Node Version	**:**	**16.13.0**
Mongo version	**:**	**5.0.4**
Robo 3T	**:**	**1.4.4**

Tools : Visual Studio Code, PostMan, Robo 3T

Create A User Login

Method : Post
url : http://localhost:7000/user/createloginorreg

This above Api is to create a new user Login

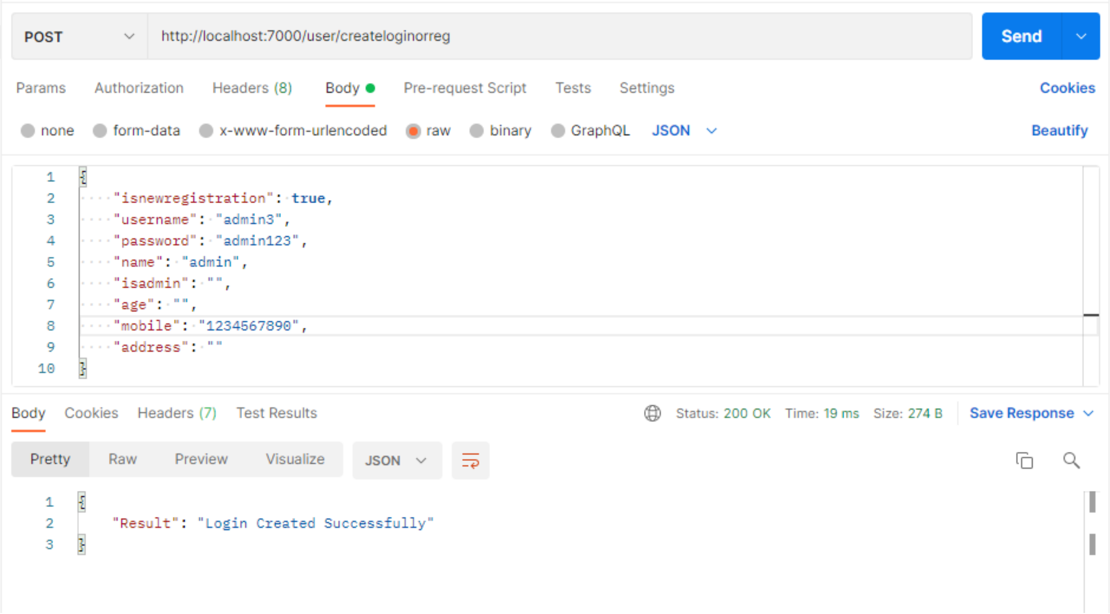
```
{
    "isnewregistration": true,
    "username": "admin2",
    "password": "admin123",
    "name": "admin",
    "isadmin": "",
    "age": "",
    "mobile": "1234567890",
    "address": ""
}
```

This above JSON is passed in body to create login for user, and "isnewregistration" should be default **true** for creating new user. If the field “isadmin“ is **true** (boolean) it creates the login for admin user

Note : Admin login can only upload (Insert / Update) the items.

Once JSON posted, Login will created to that user and will get response as

```
{
    "Result": "Login Created Successfully"
}
```

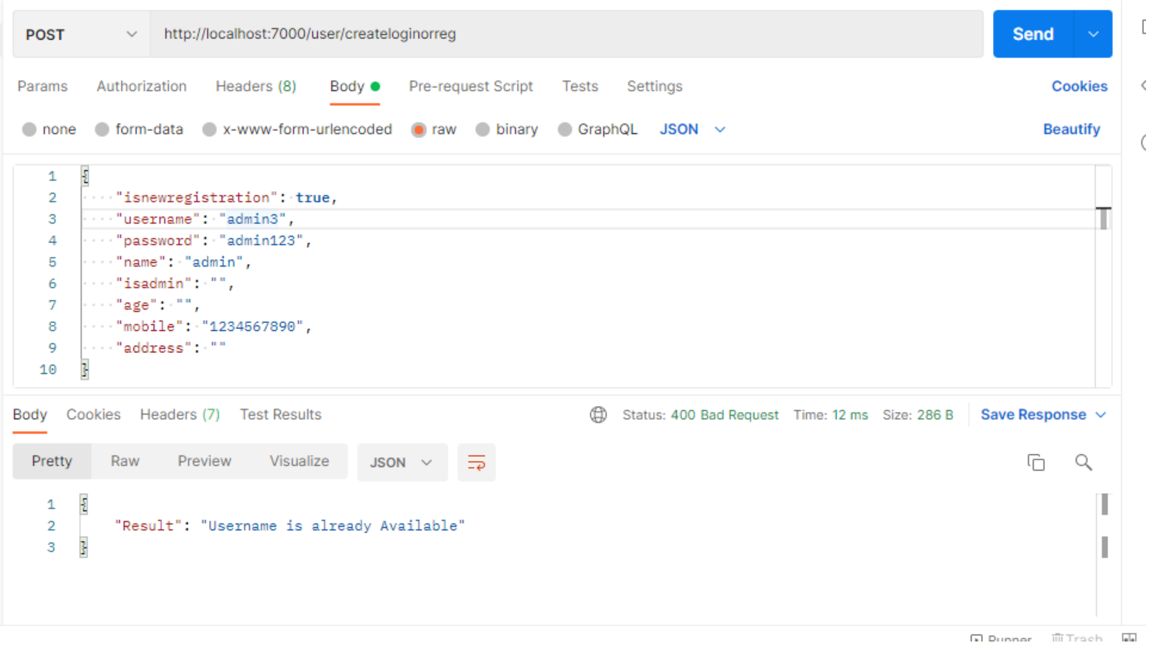


```
{
    "isnewregistration": true,
    "username": "admin3",
    "password": "admin123",
    "name": "admin",
    "isadmin": "",
    "age": "",
    "mobile": "1234567890",
    "address": ""
}
```

```
{
    "Result": "Login Created Successfully"
}
```

If the username is exist the response will be as

```
{
    "Result": "Username is already available"
}
```



```
{
    "isnewregistration": true,
    "username": "admin3",
    "password": "admin123",
    "name": "admin",
    "isadmin": "",
    "age": "",
    "mobile": "1234567890",
    "address": ""
}
```

```
{
    "Result": "Username is already Available"
}
```

Login with username and Password

Method : Post
url : http://localhost:7000/user/createloginorreg

This Api is to login with username and password

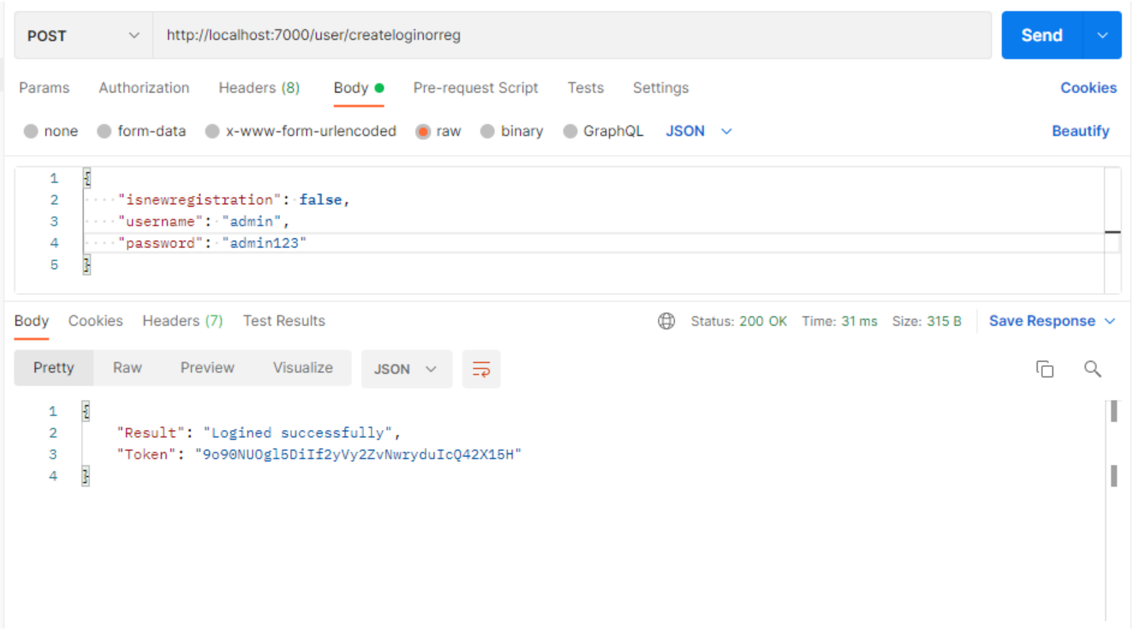
```
{
    "isnewregistration": false,
    "username": "admin",
    "password": "admin123"
}
```

This above JSON is passed in body to login and the field isnewregistration should be default false (boolean).

Once the this JSON passed. it validate the username and password and will get response and generate a user Token.

```
{
    "Result": "Logined successfully",
    "Token": "299g0ON7oz4kowJAVDnw1a6ciAiVTV5UBX49"
}
```

Note : This Token is valid for 8 min, after that it will be invalid and user has to re-login and get new user token.



POST http://localhost:7000/user/createloginorreg

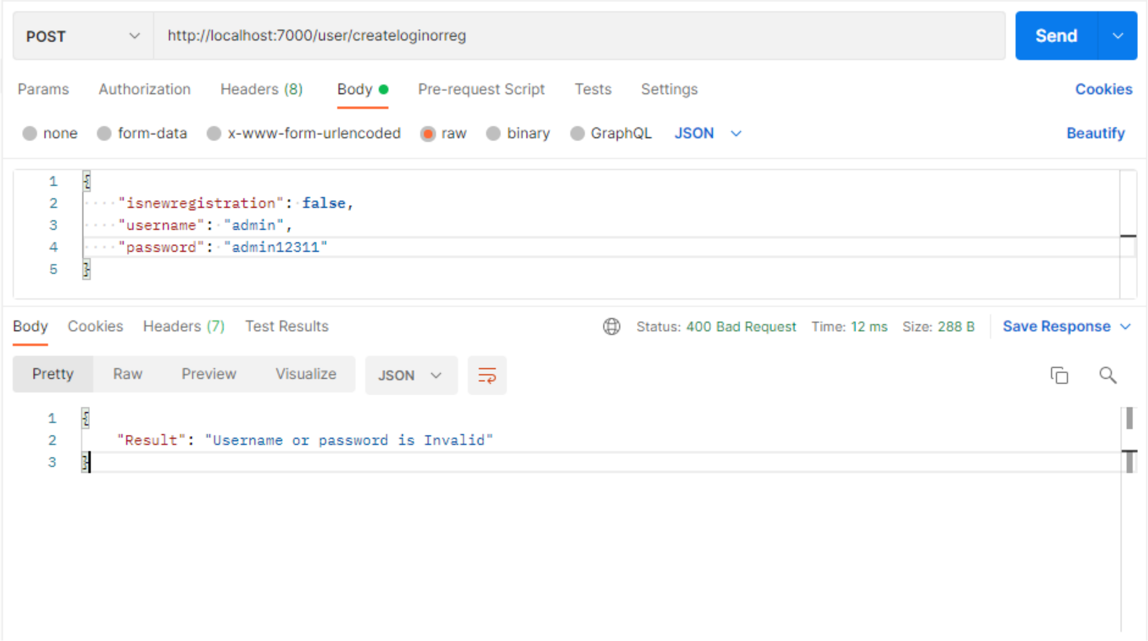
```
{
    "isnewregistration": false,
    "username": "admin",
    "password": "admin123"
}
```

Status: 200 OK Time: 31 ms Size: 315 B

```
{
    "Result": "Logined successfully",
    "Token": "9o90NUOgl5DiIf2yVy2ZvNwryduIcQ42X15H"
}
```

If the passed username or password is invalid or incorrect the response will be as

```
{
    "Result": "Username or password is Invalid"
}
```



POST http://localhost:7000/user/createloginorreg

```
{
    "isnewregistration": false,
    "username": "admin",
    "password": "admin12311"
}
```

Status: 400 Bad Request Time: 12 ms Size: 288 B

```
{
    "Result": "Username or password is Invalid"
}
```

Generating Auth Token

Method : Get

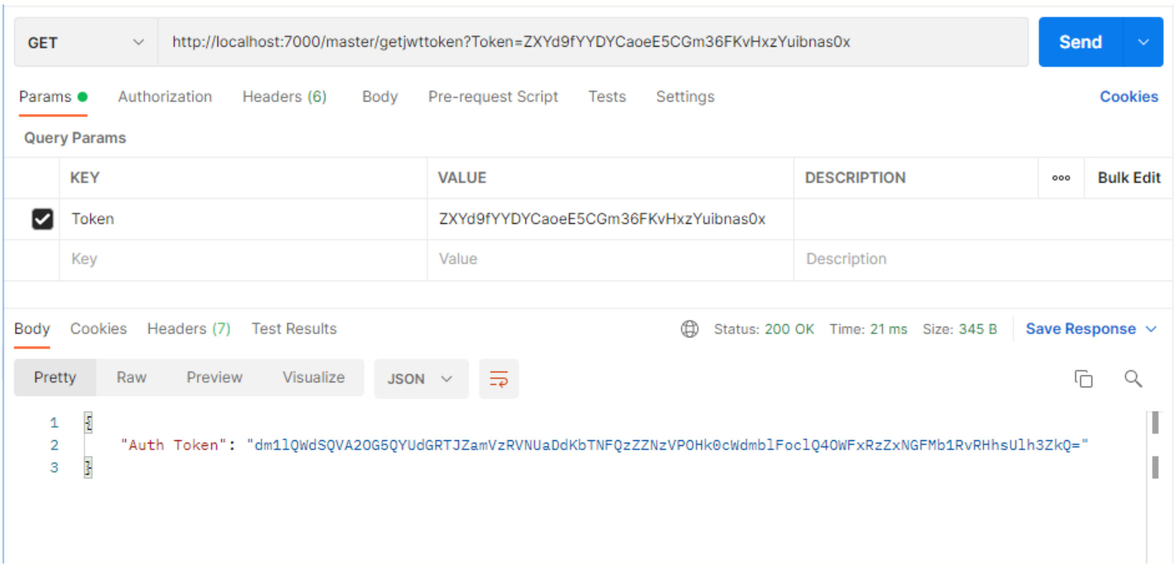
url : http://localhost:7000/master/getjwttoken

This Api generates the Auth Token to access and use the other Api (Insert / update items, Viewing the orders).

Once login is successful add “KEY” as Token and “VALUE” as user token in params which is generated at time of login and Execute, it will generate the Auth Token.

Note : Auth Token is valid for only one process, once the process is completed the Auth Token will be invalid so user has to re-generate new Auth Token to access.

```
{
    "Auth Token": "Tzd5OFNmcFJ0MHdGa1c1UGhGM0g4NjhENVk5UDY2UFNHOU4xNWxzcjVURUprZDFI
N0tmTTk4eHVyc2k4Nm45U3ZmcDg="
}
```



GET http://localhost:7000/master/getjwttoken?Token=ZXYd9fYYDYCaoeE5CGm36FKvHxzYuibnas0x

Query Params

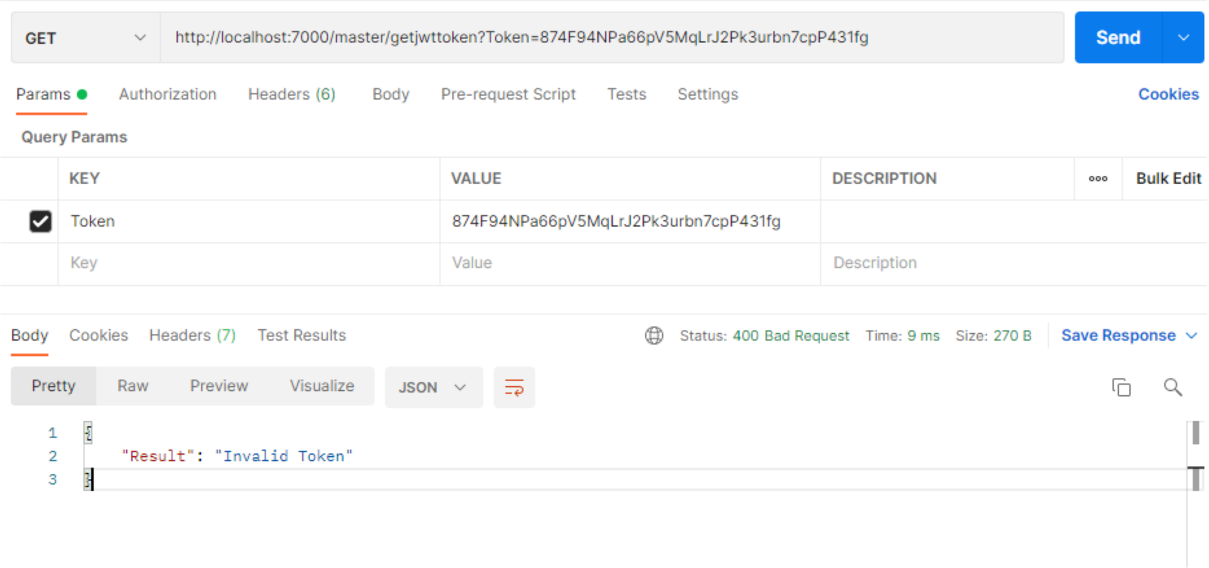
KEY	VALUE	DESCRIPTION
☑ Token	ZXYd9fYYDYCaoeE5CGm36FKvHxzYuibnas0x	
Key	Value	Description

Status: 200 OK   Time: 21 ms   Size: 345 B

```
{
    "Auth Token": "dm1lQWdSQVA2OG5QYUdGRTJZamVzRVNUaDdKbTNFQzZZNzVPOHk0cWdmblFoclQ4OWFxRzZxNGFMb1RvRHhsUlh3ZkQ="
}
```

If the Auth Token is generated after 8 minutes of user token generated it will be invalid so user need to re – login and generate new user token to generate Auth Token, the response will be as

```
{
    "Result": "Invalid Token"
}
```



GET http://localhost:7000/master/getjwttoken?Token=874F94NPa66pV5MqLrJ2Pk3urbn7cpP431fg

Query Params

KEY	VALUE	DESCRIPTION
☑ Token	874F94NPa66pV5MqLrJ2Pk3urbn7cpP431fg	
Key	Value	Description

Status: 400 Bad Request   Time: 9 ms   Size: 270 B

```
{
    "Result": "Invalid Token"
}
```

Create / Update The Items

Method : Post
url : http://localhost:7000/itemmaster/creteorupdate

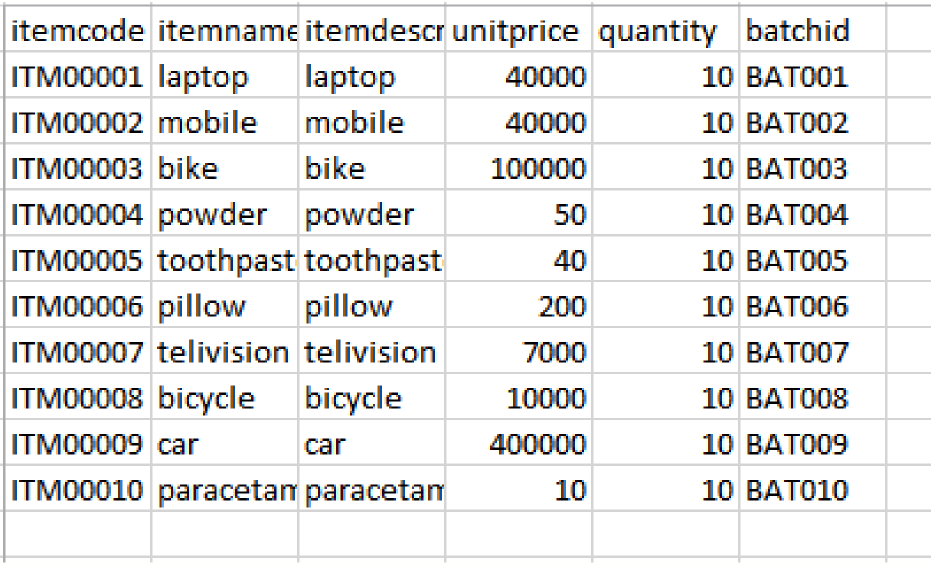
This Api will upload (creates / Update) the item datas (Stocks)

```
{
    "filename": "itemdata.xlsx",
    "username": "admin",
    "authtoken": "RDdhZG0wRzdpdE4xMDV0MjZURzZPYkFzams4eWdpcVFBNW9ENmQxbVhGSUJHRUMyO
TZtcXU4SjA1d3ZpMWZRWGc3WUw="
}
```

In the above JSON filename field is to be excel name that need to upload and field username and authtoken is once used Auth Token.

This input format is passed to body and execute the data

The excel field format will

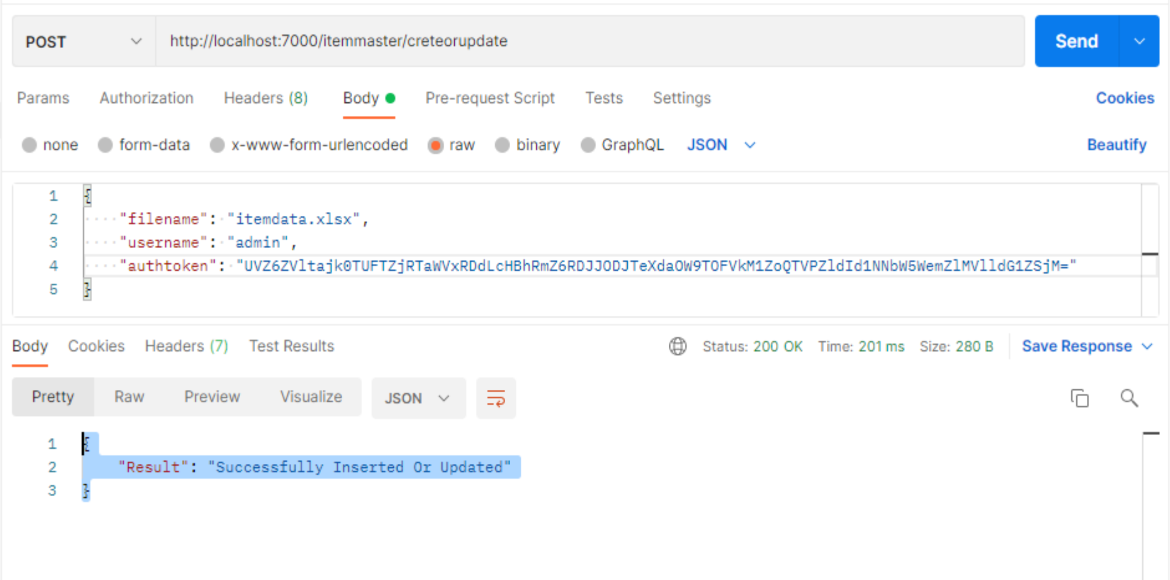


itemcode	itemname	itemdescr	unitprice	quantity	batchid
ITM00001	laptop	laptop	40000	10	BAT001
ITM00002	mobile	mobile	40000	10	BAT002
ITM00003	bike	bike	100000	10	BAT003
ITM00004	powder	powder	50	10	BAT004
ITM00005	toothpast	toothpast	40	10	BAT005
ITM00006	pillow	pillow	200	10	BAT006
ITM00007	telivision	telivision	7000	10	BAT007
ITM00008	bicycle	bicycle	10000	10	BAT008
ITM00009	car	car	400000	10	BAT009
ITM00010	paracetam	paracetam	10	10	BAT010

Note : this process allows only for admin username and generated Auth Token

Once the process completed the response as

```
{
    "Result": "Successfully Inserted or Updated"
}
```



http://localhost:7000/itemmaster/creteorupdate

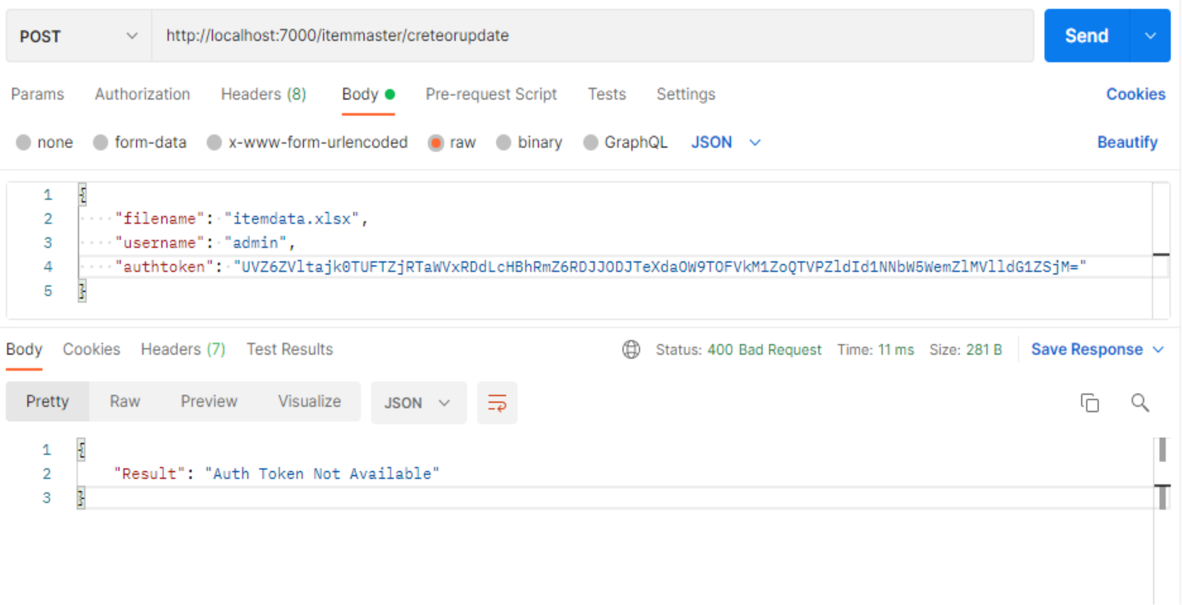
```
{
    "filename": "itemdata.xlsx",
    "username": "admin",
    "authtoken": "UVZ6ZVltajk0TUFTZjRTaWVxRDdLcHBhRmZ6RDJJODJTeXdaOW9TOFVkM1ZoQTVPZldId1NNbW5WemZlMVlldG1ZSjM="
}
```

Status: 200 OK Time: 201 ms Size: 280 B

```
{
    "Result": "Successfully Inserted Or Updated"
}
```

If the process is done again with same Auth Token it will response as not available because the Auth token is Valid for once the process

```
{
    "Result": "Auth Token Not Available"
}
```



http://localhost:7000/itemmaster/creteorupdate

```
{
    "filename": "itemdata.xlsx",
    "username": "admin",
    "authtoken": "UVZ6ZVltajk0TUFTZjRTaWVxRDdLcHBhRmZ6RDJJODJTeXdaOW9TOFVkM1ZoQTVPZldId1NNbW5WemZlMVlldG1ZSjM="
}
```

Status: 400 Bad Request Time: 11 ms Size: 281 B

```
{
    "Result": "Auth Token Not Available"
}
```

Create / Update / Cancel / Approve The Orders

Method : Post
url : http://localhost:7000/itemmaster/creteorupdate

This Api use to creates / Update / Cancel / Approve the order

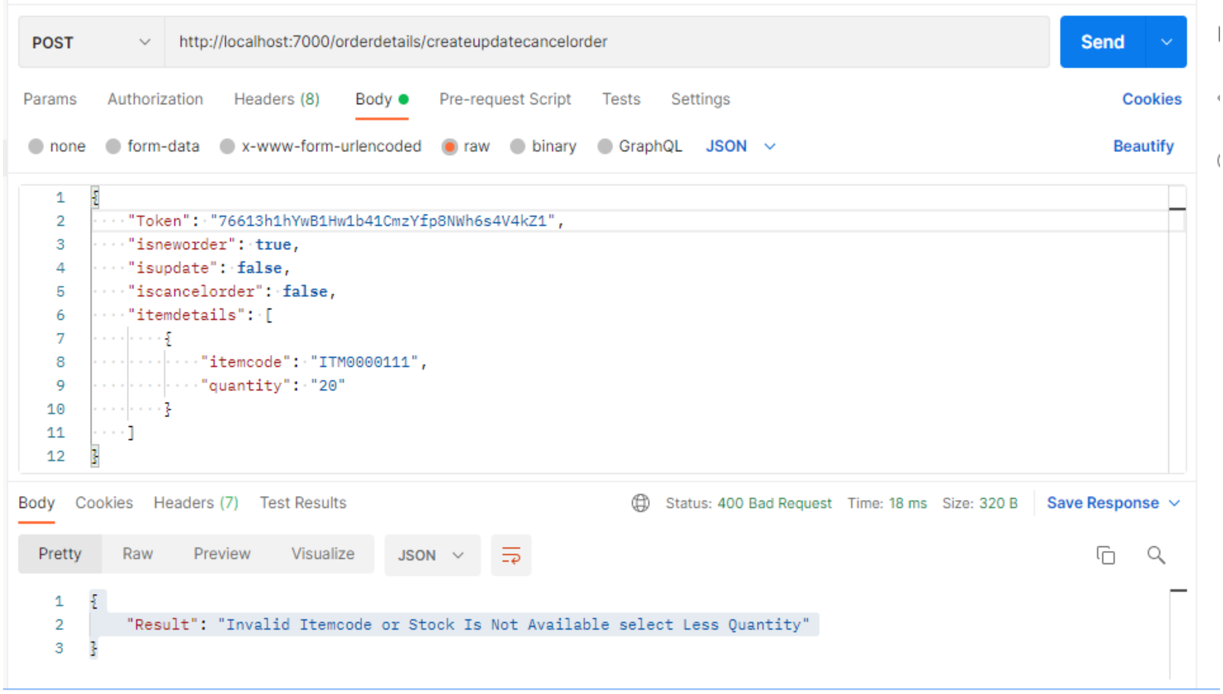
1. Creating Order:

```
{
    "Token": "v6149k4S1nu8ksFdcr219F1BJbXItmePnFQy",
    "isneworder": true,
    "isupdate": false,
    "iscancelorder": false,
    "itemdetails": [
        {
            "itemcode": "ITM0000111",
            "quantity": "20"
        }
    ]
}
```

The above JSON is passed in body to execute to create a orders, for creating orders isneworder : true (default) and isupdate and iscancelorder should be default false, and Token (user Token) field is mandatory, add the itemcode and quantity to place order in a object and execute the process.

If the itemcode is invalid or out of stock the response will be

```
{
    "Result": "Invalid Itemcode or Stock Is Not Available select Less Quantity"
}
```



POST http://localhost:7000/orderdetails/createupdatecancelorder

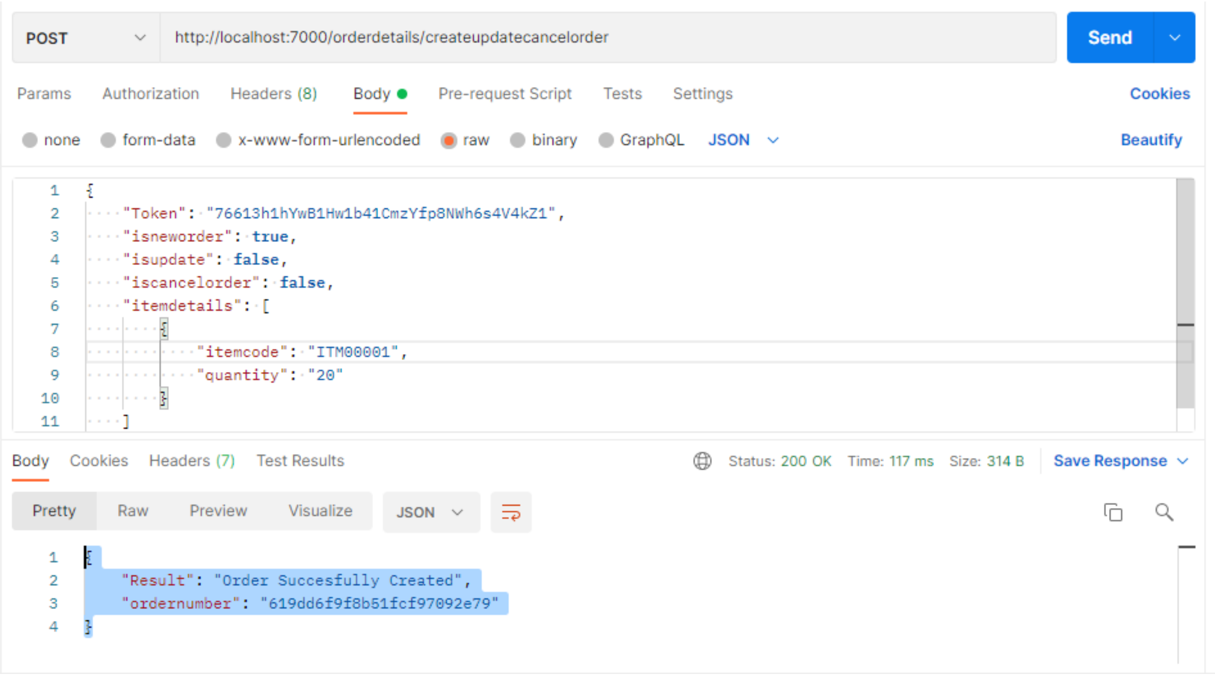
```
{
    "Token": "76613h1hYwB1Hw1b41CmzYfp8NWh6s4V4kZ1",
    "isneworder": true,
    "isupdate": false,
    "iscancelorder": false,
    "itemdetails": [
        {
            "itemcode": "ITM0000111",
            "quantity": "20"
        }
    ]
}
```

Status: 400 Bad Request Time: 18 ms Size: 320 B

```
{
    "Result": "Invalid Itemcode or Stock Is Not Available select Less Quantity"
}
```

If the item is valid and quantity is available the order will placed and will generate the order number

```
{
    "Result": "Order Succesfully Created",
    "ordernumber": "619dd6f9f8b51fcf97092e79"
}
```



POST http://localhost:7000/orderdetails/createupdatecancelorder

```
{
    "Token": "76613h1hYwB1Hw1b41CmzYfp8NWh6s4V4kZ1",
    "isneworder": true,
    "isupdate": false,
    "iscancelorder": false,
    "itemdetails": [
        {
            "itemcode": "ITM00001",
            "quantity": "20"
        }
    ]
```

Status: 200 OK Time: 117 ms Size: 314 B

```
{
    "Result": "Order Succesfully Created",
    "ordernumber": "619dd6f9f8b51fcf97092e79"
}
```

2. Updating Order :

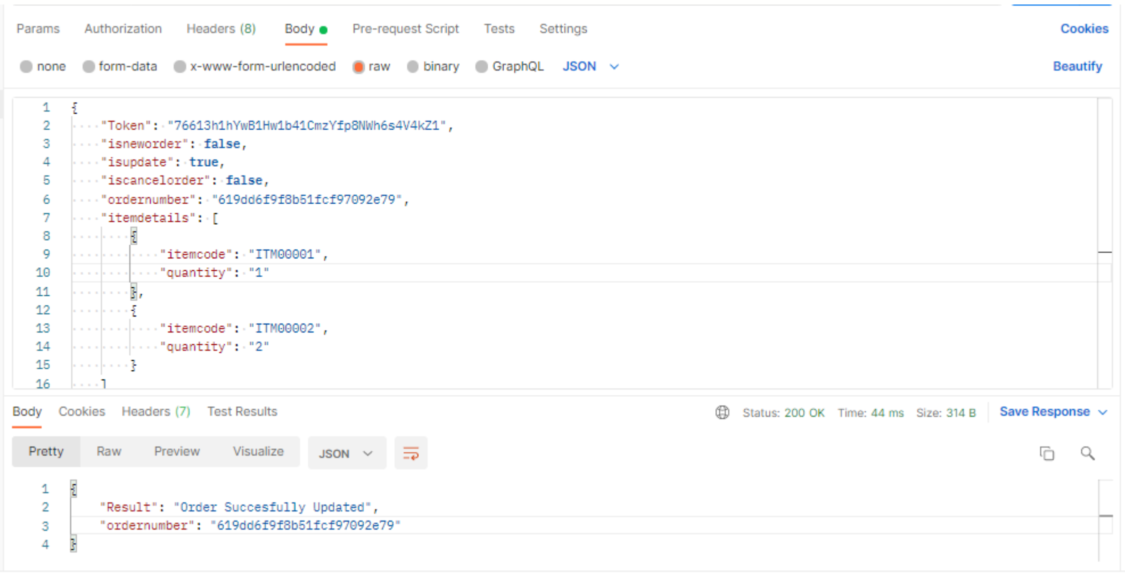
```
{
    "Token": "76613h1hYwB1Hw1b41CmzYfp8NWh6s4V4kZ1",
    "isneworder": false,
    "isupdate": true,
    "iscancelorder": false,
    "ordernumber": "619dd6f9f8b51fcf97092e79",
    "itemdetails": [
        {
            "itemcode": "ITM00001",
            "quantity": "1"
        },
        {
            "itemcode": "ITM00002",
            "quantity": "2"
        }
    ]
}
```

The JSON is passed in Api body to update the orders, for updating orders isupdate : true (default) and isneworder and iscancelorder should be default false, and Token (user Token) field and order number is mandatory to update the orders.

User can adjust the quantity or place new item to update the order items, once executed the response will be.

Note : Once the order is cancelled or Approved user cannot update the order and if the user token expired user has to re login and add the token to process the steps.

```
{
    "Result": "Order Successfully Updated",
    "ordernumber": "619dd6f9f8b51fcf97092e79"
}
```



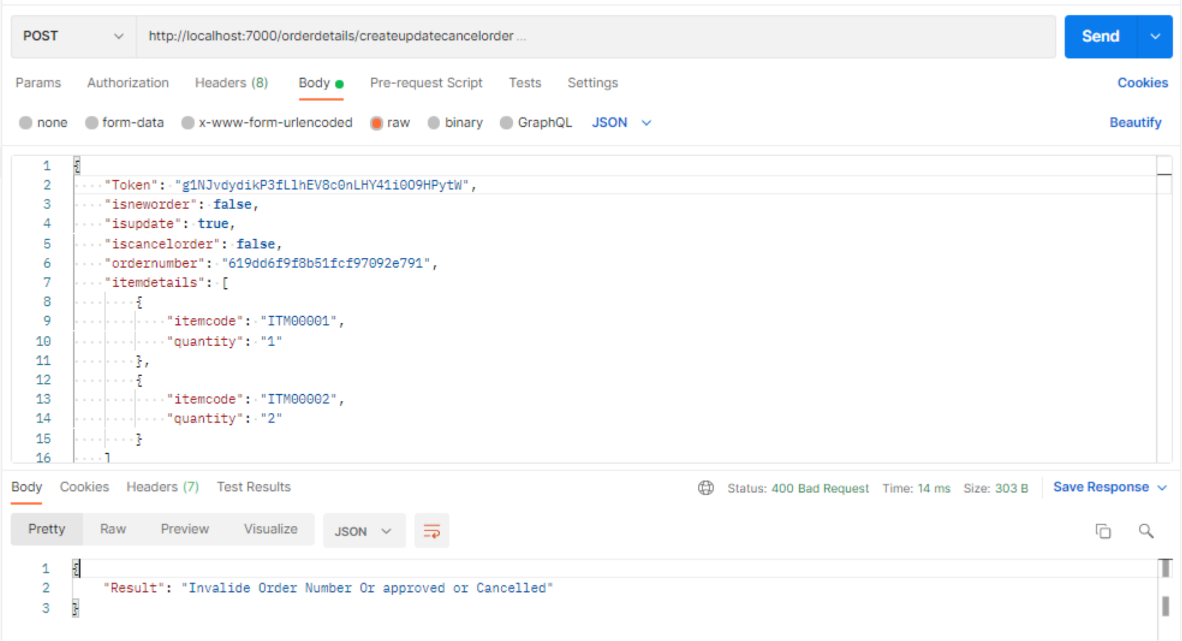
```
{
    "Token": "76613h1hYwB1Hw1b41CmzYfp8NWh6s4V4kZ1",
    "isneworder": false,
    "isupdate": true,
    "iscancelorder": false,
    "ordernumber": "619dd6f9f8b51fcf97092e79",
    "itemdetails": [
        {
            "itemcode": "ITM00001",
            "quantity": "1"
        },
        {
            "itemcode": "ITM00002",
            "quantity": "2"
        }
    ]
```

Status: 200 OK Time: 44 ms Size: 314 B

```
{
    "Result": "Order Succesfully Updated",
    "ordernumber": "619dd6f9f8b51fcf97092e79"
}
```

If the passed order number is invalid the response will be as

```
{
    "Result": "Invalid Order Number or approved or Cancelled"
}
```



http://localhost:7000/orderdetails/createupdatecancelorder ...

```
{
    "Token": "g1NJvdydikP3fLlhEV8c0nLHY41i0O9HPytW",
    "isneworder": false,
    "isupdate": true,
    "iscancelorder": false,
    "ordernumber": "619dd6f9f8b51fcf97092e791",
    "itemdetails": [
        {
            "itemcode": "ITM00001",
            "quantity": "1"
        },
        {
            "itemcode": "ITM00002",
            "quantity": "2"
        }
    ]
```

Status: 400 Bad Request Time: 14 ms Size: 303 B

```
{
    "Result": "Invalide Order Number Or approved or Cancelled"
}
```

3. Cancel Order

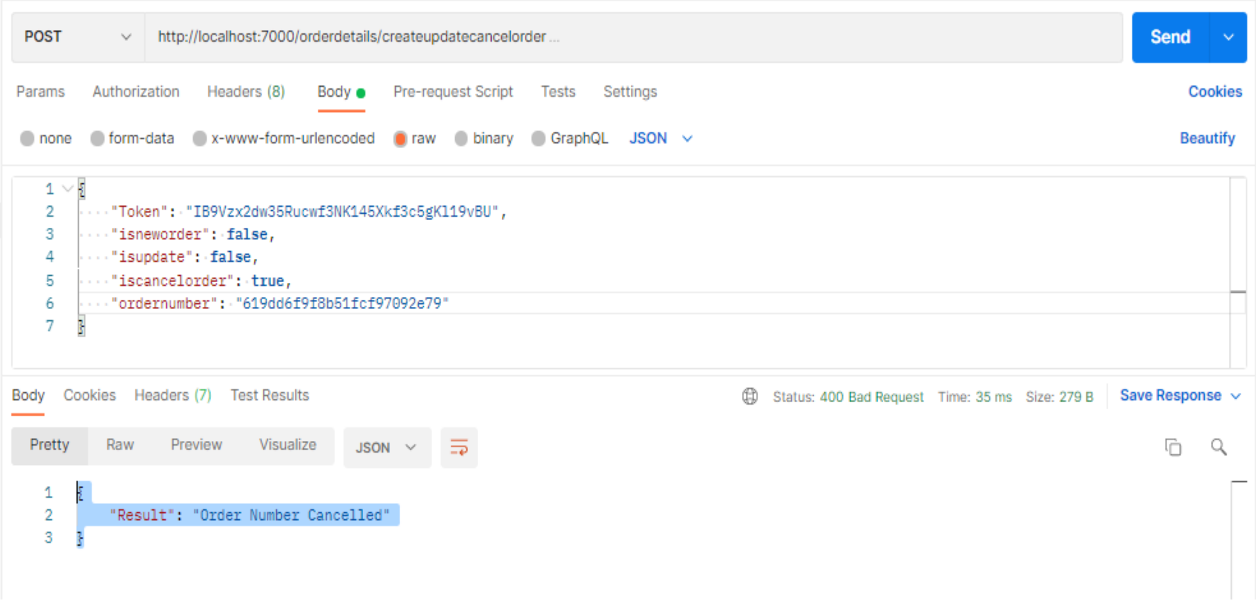
```
{
    "Token": "Q8H7c6ojcjoQmL19nxv8SZSktYLAVBbChgxM",
    "isneworder": false,
    "isupdate": false,
    "iscancelorder": true,
    "ordernumber": "61993f4c41dc4482dcfa8156"
}
```

The JSON is passed in Api body to cancel the orders, for cancel orders iscancelorder: true (default) and isneworder and isupdate should be default false, and Token (user Token) field and order number is mandatory to cancel the orders.

Note : once user cancelled the order they cannot approve or update the order and if the user token expired user has to re login and add the token to process the steps.

Once the order cancelled the response will be as

```
{
    "Result": "Order Number Cancelled"
}
```



http://localhost:7000/orderdetails/createupdatecancelorder ...

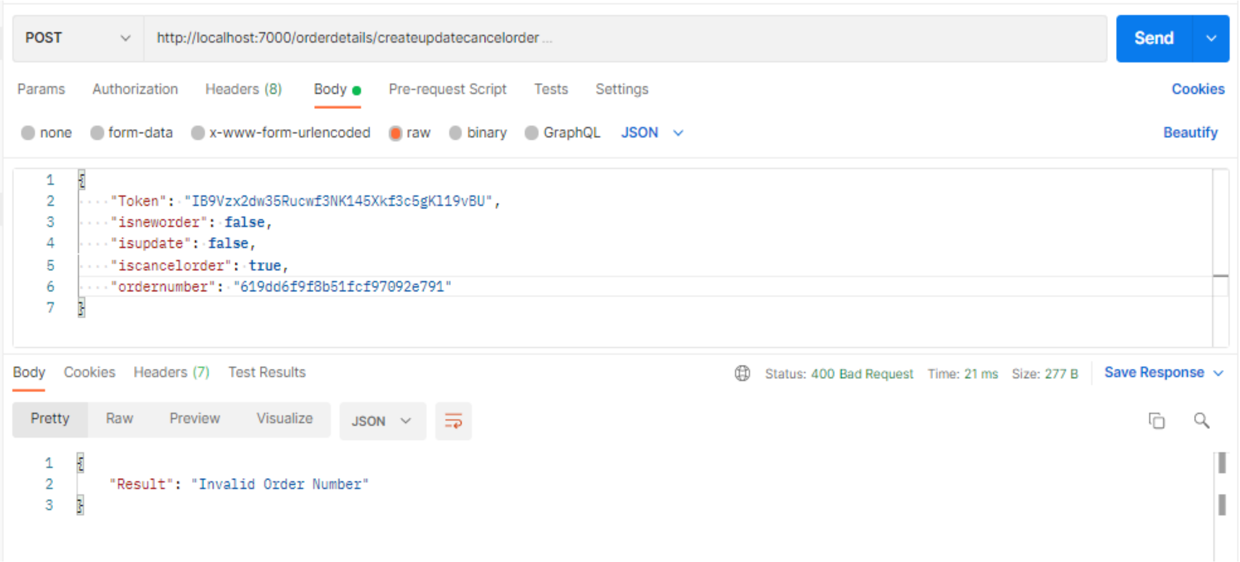
```
{
    "Token": "IB9Vzx2dw35Rucwf3NK145Xkf3c5gKl19vBU",
    "isneworder": false,
    "isupdate": false,
    "iscancelorder": true,
    "ordernumber": "619dd6f9f8b51fcf97092e79"
}
```

Status: 400 Bad Request Time: 35 ms Size: 279 B

```
{
    "Result": "Order Number Cancelled"
}
```

If the order number is Invalid it will response as

```
{
    "Result": "Invalid Order Number"
}
```



http://localhost:7000/orderdetails/createupdatecancelorder ...

```
{
    "Token": "IB9Vzx2dw35Rucwf3NK145Xkf3c5gKl19vBU",
    "isneworder": false,
    "isupdate": false,
    "iscancelorder": true,
    "ordernumber": "619dd6f9f8b51fcf97092e791"
}
```

Status: 400 Bad Request Time: 21 ms Size: 277 B

```
{
    "Result": "Invalid Order Number"
}
```

4. Approve Order

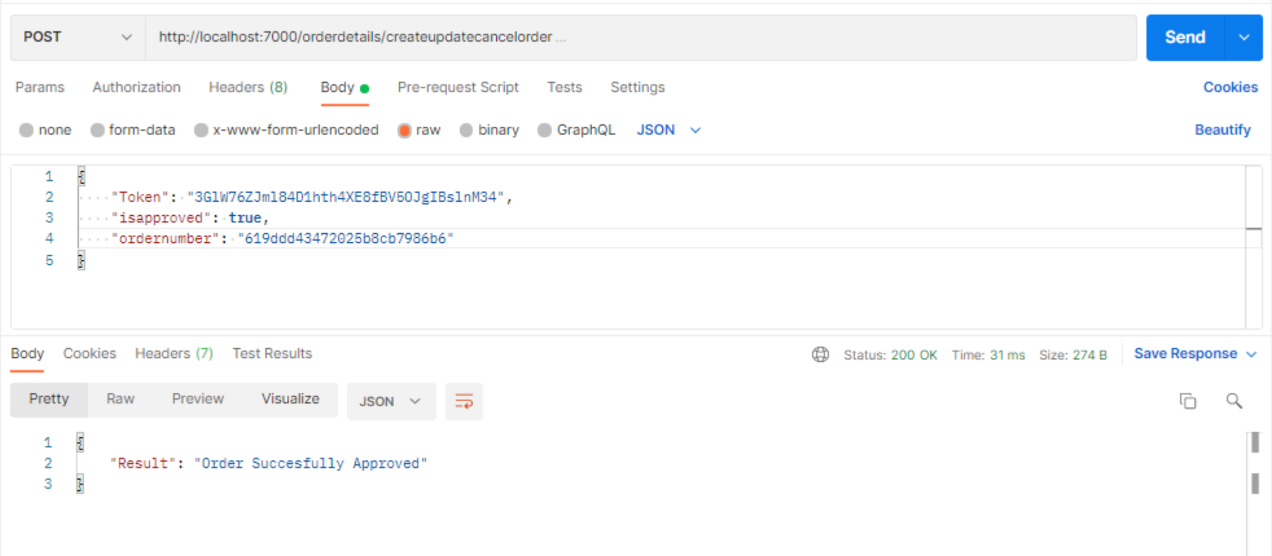
```
{
    "Token": "Q8H7c6ojcjoQmL19nxv8SZSktYLAVBbChgxM",
    "isapproved": true,
    "ordernumber": "6199e5eddcbdd4de27f1d313"
}
```

The JSON is passed in Api body to Approve the orders, for Approve orders isapproved: true (default) and Token (user Token) field and order number is mandatory to approve the orders.

Note : Once the order approved user cannot cancel or update the order and stock adjust from total from available quantity and if the user token expired user has to re login and add the token to process the steps.

Once this process executed the response will be

```
{
    "Result": "Order Successfully Approved"
}
```



http://localhost:7000/orderdetails/createupdatecancelorder ...

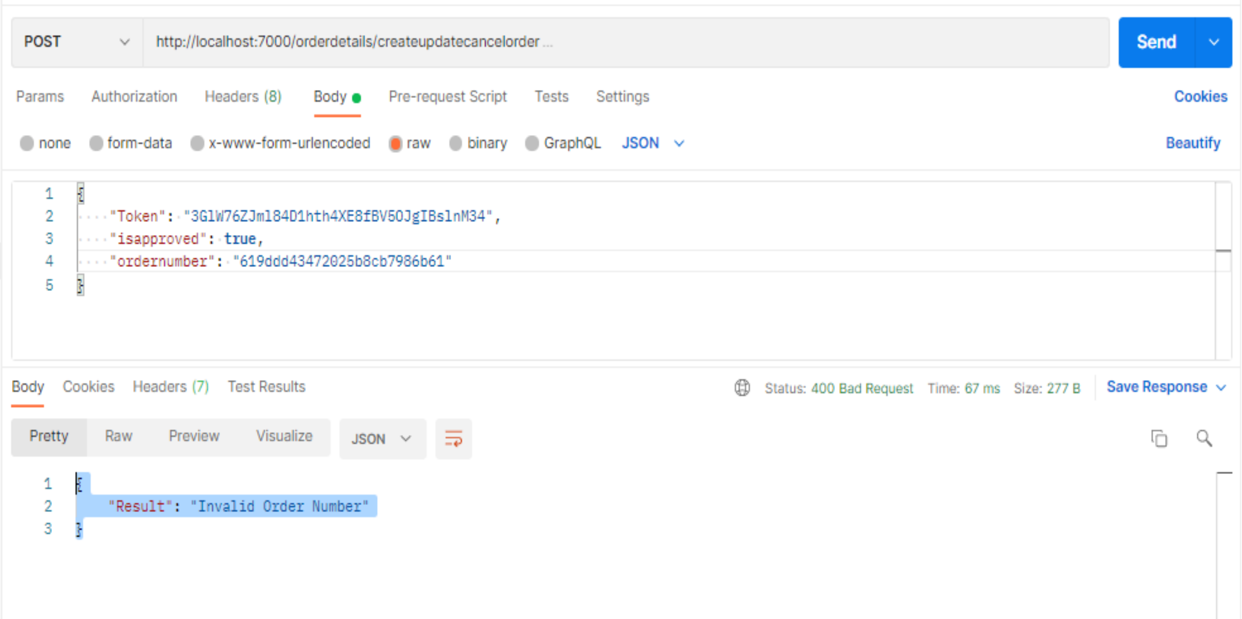
```
{
    "Token": "3GlW76ZJml84D1hth4XE8fBV50JgIBslnM34",
    "isapproved": true,
    "ordernumber": "619ddd43472025b8cb7986b6"
}
```

Status: 200 OK Time: 31 ms Size: 274 B

```
{
    "Result": "Order Succesfully Approved"
}
```

If the passed order number is invalid the response will be as

```
{
    "Result": "Invalid Order Number"
}
```



http://localhost:7000/orderdetails/createupdatecancelorder ...

```
{
    "Token": "3G1W76ZJm184D1hth4XE8fBV50JgIBslnM34",
    "isapproved": true,
    "ordernumber": "619ddd43472025b8cb7986b61"
}
```

Status: 400 Bad Request Time: 67 ms Size: 277 B

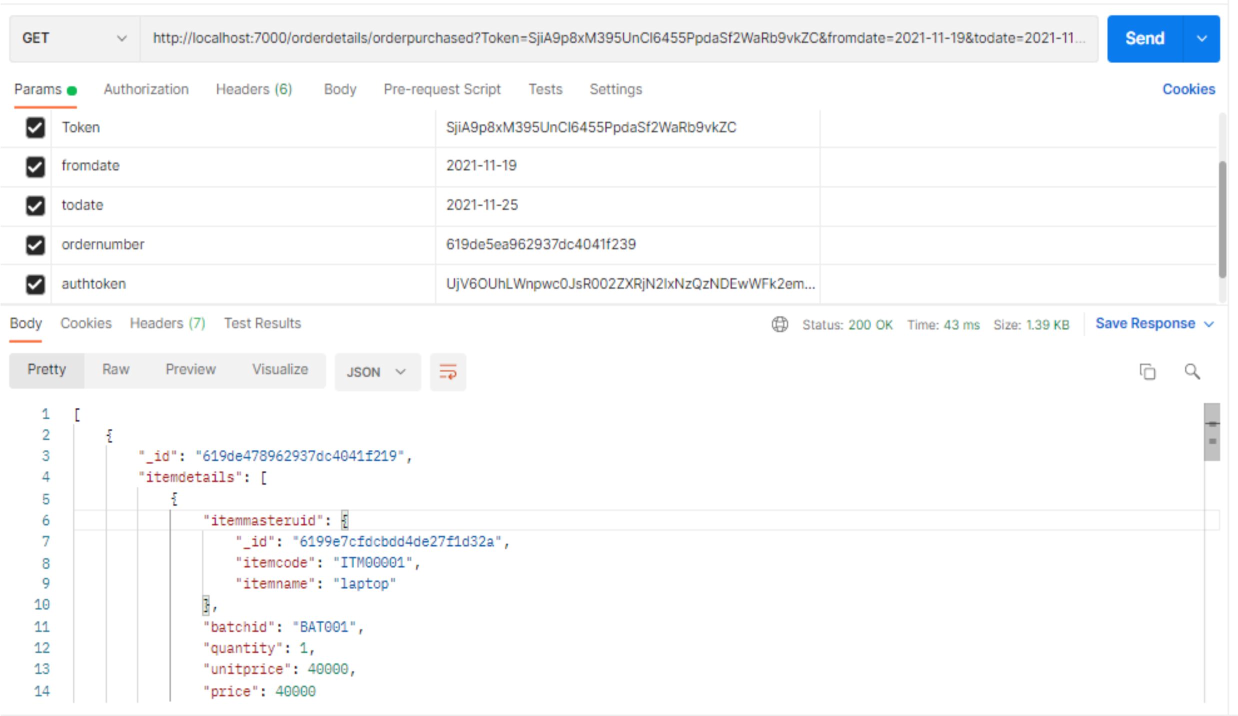
```
{
    "Result": "Invalid Order Number"
}
```

Viewing Order Purchased User Wise

Method : Get
url : http://localhost:7000/orderdetails/orderpurchased

This Api fetches the data based on user, User token and Auth Token is mandatory to fetch the data, “KEY” as Token and “VALUE” is user token and “KEY” as authtoken and “VALUE” is Auth Token and execute the data and it display orders based on user, order number, from date, to date field is Optional for fetching the specific data.

```
[
    {
        "_id": "619de5ea962937dc4041f238",
        "itemdetails": [
            {
                "itemmasteruid": {
                    "_id": "6199e7cfdcbdd4de27f1d32a",
                    "itemcode": "ITM00001",
                    "itemname": "laptop"
                },
                "batchid": "BAT001",
                "quantity": 2,
                "unitprice": 40000,
                "price": 80000
            }
        ],
        "ordercode": "619de5ea962937dc4041f239",
        "totalprice": 80000,
        "iscancelled": false,
        "isapproved": true,
        "modifiedby": {
            "_id": "619de2ec962937dc4041f1d1",
            "name": "admin"
        },
        "modifiedat": "2021-11-24T07:12:54.035Z",
        "createdby": {
            "_id": "619de2ec962937dc4041f1d1",
            "name": "admin"
        },
        "createdat": "2021-11-24T07:12:42.476Z"
    }
]
```



http://localhost:7000/orderdetails/orderpurchased?Token=SjiA9p8xM395UnCl6455PpdaSf2WaRb9vkZC&fromdate=2021-11-19&todate=2021-11...

☑	Token	SjiA9p8xM395UnCl6455PpdaSf2WaRb9vkZC
☑	fromdate	2021-11-19
☑	todate	2021-11-25
☑	ordernumber	619de5ea962937dc4041f239
☑	authtoken	UjV6OUhLWnpwc0JsR002ZXRjN2lxNzQzNDEwWFk2em...

Status: 200 OK Time: 43 ms Size: 1.39 KB

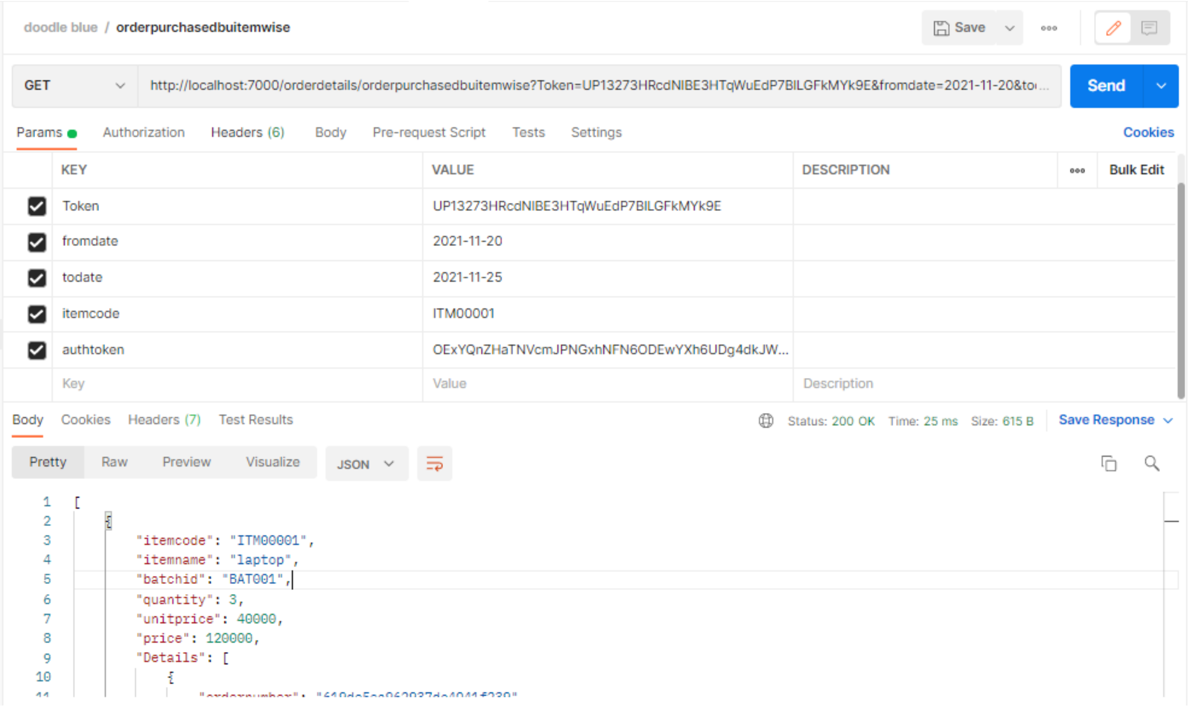
```
[
    {
        "_id": "619de478962937dc4041f219",
        "itemdetails": [
            {
                "itemmasteruid": {
                    "_id": "6199e7cfdcbdd4de27f1d32a",
                    "itemcode": "ITM00001",
                    "itemname": "laptop"
                },
                "batchid": "BAT001",
                "quantity": 1,
                "unitprice": 40000,
                "price": 40000
```

Viewing Orders Based on Items

Method : Get
url : http://localhost:7000/orderdetails/orderpurchasedbuitemwise

This Api fetches the data based on item, User token, Auth Token and item code is mandatory to fetch the data, “KEY” as Token and “VALUE” is user token and “KEY” as authtoken and “VALUE” is Auth Token and “KEY” as item code “VALUE” is code value and execute the data and it display order item based on items , from date, to date field is Optional for fetching the specific data. This api show data group based on item wise order.

```
[
  {
    "itemcode": "ITM00001",
    "itemname": "laptop",
    "batchid": "BAT001",
    "quantity": 3,
    "unitprice": 40000,
    "price": 120000,
    "Details": [
      {
        "ordernumber": "619de5ea962937dc4041f239",
        "createdbyname": "admin",
        "cancelled": "No",
        "approved": "Yes",
        "quantity": 2,
        "price": 80000
      },
      {
        "ordernumber": "619de478962937dc4041f21a",
        "createdbyname": "admin",
        "cancelled": "Yes",
        "approved": "No",
        "quantity": 1,
        "price": 40000
      }
    ]
  }
]
```



http://localhost:7000/orderdetails/orderpurchasedbuitemwise?Token=UP13273HRcdNIBE3HTqWuEdP7BILGFkMYk9E&fromdate=2021-11-20&to...

	KEY	VALUE	DESCRIPTION
☑	Token	UP13273HRcdNIBE3HTqWuEdP7BILGFkMYk9E	
☑	fromdate	2021-11-20	
☑	todate	2021-11-25	
☑	itemcode	ITM00001	
☑	authtoken	OExYQnZHaTNVcmJPNGxhNFN6ODEwYXh6UDg4dkJW...	
	Key	Value	Description

Status: 200 OK Time: 25 ms Size: 615 B

```
[
    {
        "itemcode": "ITM00001",
        "itemname": "laptop",
        "batchid": "BAT001",
        "quantity": 3,
        "unitprice": 40000,
        "price": 120000,
        "Details": [
            {
                "ordernumber": "619de5ea962937dc4041f239"
```