Issue

23/10/2025 11:21 am BST

Issue

Tags: API

Versions

Versions this documentation is relevant for:

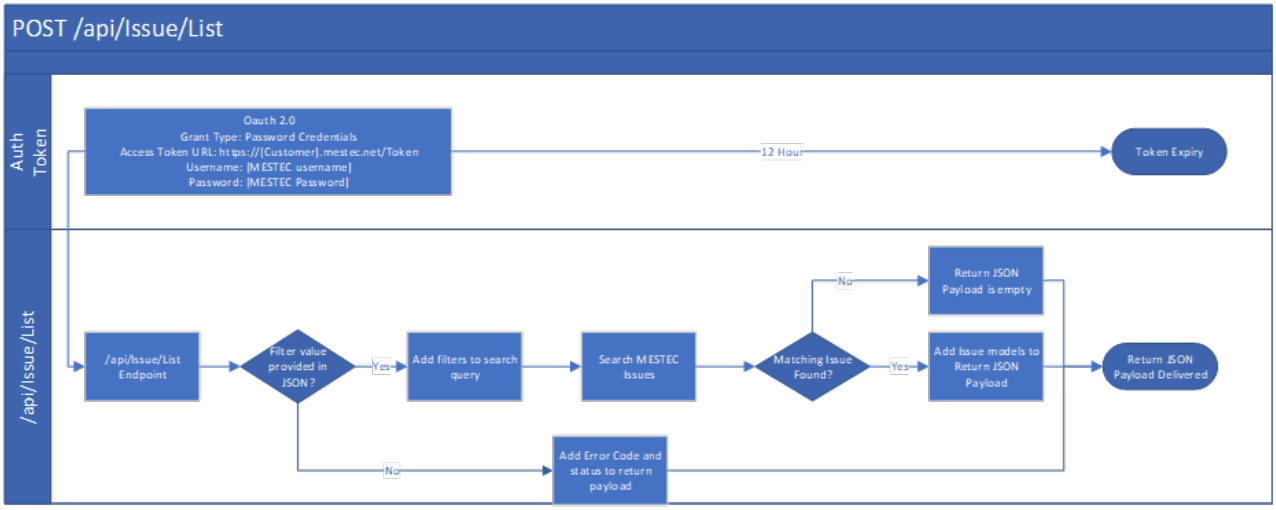
- *8.7.0+*

List

The Issue – List API call reads existing Issues 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 Issues.

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

Figure 1 - Logic within Issues/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 Issues. 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 Issues.

Table 1 - Outbound Message Detail for Issues - List

Method	URL Structure	Endpoint
POST	https://[environment].mestec.net	/api/Issues/List

The body of the payload should follow the format below.

JSON Structure for Issues List

```
{
  "id": 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 Issues - List

Parameter Name	Data Mapping	Data Type	Mandatory	Case Sensitive	Match Type
ID	issue.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.

To list Issues where the ID of the issue is 1234:

JSON Sample for Issues List Request B

```
{
  "id": 1234
}
```

Response

When using the Issues – 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 Issues List

```
[
  {
    "id": 0,
    "issueType": "string",
    "title": "string",
    "description": "string",
    "owner": "string",
    "raisedBy": "string",
    "externalRef": "string",
    "issueStatus": "string",
    "createdDT": "2024-09-24T08:19:29.539Z",
    "closedDT": "2024-09-24T08:19:29.539Z",
    "issueReason": "string",
    "qty": 0,
    "team": "string",
    "dueDT": "2024-09-24T08:19:29.539Z",
    "issueParentID": 0,
    "resolutionType": "string",
    "rank": 0,
    "completePercent": 0,
    "priority": "string",
    "itemScope": "string",
    "resolutionDetail": "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 Issues List Response A

```
{

}
```

Issue found that matched given parameters:

JSON Sample for Issues List Response B

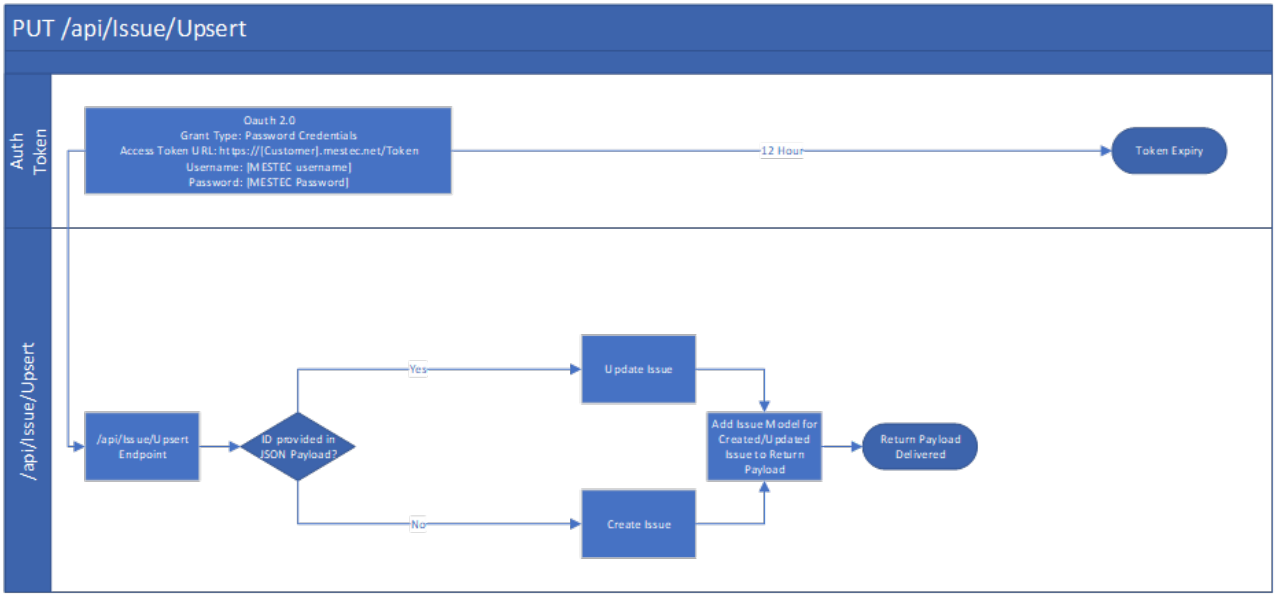

```
[
    {
        "ID": 119,
        "Title": "12345",
        "Description": "12345",
        "Owner": "Adam",
        "RaisedBy": "kenny",
        "ExternalRef": "12345",
        "IssueStatus": "Open",
        "CreatedDT": "2018-08-10T14:58:35.39556Z",
        "IssueReason": "ToniesIssueReason",
        "CompletePercent": 0,
        "Priority": "Medium",
        "ItemScope": "Item",
        "ResolutionDetail": ""
    }
]
```


Upsert

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

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

Figure 2 - Logic within Issue/Upsert API Call



Data Prerequisites

The following data entities must exist if they are referenced in an upsert call.

- Issue Type
- Owner (User Account)
- Raised By (User Account)
- Issue Status
- Issue Reason
- Issue Priority
- Team
- Resolution Type
- Item Scope

Request

Table 3 shows the method and endpoint required to make the API call to upsert Issue.

Table 3 - Outbound Message Detail for Issue - Upsert

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

The body of the payload should follow the format below.

```
{
  "id": 0,
  "issueType": "string",
  "title": "string",
  "description": "string",
  "owner": "string",
  "raisedBy": "string",
  "externalRef": "string",
  "issueStatus": "string",
  "createdDT": "2024-09-24T08:19:29.538Z",
  "closedDT": "2024-09-24T08:19:29.538Z",
  "issueReason": "string",
  "qty": 0,
  "team": "string",
  "dueDT": "2024-09-24T08:19:29.538Z",
  "issueParentID": 0,
  "resolutionType": "string",
  "rank": 0,
  "completePercent": 0,
  "priority": "string",
  "itemScope": "string",
  "resolutionDetail": "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 Issue - Upsert

Parameter Name	Data Mapping	Data Type	Mandatory	Case Sensitive	Match Type
ID	issue.id	NUMBER	No	N/A	Exact
IssueType	issuetype.name	VARCHAR2(50)	No	No	Exact
Title	issue.title	VARCHAR2(100)	No	N/A	N/A
Description	issue.description	VARCHAR2(2000 BYTE)	No	N/A	N/A
Owner	useraccount.username	VARCHAR2(50)	No	No	Exact
ExternalRef	issue.externalref	VARCHAR2(20)	No	N/A	N/A
IssueStatus	issuestatus.status	VARCHAR2(20)	No	No	Exact
ClosedDT	issue.closeddt	TIMESTAMP(6)	No	N/A	N/A
IssueReason	issuereason.reason	VARCHAR2(20)	No	No	Exact
Qty	issue.quantity	NUMBER(5)	No	N/A	N/A
Team	team.name	VARCHAR2(50)	No	No	Exact
DueDT	issue.duedt	TIMESTAMP(6)	No	N/A	N/A
IssueParentID	issue.issueparentid	NUMBER(10)	No	No	Exact
ResolutionType	issueresolutiontype.resolutiontype	VARCHAR2(100)	No	No	Exact
Rank	issue.rank	NUMBER(10)	No	N/A	N/A
CompletePercent	issue.completepercent	NUMBER(3)	No	N/A	N/A

Priority	Issuepriority.name	VARCHAR2(100)	No	No	Exact
ItemScope	issueitemscope.name	VARCHAR2(20)	No	No	Exact
ResolutionDetail	issue.resolutiondetails	VARCHAR2(2000)	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 create an Issue.

```
{
  "issueType": "Quality",
  "title": "APITest",
  "description": "Issue Created through API Call",
  "owner": "IntegrationAccount",
  "externalRef": "TEST123",
  "issueStatus": "Open",
  "issueReason": "QC Reason 1",
  "qty": 100,
  "team": "Test team A",
  "dueDT": "2024-09-30T00:00:00.000Z",
  "rank": 1,
  "completePercent": 0,
  "priority": "High",
  "itemScope": "Job"
}
```

To update title and description by ID.

```
{
    "ID": 6782,
    "Title": "APITest-Updated",
    "Description": "APITest Updated"
}
```

Response

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

```
[
  {
    "ID": 0,
    "Title": "string",
    "Description": "string",
    "Owner": "string",
    "RaisedBy": "string",
    "ExternalRef": "string",
    "IssueStatus": "string",
    "CreatedDT": "2024-09-24T11:29:49.76676Z",
    "IssueReason": "string",
    "Qty": 0,
    "Team": "string",
    "DueDT": "2024-09-30T00:00:00Z",
    "IssueParentID": 0,
    "Rank": 0,
    "CompletePercent": 0,
    "Priority": "string",
    "ItemScope": "string",
    "ResolutionDetail": "string"
}
]
```


Sample Response

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

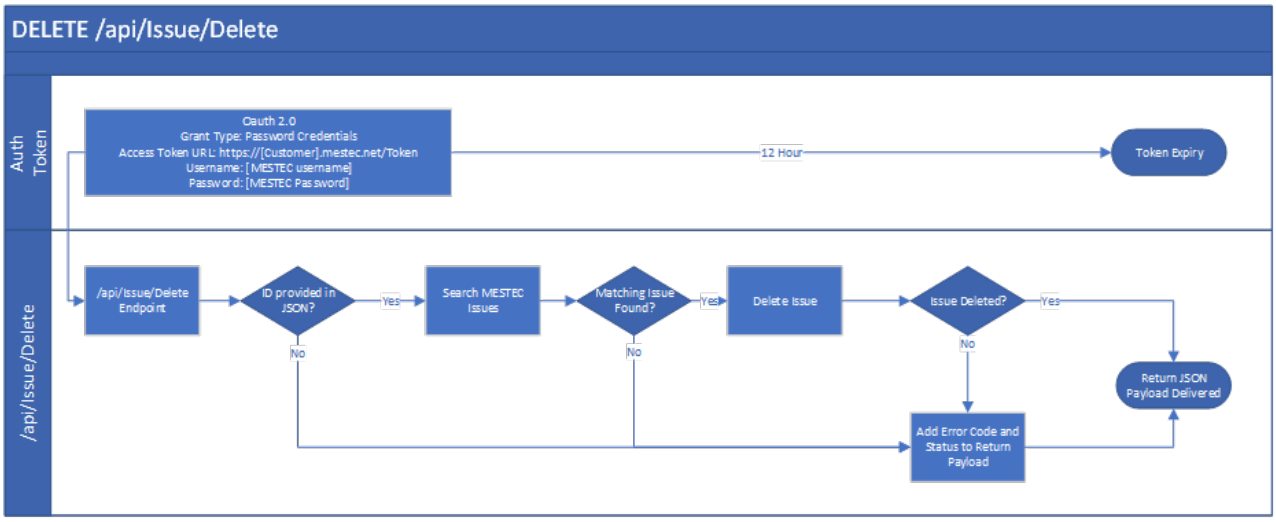

```
{
    "ID": 6782,
    "Title": "APITest-Updated",
    "Description": "APITest-Updated",
    "Owner": "IntegrationAccount",
    "RaisedBy": "IntegrationAccount",
    "ExternalRef": "APITest-Updated",
    "IssueStatus": "Open",
    "CreatedDT": "2024-09-24T11:29:49.76676Z",
    "IssueReason": "QC Reason 1",
    "Qty": 100.0,
    "Team": "Test team A",
    "DueDT": "2024-09-30T00:00:00Z",
    "IssueParentID": 1,
    "Rank": 1,
    "CompletePercent": 0,
    "Priority": "High",
    "ItemScope": "Job",
    "ResolutionDetail": ""
}
```


Delete

The Issue – Delete API call attempts to delete an Issue. An Issue can be deleted as long as there are no Issue Items linked to the Issue.

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

Figure 3 - Logic within Issue/Delete API Call



Data Prerequisites

In order to delete an Issue, the Issue must exist in Eyelit MES-M. The Issue must have no Issue Items linked.

Request

Table 5 shows the method and endpoint required to make the API call to delete an Issue.

Table 5 - Outbound Message Detail for Issue - Delete

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

The body of the payload should follow the format below.

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

See Table 7 for information on which fields are optional, the appropriate data types and the mappings to fields in Eyelit MES-M.

Table 7 - Parameter Information for Issue - Delete

Parameter Name	Data Mapping	Data Type	Mandatory	Case Sensitive	Match Type
ID	issue.id	NUMBER	No	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 Issue – Delete API call, if an Issue has been Deleted a null JSON payload will be returned.