

Product - Product Group

30/06/2025 8:38 am BST

Product
Tags: API

Versions

Versions this documentation is relevant for:

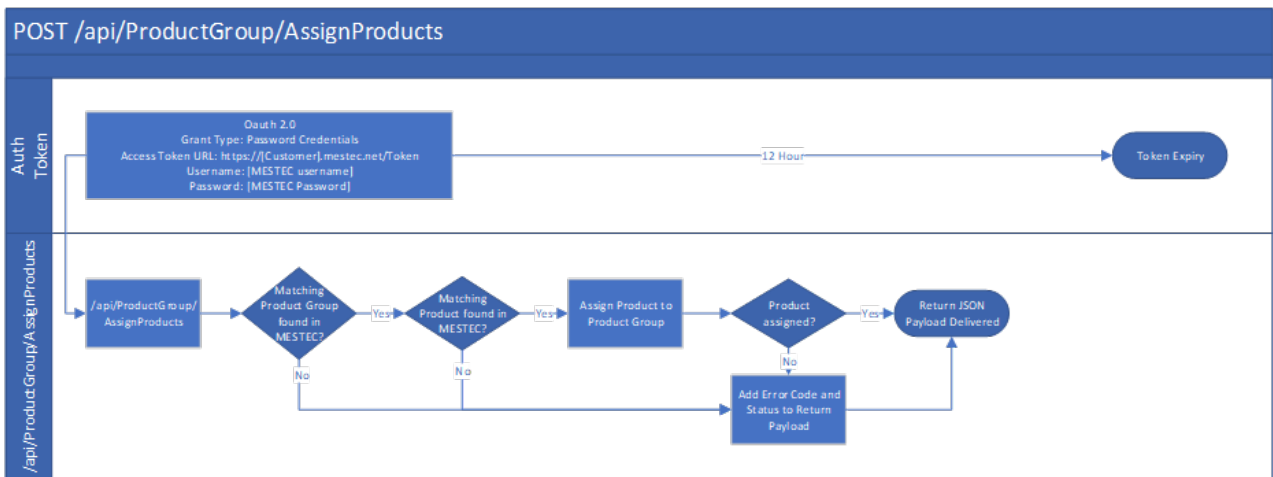
- 8.7.0+

AssignProducts

The Product Group – AssignProducts API call attempts to assign a Product to a Product Group .

Figure 1 shows the logic used within the Product Group – AssignProducts API call.

Figure 1 - Logic within ProductGroup/AssignProducts API Call



Prerequisites

When assigning a Product to a Product Group, both the Product and Product Group must exist in Eyelit MES-M.

Request

Table 1 shows the method and endpoint required to make the API call to AssignProducts.

Table 1 - Outbound Message Detail for Product Group - AssignProducts

Method	URL Structure	Endpoint
POST	https://[environment].mestec.net	/api/ProductGroup/AssignProducts

The body of the payload should follow the format below.

```
{
  "id": 0,
  "products": [
    "string"
  ]
}
```

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 ProductGroup - AssignProducts

Parameter Name	Data Mapping	Data Type	Mandatory	Case Sensitive	Match Type
id	productgroup.id	NUMBER	No	N/A	Exact
Products	product.name	VARCHAR2(250 BYTE)	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.

Assign by ID

```
{
  "id": 28,
  "products": [
    "Product_1"
  ]
}
```

Response

When using the Product Group – AssignProducts API call, if a Product has been assigned from a Product Group, a JSON payload will be returned containing data in the following structure:

```
[
  {
    "id": 0,
    "name": "string" ,
    "Products": [
      "string",
      "string"
    ]
  }
]
```

Sample Response

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

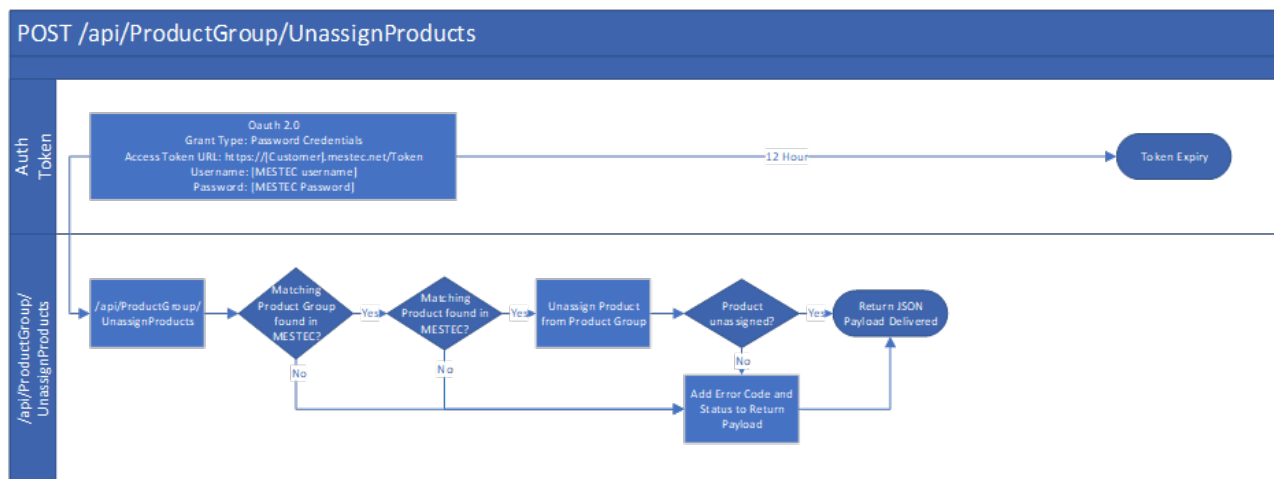
```
[
  {
    "id": 4849,
    "name": "ProductGroup",
    "Products": [
      "Product1",
      "Product2"
    ]
  }
]
```

UnassignProducts

The Product Group – UnassignProducts API call attempts to unassign a Product from a Product Group.

Figure 2 shows the logic used within the Product Group – UnassignProducts API call.

Figure 2 - Logic within ProductGroup/UnassignProducts API Call



Prerequisites

When unassigning a Product from a Product Group, both the Product and Product Group must exist in Eyleit MES-M, Additionally, the Product must be assigned to the Product Group.

Request

Table 3 shows the method and endpoint required to make the API call to unassign a Product from a Product Group.

Table 3 - Outbound Message Detail for Product Group - UnAssignProducts

Method	URL Structure	Endpoint
POST	https://[environment].mestec.net	/api/ProductGroup/UnassignProducts

The body of the payload should follow the format below.

```
{
  "id": 0,
  "Products": [
    "string",
    "string"
  ]
}
```

```
}
```

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 ProductGroup -UnAssignProducts

Parameter Name	Data Mapping	Data Type	Mandatory	Case Sensitive	Match Type
ID	productgroup.id	NUMBER	Yes	N/A	Exact
Products	product.name	VARCHAR2(50 BYTE)	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. See examples in Section 3.1.3.1 for use cases.

Sample Request

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

Unassign by ID

```
{
  "id": 12,
  "Products": [
    "Product_1",
    "Product_2"
  ]
}
```

Response

When using the ProductGroup – Unassign API call, if a Product has been unassigned from a Product Group, a JSON payload will be returned containing data in the following structure:

```
[
  {
    "id": 0,
    "name": "string" ,
    "Products": [
      "string",
      "string"
    ]
  }
]
```

Sample Response

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

```
[
  {
    "id": 0,
    "name": "Product_Grp" ,
    "Products": [
      "Product_3",
      "Product_4"
    ]
  }
]
```

