "string",
    "name": "string",
    "suid": "string",
    "description": "string",
    "changeReference": "string",
    "defaultItemVersion": "string",
    "approvalStatus": "string"
  }
]
```

Sample Response

See below for sample use cases with examples of the JSON payload format returned.

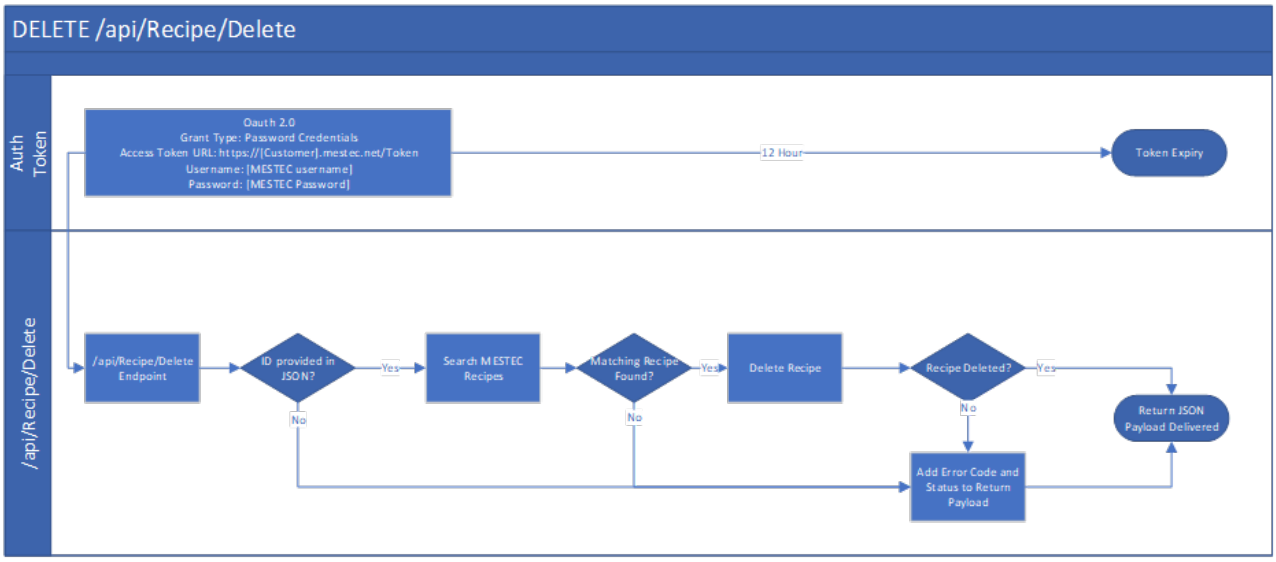
```
[
  {
    "id": 123,
    "productName": "ProductABC",
    "workflow": "WF0001",
    "worflowversion": "1.0",
    "bom": "BOM0001",
    "bomVersion": "1.1",
    "name": "Recipe ABC",
    "suid": "R-ABC",
    "description": "Default Recipe for Product ABC",
    "changeReference": "CR00001",
    "defaultItemVersion": "11",
    "approvalStatus": "Approved"
  }
]
```

Delete

The recipe – Delete API call attempts to delete a Recipe. A Recipe can be deleted as long as it has not been used.

Figure 3 shows the logic used within the Recipe – Delete API call.

Figure 3 - Logic within Recipe/Delete API Call



Data Prerequisites

In order to delete a Recipe it must exist in Eyelit MES-M and not currently be in use.

Request

Table 5 shows the method and endpoint required to make the API call to delete a Recipe.

Table 5 - Outbound Message Detail for Recipe - Delete

Method	URL Structure	Endpoint
DELETE	https://[environment].mestec.net	/api/Recipe/Delete

The body of the payload should follow the format below.

```
{
  "id": 0
}
```

See Table 6 - Parameter Information for Recipe - Delete for information on which fields are optional, the appropriate data types and the mappings to fields in Eyelit MES-M.

Table 6 - Parameter Information for Recipe - Delete

Parameter Name	Data Mapping	Data Type	Mandatory	Case Sensitive	Match Type
ID	recipe.id	NUMBER	Yes	N/A	Exact

Note: For any fields where the match type is 'Like', a percent symbol should be used as a wildcard character to indicate a number of characters within the given string.

Sample Request

See below for sample use cases with examples of the JSON payload format required.

Delete by ID

```
{
  "id": 28
}
```

Response

When using the Recipe – Delete API call, if a Scheduled Job has been Deleted a 200 status response will be returned.