

#####

API Documentation for Product module in MyToysGroup

Total API's - 3

#####

API - 1

API Description: API to upload the products as CSV into the datastore in MyToysGroup

API URL: http://{BASE_URL}/product/upload

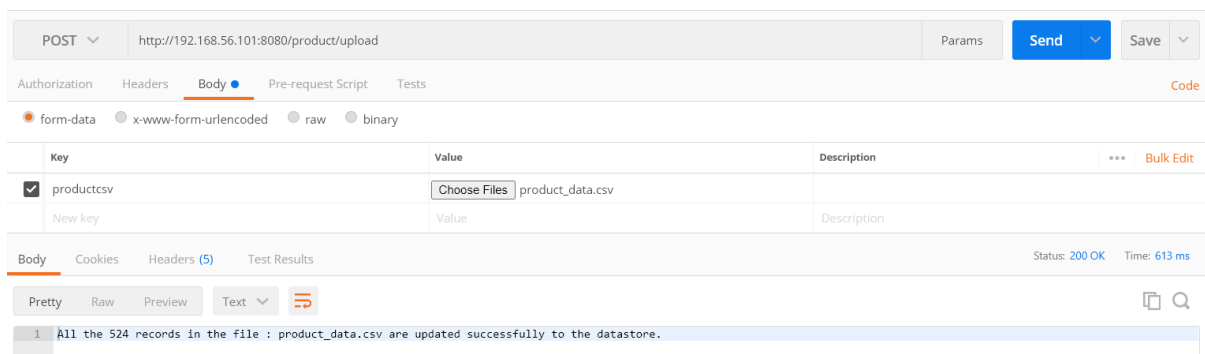
API Method Type: POST

API Content Type: text/csv

Upload Field Name: productcsv

Response: Json

Also added the validation for invalid headers / no input etc.,



POST http://192.168.56.101:8080/product/upload

Authorization Headers Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
productcsv	Choose Files product_data.csv	
New key	Value	Description

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 613 ms

Pretty Raw Preview Text

```
1 All the 524 records in the file : product_data.csv are updated successfully to the datastore.
```

#####

API - 2

#####

API Description: API to GET / GLOBAL / EXTENSIVE SEARCH on the products of MyToysGroup

API URL: http://{BASE_URL}/product

API Method Type: GET

Response: Json

```
{
  "data": [XXXX],
  "total": 1200,
  "totalRecordInCurrentSearch": 20,
  "scrollId": "XXXX"
}
```

Possible Inputs:

This service is used to get all the products based on the following criteria

- Search Keyword on the field like ID, NAME, BRAND, STOCK
- Sort on all the fields
- Limit by Size
- Pagination

Search Criteria 1: API to get all the data of the product

Search Criteria 2: API to get all the data which matches to the search text

Search Criteria 3: API to get all the data which will be sorted by accepted fields

Search Criteria 4: API to get only the limited sets of data based on limit size

Search Criteria 5: API to get data based on pagination.

Search Criteria 6: API to get data based on all the combinations of search, sort by field, limit

Case 1

Search Criteria 1: API to get all the data

<http://192.168.56.101:8080/product>

The screenshot shows a REST client interface with a GET request to <http://192.168.56.101:8080/product>. The response is a JSON array of product data. The status is 200 OK and the time taken is 838 ms.

```
{
  "scrollId": "FgluY2x1ZGVFY29udGV4dF91dW1kdXF1ZXJ5QW5kRmV0Y2gBFFBFZkZESFFCZUZVc3VfeWZTMXB1AAAAAqIwBtBgcdHUNHdSV21UcERSNm4zY3FOZw==",
  "data": [
    {
      "PRICE": "50.95",
      "OLD_PRICE": "54.95",
      "ID": "10258",
      "STOCK": "0",
      "NAME": "NIKE #3561135",
      "BRAND": "NIKE"
    },
    {
      "PRICE": "39.99",
      "OLD_PRICE": "39.99",
      "ID": "10256",
      "STOCK": "0",
      "NAME": "s.Oliver #39079",
      "BRAND": "s.Oliver"
    },
    {
      "PRICE": "31.99",
      "OLD_PRICE": "39.99",
      "ID": "10416",
      "STOCK": "0",
      "NAME": "EB KIDS #1930809",
      "BRAND": "EB KIDS"
    }
  ]
}
```

Case 2

Search Criteria 1: API to get all the data

<http://192.168.56.101:8080/product?search=Tommy>

GET http://192.168.56.101:8080/product?search=Tommy Params Send Save

Authorization Headers Body Pre-request Script Tests Code

Key	Value	Description	*** Bulk Edit Presets
New key	Value	Description	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 216 ms

Pretty Raw Preview JSON

```
1 {
2   "scrollId": "FGluY2x1ZGVFY29udGV4dF91dW1kDXF1ZXJ5QW5kRmV0Y2gBFE8wZkxESFFCZUZVc3VfeWZlYXAAAAAUAUmbTBGcDhUJHdSV21UcER5Im4zY3FOZUw==",
3   "data": [
4     {
5       "PRICE": "15.99",
6       "OLD_PRICE": "49.99",
7       "ID": "10475",
8       "STOCK": "0",
9       "NAME": "TOMMY HILFIGER #803396",
10      "BRAND": "TOMMY HILFIGER"
11    },
12    {
13      "PRICE": "27.99",
14      "OLD_PRICE": "27.99",
15      "ID": "1552",
16      "STOCK": "19",
17      "NAME": "TOMMY HILFIGER #208032",
18      "BRAND": "TOMMY HILFIGER"
19    },
20    {
21      "PRICE": "54.99",
22      "OLD_PRICE": "54.99",
23      "ID": "24185",
24      "STOCK": "4",
25      "NAME": "TOMMY HILFIGER #1128833",
26      "BRAND": "TOMMY HILFIGER"
27    },
28  ]
29 }
```

Case 3

Search Criteria 3: API to get all the data which will be sorted by accepted fields

http://192.168.56.101:8080/product?sort=NAME

Accepted Fields : "ID", "NAME", "PRICE", "OLD_PRICE", "STOCK", "BRAND"

GET http://192.168.56.101:8080/product?sort=NAME Params Send Save

Authorization Headers Body Pre-request Script Tests Code

Key	Value	Description	*** Bulk Edit Presets
New key	Value	Description	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 34 ms

Pretty Raw Preview JSON

```
1 {
2   "scrollId": "FGluY2x1ZGVFY29udGV4dF91dW1kDXF1ZXJ5QW5kRmV0Y2gBFDfZk1ESFFCZUZVc3VfeWZlYXAAAAAUAUmbTBGcDhUJHdSV21UcER5Im4zY3FOZUw==",
3   "data": [
4     {
5       "PRICE": "31.99",
6       "OLD_PRICE": "39.99",
7       "ID": "10416",
8       "STOCK": "0",
9       "NAME": "EB KIDS #1930809",
10      "BRAND": "EB KIDS"
11    },
12    {
13      "PRICE": "44.99",
14      "OLD_PRICE": "49.99",
15      "ID": "43768",
16      "STOCK": "3",
17      "NAME": "ESPRIT #11388875",
18      "BRAND": "ESPRIT"
19    },
20    {
21      "PRICE": "8.99",
22      "OLD_PRICE": "9.99",
23      "ID": "47901",
24      "STOCK": "1",
25      "NAME": "ESPRIT #12046619",
26      "BRAND": "ESPRIT"
27    },
28  ]
29 }
```

Case 4

Search Criteria 4: API to get only the limited sets of data based on limit size

http://192.168.56.101:8080/product?size=20

```
GET http://192.168.56.101:8080/product?size=20

{
  "total": 521,
  "totalRecordInCurrentSearch": 20
}
```

Search Criteria 5: API to get data based on pagination.

In any search call, we will have scroll id used for pagination

```
GET http://192.168.56.101:8080/product?size=20

{"scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDXF1ZXJ5QW5kRmV0Y2gBF5rZk5ESFFCZUZVc3VfeWZ4RjFTAAAAAaAwgWbTBGcDhUNHdSV21UcER5Nm4zY3FOZw==",
  "data": [
    {
      "PRICE": "59.95",
      "OLD_PRICE": "54.95",
      "ID": "10250",
      "STOCK": "0",
      "NAME": "NIKE #3561135",
      "BRAND": "NIKE"
    },
    {
      "PRICE": "39.99",
      "OLD_PRICE": "39.99",
      "ID": "10256",
      "STOCK": "0",
      "NAME": "NIKE #3561135",
      "BRAND": "NIKE"
    }
  ]
}
```

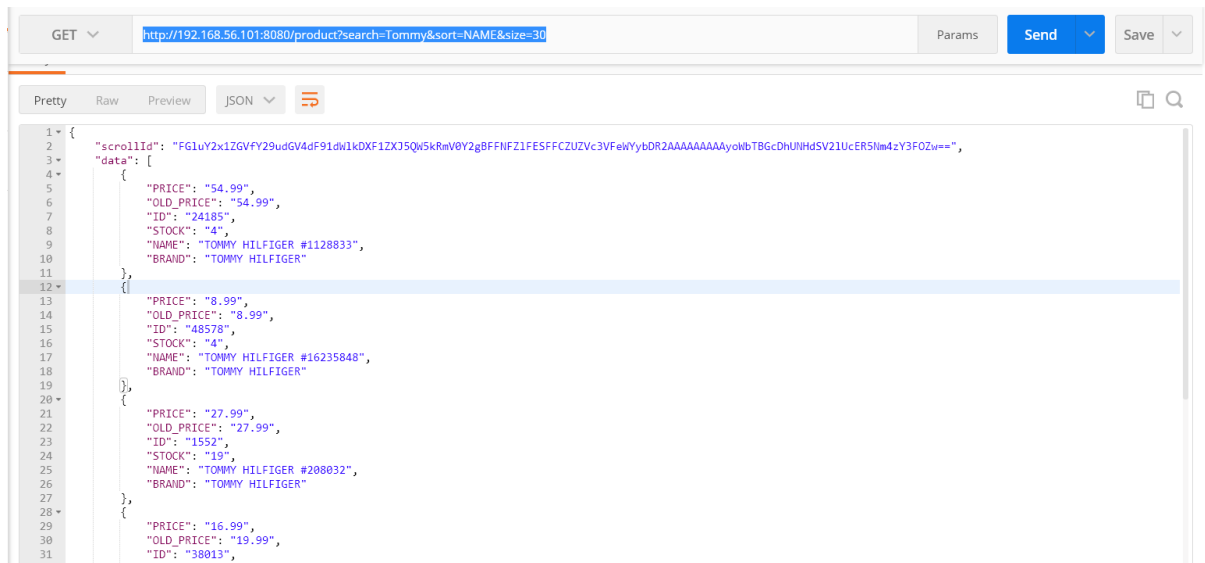
<http://192.168.56.101:8080/product?scrollId=FGluY2x1ZGVfY29udGV4dF91dWlkDXF1ZXJ5QW5kRmV0Y2gBF5rZk5ESFFCZUZVc3VfeWZ4RjFTAAAAAaAwgWbTBGcDhUNHdSV21UcER5Nm4zY3FOZw==>

```
GET http://192.168.56.101:8080/product?scrollId=FGluY2x1ZGVfY29udGV4dF91dWlkDXF1ZXJ5QW5kRmV0Y2gBF5rZk5ESFFCZUZVc3VfeWZ4RjFTAAAAAaAwgWbTBGcDhUNHdSV21UcER5Nm4zY3FOZw==

{
  "scrollId": "FGluY2x1ZGVfY29udGV4dF91dWlkDXF1ZXJ5QW5kRmV0Y2gBF5rZk5ESFFCZUZVc3VfeWZ4RjFTAAAAAaAwgWbTBGcDhUNHdSV21UcER5Nm4zY3FOZw==",
  "data": [
    {
      "PRICE": "80.99",
      "OLD_PRICE": "99.99",
      "ID": "11781",
      "STOCK": "54",
      "NAME": "LEGO #2231871",
      "BRAND": "LEGO"
    },
    {
      "PRICE": "42.99",
      "OLD_PRICE": "49.99",
      "ID": "11789",
      "STOCK": "141",
      "NAME": "LEGO #3911129",
      "BRAND": "LEGO"
    },
    {
      "PRICE": "19.99",
      "OLD_PRICE": "22.99",
      "ID": "11919",
      "STOCK": "5",
      "NAME": "SIKU #2148864",
      "BRAND": "SIKU"
    },
    {
      "PRICE": "25.99",
      "OLD_PRICE": "33.99",
      "ID": "12025",
      "STOCK": "389",
      "NAME": "PLAYMOBIL #2157081",
      "BRAND": "PLAYMOBIL"
    }
  ]
}
```

Search Criteria 6: API to get data based on all the combinations of search, sort by field, limit

<http://192.168.56.101:8080/product?search=Tommy&sort=NAME&size=30>



#####

API - 3

API Description: API to GET / GLOBAL / EXTENSIVE SEARCH on the products of MyToysGroup using RANGE Query

API URL: http://{BASE_URL}/product

API Method Type: POST

Response: Json

```
{
  "data": [XXXX],
  "total": 1200,
  "totalRecordInCurrentSearch": 20,
  "scrollId": "XXXX"
}
```

Possible Input :

```
{
  "field": "PRICE",
  "startvalue": 10,
  "endvalue": 20
}
```

```
}
```

Accepted Values in field: PRICE and OLD_PRICE

[Search Criteria 7:](#) API to get filter the data by range

<http://localhost:8080/product>

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/product`. The request body is a JSON object: `{ "field": "PRICE", "startvalue": 10, "endvalue": 20 }`. The response status is 200 OK, and the response body is a JSON array of two product objects. The first object has `"PRICE": "16.99"`, `"OLD_PRICE": "17.99"`, `"ID": "28348"`, `"STOCK": "86"`, `"NAME": "PLAYMOBIL #6962213"`, and `"BRAND": "PLAYMOBIL"`. The second object has `"PRICE": "18.99"`, `"OLD_PRICE": "19.99"`, `"ID": "18171"`, `"STOCK": "43"`, and `"NAME": "SIKU #4143756"`.

```
1 {
2   "field": "PRICE",
3   "startvalue": 10,
4   "endvalue": 20
5 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 433 ms

Pretty Raw Preview JSON

```
1 {
2   "scrollId": "FgluY2x1ZGVfY29udGV4dF91dWlkDXF1ZXJ5QW5kRmV0Y2gBFHmeXpDwFFCMUNfcTRQ0WxkR1p6AAAAAAAHKwMbTBGcDhUNHdSV21UcER5Nm4zY3FOZv==",
3   "data": [
4     {
5       "PRICE": "16.99",
6       "OLD_PRICE": "17.99",
7       "ID": "28348",
8       "STOCK": "86",
9       "NAME": "PLAYMOBIL #6962213",
10      "BRAND": "PLAYMOBIL"
11     },
12     {
13       "PRICE": "18.99",
14       "OLD_PRICE": "19.99",
15       "ID": "18171",
16       "STOCK": "43",
17       "NAME": "SIKU #4143756",
18       "BRAND": "SIKU"
19     }
20   ]
21 }
```

Pagination can also be done using the `scrollId` (as described in API 2 for Range Query too)

