

Table of Content

[Check Status Code 404](#)

[Use Snippets](#)

[Check Status Code 201 - Submit an Order](#)

[Check Status Code 200 - Get an Order](#)

[Check Status Code 204 - Delete an Order](#)

[Submit an Order](#)

Check Status Code 404

Add script to the request, change Body to Test Result

The screenshot displays the Postman interface for a request collection named "GET Order". The left sidebar shows a list of requests, with "GET Order" selected and highlighted by a red arrow. The main panel shows the request details for a GET method to the URL `{{BaseURL}}/orders/:orderId`. The "Scripts" tab is active, showing a test script:

```
1 pm.test("Status code is 404", function () {  
2   | pm.response.to.have.status(404);  
3 });
```

A red arrow points from the "Scripts" tab to the test script. Another red arrow points from the "Body" tab to the response body. The "Body" tab shows a JSON response with a 404 status:

```
1 {  
2   "error": "No order with id cuYLS0ZIAEqMnB309Qr."  
3 }
```

The status bar at the bottom indicates "404 Not Found" with a response time of 698 ms and a size of 798 B. The "Save Response" button is visible.

HTTP

Request Collection / GET Order

Save

Share

GET

{{BaseURL}}/orders/:orderId

Send

Docs

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Pre-request

Post-response

```
1 pm.test("Status code is 404", function () {
2   |   pm.response.to.have.status(404);
3   | });
```

Packages

Snippets

Test Results

404 Not Found

698 ms

798 B

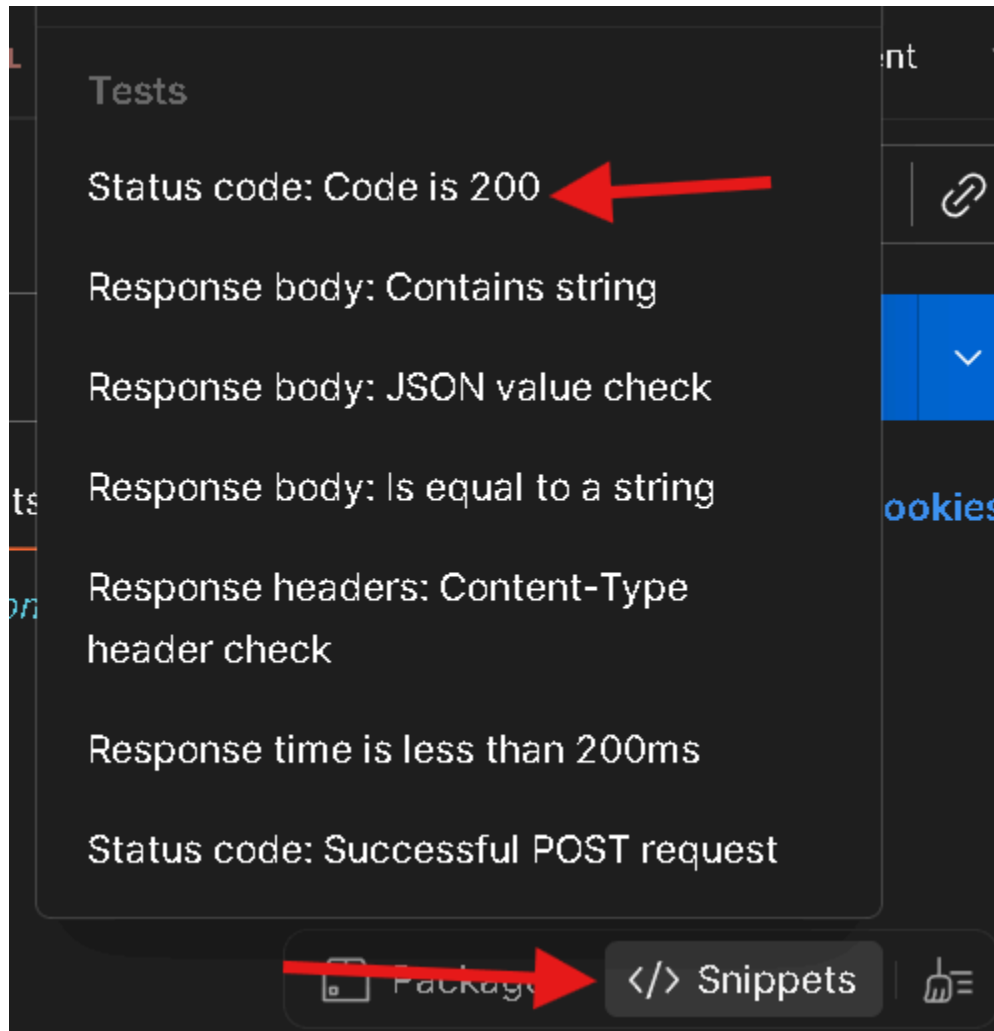
Save Response

Filter Results

PASSED

Status code is 404

Use Snippets



Check Status Code 201 - Submit an Order

Request Collection / POST Submit Order

POST `{{BaseURL}}/orders`

Scripts

```
1 pm.test("Status code is 201", function () {  
2   pm.response.to.have.status(201);  
3 });
```

Test Results

PASSED Status code is 201

Body

JSON

```
1 {  
2   "created": true,  
3   "orderId": "7rFD_-9o__Lo0mwzBxwsV"  
4 }
```

Add orderid to update

Request Collection / PATCH Update Order

PATCH `{{BaseURL}}/orders/:orderId`

Params

Key	Value	Description
orderId	7rFD_-9o__Lo0mwzBxwsV	Description

Body

204 No Content

Request Collection / PATCH Update Order

PATCH `{{BaseURL}} /orders/ :orderId` Send

Docs Params Authorization Headers (10) Body Scripts Settings Cookies

Pre-request

Post-response

```
1 pm.test("Status code is 204", function () {
2   |   pm.response.to.have.status(204);
3   | });
```

Packages </> Snippets

PATCH `{{BaseURL}} /orders/ :orderId` Send

Docs Params Authorization Headers (10) Body Scripts Settings Cookies

Pre-request

Post-response

```
1 pm.test("Status code is 204", function () {
2   |   pm.response.to.have.status(204);
3   | });
```

Packages </> Snippets

Test Results 204 No Content • 313 ms • 746 B • Save Response

Filter Results

PASSED Status code is 204

After updating an Order -> Get an Order
Add order ID

Request Collection

- GET GET Status
- GET GET List of Books
- GET GET List of Books by Query
- GET GET Book by ID
- POST POST Register API Client
- POST POST Submit Order
- PATCH PATCH Update Order
- GET GET Order

Request Collection / GET Order

GET `{{BaseURL}} /orders/ :orderId` Send

Docs Params Authorization Headers (8) Body Scripts Settings Cookies

Key	Value	Description
orderId	7rFD_-9c_Lo0mwzBxwsV	Description

Path Variables

Key	Value	Description	Bulk Edit
orderId	7rFD_-9c_Lo0mwzBxwsV	Description	

Check Status Code 200 - Get an Order

Workspace

New Import

Request Collection

- GET GET Status
- GET GET List of Books
- GET GET List of Books by Query
- GET GET Book by ID
- POST POST Register API Client
- POST POST Submit Order
- PATCH PATCH Update Order
- GET GET Order
- DEL DELETE Order

Request Collection / GET Order

GET {{BaseURL}}/orders/:orderId

Send

Docs Params Authorization Headers (8) Body Scripts Settings Cookies

Pre-request

Post-response

```
1 pm.test("Status code is 200", function () {  
2   pm.response.to.have.status(200);  
3 });
```

Test Results

200 OK • 708 ms • 852 B • Save Response

Filter Results

PASSED Status code is 200

Request Collection / GET Order

GET {{BaseURL}}/orders/:orderId

Send

Docs Params Authorization Headers (8) Body Scripts Settings Cookies

Pre-request

Post-response

```
1 pm.test("Status code is 200", function () {  
2   pm.response.to.have.status(200);  
3 });
```

Body

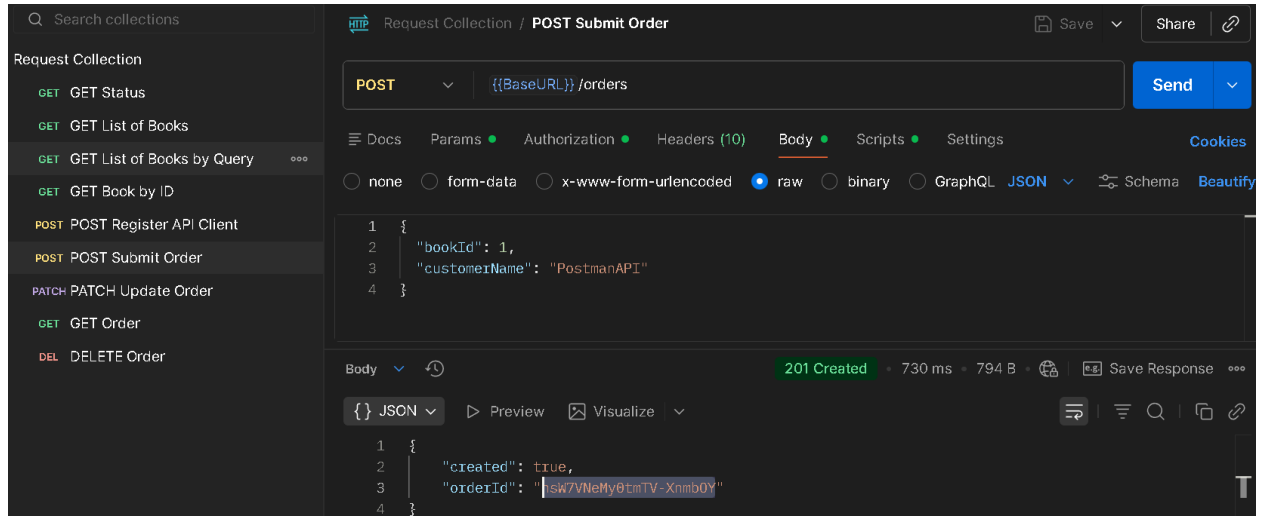
200 OK • 708 ms • 852 B • Save Response

{ } JSON Preview Visualize

```
1 {  
2   "id": "7rFD_-9o_Lo0mwzBxwsV",  
3   "bookId": 1,  
4   "customerName": "new PostmanAPI",  
5   "quantity": 1,  
6   "timestamp": 1768396468373  
7 }
```

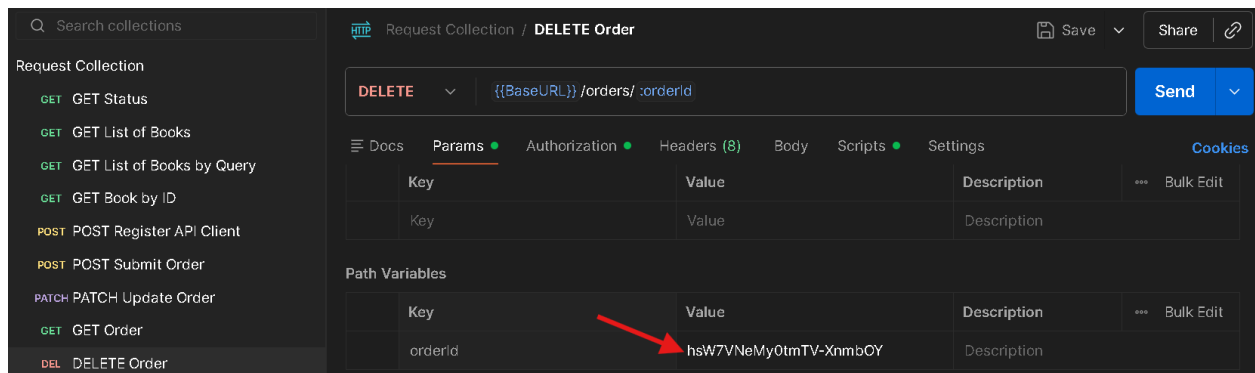
Check Status Code 204 - Delete an Order

Submit an Order

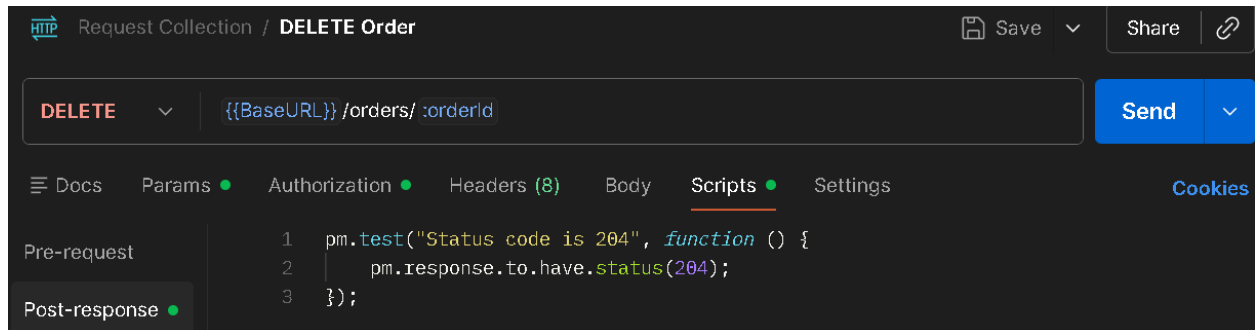


The screenshot shows the Postman interface for a 'POST Submit Order' request. The URL is `{{BaseURL}}/orders`. The request body is a JSON object: `{ "bookId": 1, "customerName": "PostmanAPI" }`. The response status is '201 Created' with a response time of 730 ms and a size of 794 B. The response body is a JSON object: `{ "created": true, "orderId": "hsW7VNeMy0tmTV-XnmbOY" }`.

Get orderId and apply in DELETE



The screenshot shows the Postman interface for a 'DELETE Order' request. The URL is `{{BaseURL}}/orders/:orderId`. The 'Path Variables' section shows the variable `orderId` with the value `hsW7VNeMy0tmTV-XnmbOY`, which is highlighted with a red arrow.



The screenshot shows the 'Scripts' tab in the Postman interface for the 'DELETE Order' request. The script is as follows:

```
1 pm.test("Status code is 204", function () {
2   pm.response.to.have.status(204);
3 });
```

HTTP Request Collection / DELETE Order

Save Share

DELETE {{BaseURL}}/orders/:orderId Send

Docs Params Authorization Headers (8) Body Scripts Settings Cookies

Pre-request

Post-response

```
1 pm.test("Status code is 204", function () {
2   pm.response.to.have.status(204);
3 });
```

Test Results 204 No Content 326 ms 746 B Save Response

Filter Results

PASSED Status code is 204