

a) POST API to Create an Order

The screenshot shows the Postman interface for a POST request to `localhost:8080/orders/placeOrder`. The request body is a JSON object with an `orderId` of 100 and an `itemList` containing two items: a smartphone and a phone cover. The response status is 201 Created, and the message is "Order placed Successfully".

Request Details:

- Method: POST
- URL: `localhost:8080/orders/placeOrder`
- Body Type: raw (JSON)
- Body Content:

```
1 {
2   "orderId":100,
3   "itemList":[
4     {
5       "itemId":1000,
6       "itemName":"real me 7 smartphone",
7       "itemQuantity":1
8     },
9     {
10      "itemId":1001,
11      "itemName":"phone cover",
12      "itemQuantity":2
13    }
14  ]
15 }
```

Response Details:

- Status: 201 Created
- Time: 601 ms
- Size: 194 B
- Message: 1 Order placed Successfully

b) POST API to Add items in an Order

The screenshot shows the Postman interface for a POST request to `localhost:8080/orders/item/add?orderId=100`. The request body is a JSON object for a new item: wireless earphones. The interface also shows other requests in the background.

Request Details:

- Method: POST
- URL: `localhost:8080/orders/item/add?orderId=100`
- Body Type: raw (JSON)
- Body Content:

```
1 {
2   "itemId":1002,
3   "itemName":"real me wireless earphones",
4   "itemQuantity":2
5 }
```

Background Requests:

- POST add item to order
- GET get all orders
- POST place new order

```
1 {
2   "orderId": 100,
3   "itemList": [
4     {
5       "itemId": 1000,
6       "itemName": "real me 7 smartphone",
7       "itemQuantity": 1
8     },
9     {
10      "itemId": 1001,
11      "itemName": "phone cover",
12      "itemQuantity": 2
13    },
14    {
15      ...
16    }
17  ]
18 }
```

Body Cookies (2) Headers (5) Test Results Status: 201 Created Time: 13 ms Size: 395 B Save Response

```
7   "itemQuantity": 1
8 },
9 {
10  "itemId": 1001,
11  "itemName": "phone cover",
12  "itemQuantity": 2
13 },
14 {
15  "itemId": 1002,
16  "itemName": "real me wireless earphones",
17  "itemQuantity": 2
18 }
19 ]
20 }
```

Activate Windows
Go to Settings to activate Windows.

c) PUT API to Update Item details

The screenshot shows a REST client interface with a tab for 'PUT update item det...'. The request is configured as follows:

- Method:** PUT
- URL:** localhost:8080/orders/item/update?orderId=100&itemId=1000
- Body:** JSON, containing:

```
1 {
2   "itemId": 1000,
3   "itemName": "real me 7 smartphone",
4   "itemQuantity": 2
5 }
```

The response is displayed below the request, showing a status of 200 OK, time of 11 ms, and size of 230 B. The response body is JSON, containing:

```
1 {
2   "itemId": 1000,
3   "itemName": "real me 7 smartphone",
4   "itemQuantity": 2
5 }
```

At the bottom right, there is a watermark: "Activate Windows Go to Settings to activate Windows."

d) GET API to Get all the Item details in an Order

The screenshot shows a REST client interface with a tab for 'GET get all item...'. The request is configured as follows:

- Method:** GET
- URL:** localhost:8080/orders/items?orderId=100

The response is displayed below the request, showing a status of 200 OK, time of 38 ms, and size of 363 B. The response body is JSON, containing:

| KEY | VALUE | DESCRIPTION |
|---|-------|-------------|
| <input checked="" type="checkbox"/> orderId | 100 | |
| Key | Value | Description |

GET

localhost:8080/orders/items?orderId=100

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Body

Cookies (2)

Headers (5)

Test Results



Status: 200 OK Time: 38

Pretty

Raw

Preview

Visualize

JSON



```
1  [
2    {
3      "itemId": 1000,
4      "itemName": "real me 7 smartphone",
5      "itemQuantity": 2
6    },
7    {
8      "itemId": 1001,
9      "itemName": "phone cover",
10     "itemQuantity": 2
11   },
12   {
13     "itemId": 1002,
14     "itemName": "real me wireless earphones",
15     "itemQuantity": 2
16   }
17 ]
```

Activat
Go to Se

e) DELETE API to Delete an Item in an Order

GET get all i...

POST place ...

POST add it...

GET get all ...

PUT updat...

DEL delete ... X

+

...

Task APP(dev) [dropdown] [eye icon] [settings icon]

▶ delete item in a order [Examples 0] [BUILD] [edit icon] [comment icon]

DELETE [dropdown]

localhost:8080/orders/item/delete?orderId=100&itemId=1001

Send [dropdown]

Save [dropdown]

Params ●

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Code

Query Params

| | KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-------------------------------------|---------|-------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | orderId | 100 | | | |
| <input checked="" type="checkbox"/> | itemId | 1001 | | | |
| | Key | Value | Description | | |

Body

Cookies (2)

Headers (5)

Test Results

🌐 Status: 200 OK Time: 7 ms Size: 200 B Save Response [dropdown]

Pretty

Raw

Preview

Visualize

Text [dropdown]

[icon]

1 Item Deleted from order successfully