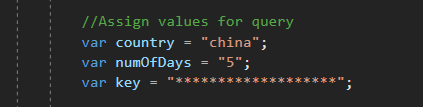
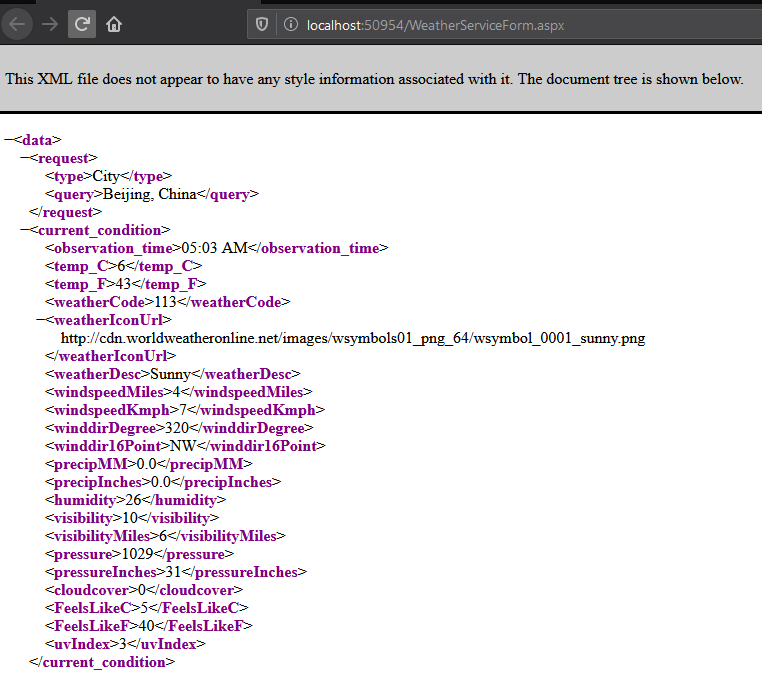
CSC Assignment 1 Documentation

Task 1

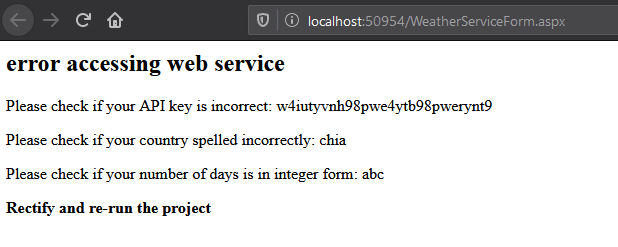
Assign the country, number of days and insert your key from <https://www.worldweatheronline.com/developer/> .



Debug run the application and you will see the following page on load.



In the case of error, the web service will ask u to verify your API key as well as your condition inputs.

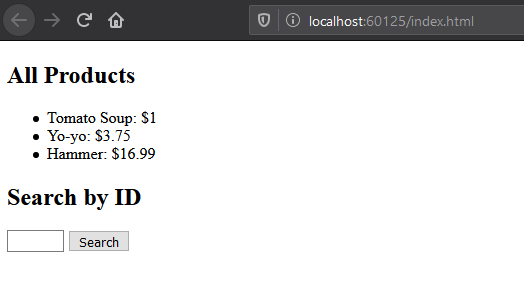


Task 2

Browser

Debug run the application and you will see the following page on load.

(There is a simulated loading, loading circle will appear)



Enter a number between 1 to 3 (ID for the products).



Postman

Localhost:

http://localhost:60125/

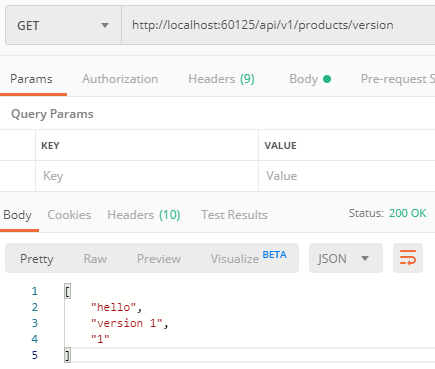
APIs:

|  |  |  |
| --- | --- | --- |
| Version | Type | API Route |
| V1 | GET | api/v1/products/version |
| V1 | GET | api/v1/products/message?name1=<**name**>&name2=<**name**>&name3=<**name**> |
| V1 | GET | api/v1/products |
| V1 | GET | api/v1/products/{**id**} |
| V2 | GET | api/v2/products |
| V2 | GET | api/v2/products/{**id**} |
| V2 | GET | api/v2/products?Category=<**category**> |
| V2 | POST | api/v2/products |
| V2 | PUT | api/v2/products/{**id**} |
| V2 | DELETE | api/v2/products/{**id**} |
| V3 | GET | api/v3/products |
| V3 | GET | api/v3/products/{**id**} |
| V3 | GET | api/v3/products?Category=<**category**> |
| V3 | POST | api/v3/products |
| V3 | PUT | api/v3/products/{**id**} |
| V3 | DELETE | api/v3/products/{**id**} |

GET: api/v1/products/version

Link: <http://localhost:25157/api/v1/products/version>

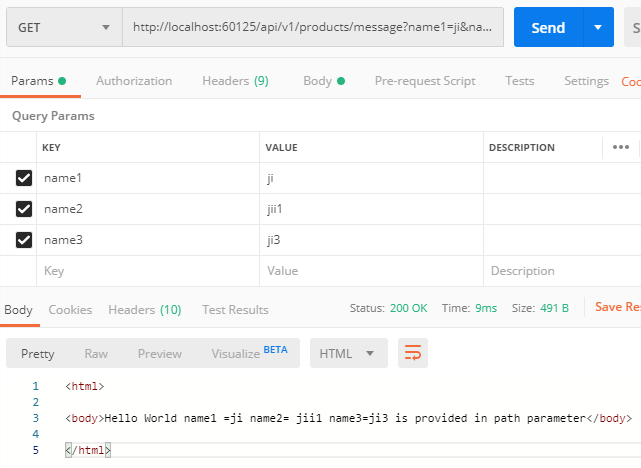
Get API version.



GET: api/v1/products/message?name1=<**name**>&name2=<**name**>&name3=<**name**>

Link: <http://localhost:60125/api/v1/products/message/name1=ji&name2=jii1&name3=ji3>

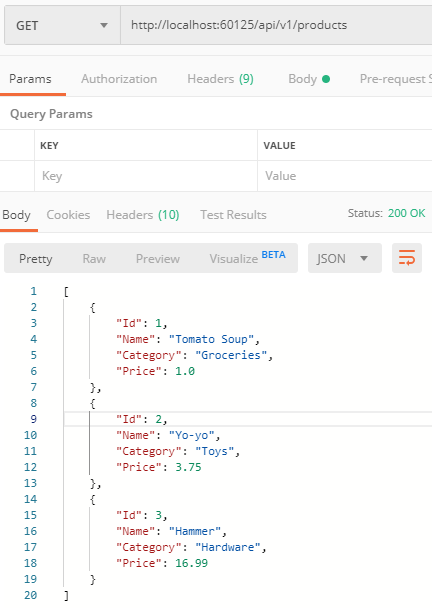
Get method with user defined parameters.



GET: api/v1/products

Link: <http://localhost:60125/api/v1/products>

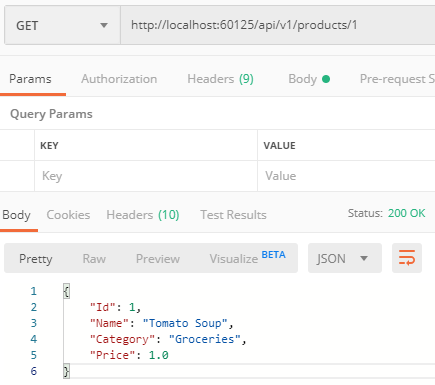
Get all products.



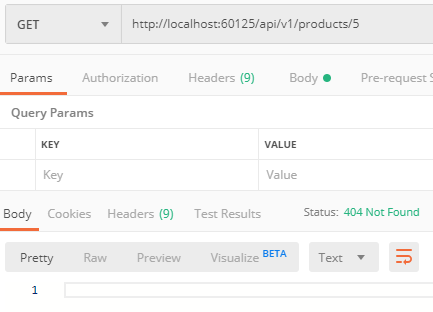
GET: api/v1/products/{**id**}

Link: <http://localhost:60125/api/v1/products/1>

Get product based on id. (Available Id between 1 to 3)



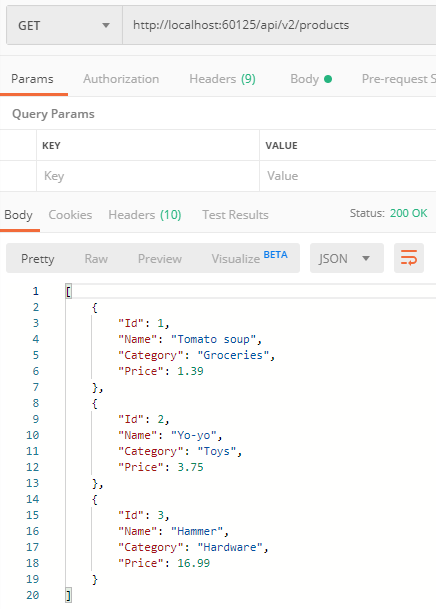
Providing invalid id will lead to 404 Not Found.



GET: api/v2/products

Link: <http://localhost:60125/api/v2/products>

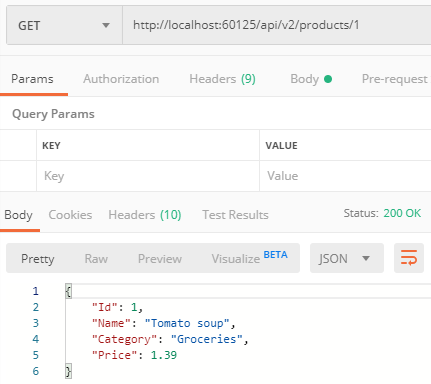
Get all products.



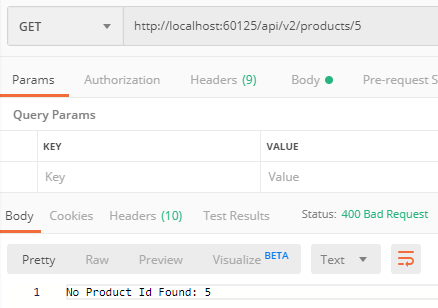
GET: api/v2/products/{**id**}

Link: <http://localhost:60125/api/V2/products>

Get product based on id. (Available Id between 1 to 3)



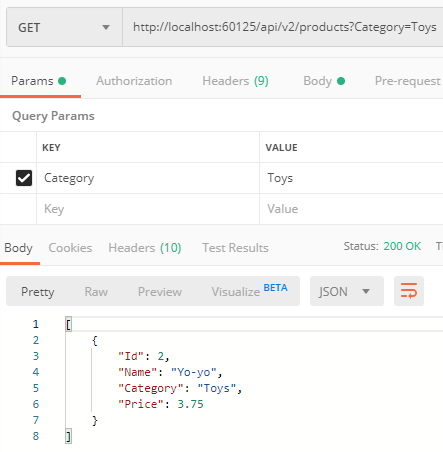
Providing invalid id will lead to 400 Bad Request



GET: api/v2/products?Category=<**category**>

Link: <http://localhost:60125/api/v2/products?Category=Toys>

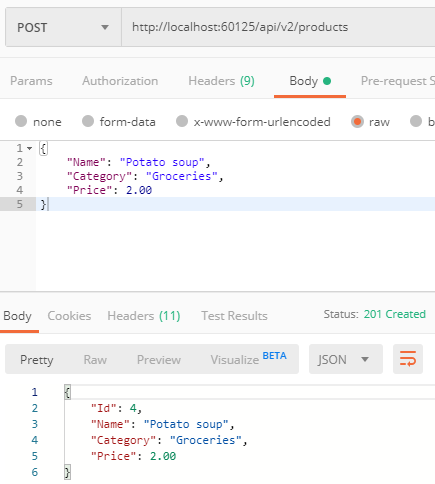
Get products by Category as parameter.



POST: api/v2/products

Link: <http://localhost:60125/api/v2/products>

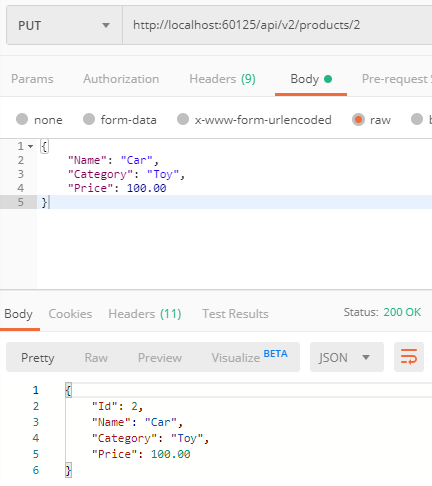
Create product using Post.



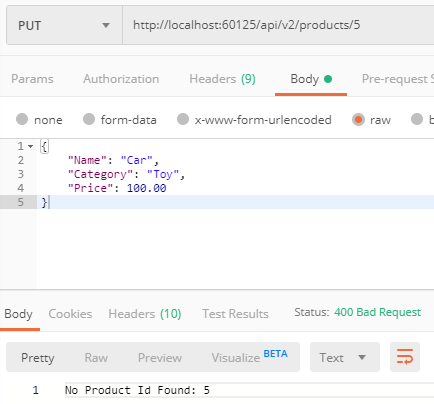
PUT: api/v2/products/{**id**}

Link: <http://localhost:60125/api/v2/products/2>

Change product details based on id using Put. (Available Id between 1 to 3)



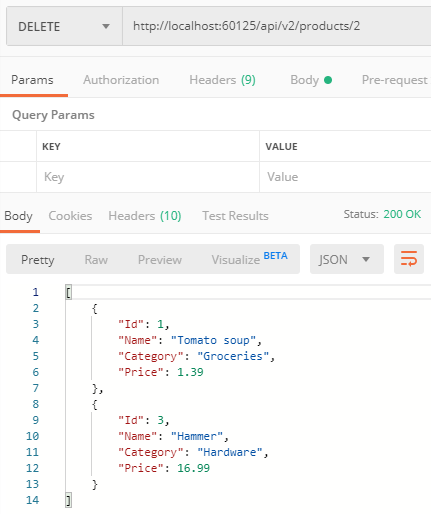
Providing invalid id will lead to 400 Bad Request



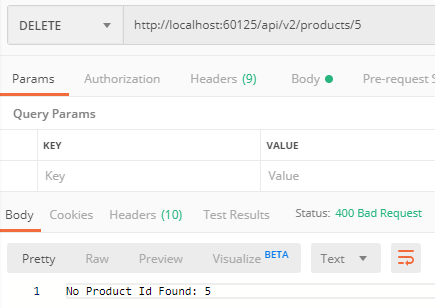
DELETE: api/v2/products/{**id**}

Link: <http://localhost:60125/api/v2/products/2>

Remove product based on id using Delete. (Available Id between 1 to 3)



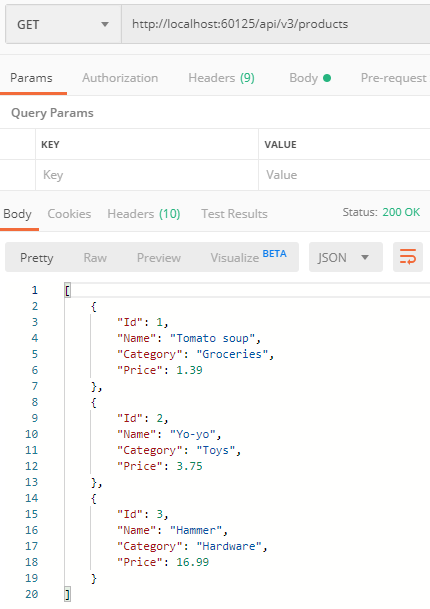
Providing invalid id will lead to 400 Bad Request



GET: api/v3/products

Link: <http://localhost:60125/api/v3/products>

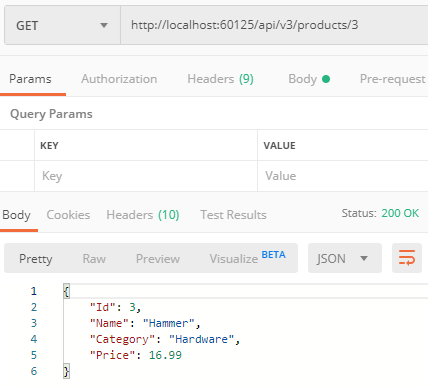
Get all products.



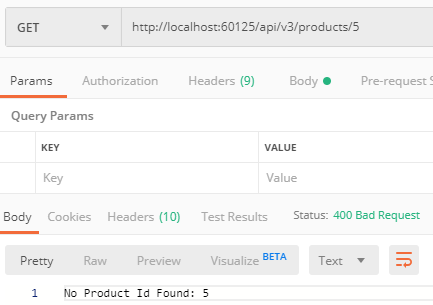
GET: api/v3/products/{**id**}

Link: <http://localhost:60125/api/v3/products/3>

Get product based on id. (Available Id between 1 to 3)



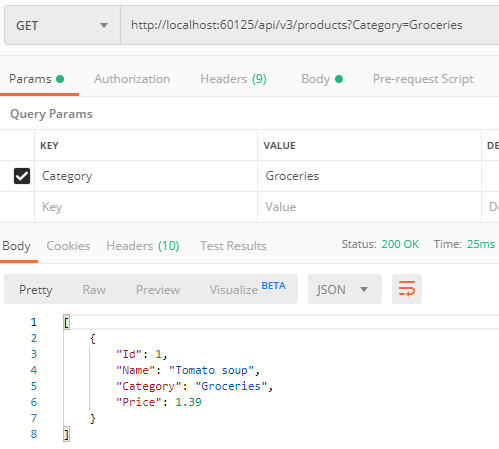
Providing invalid id will lead to 400 Bad Request



GET: api/v3/products?Category=<**category**>

Link: <http://localhost:60125/api/v3/products?Category=Groceries>

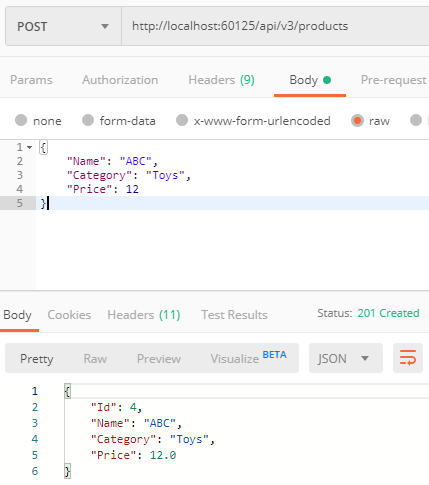
Get products by Category as parameter.



POST: api/v3/products

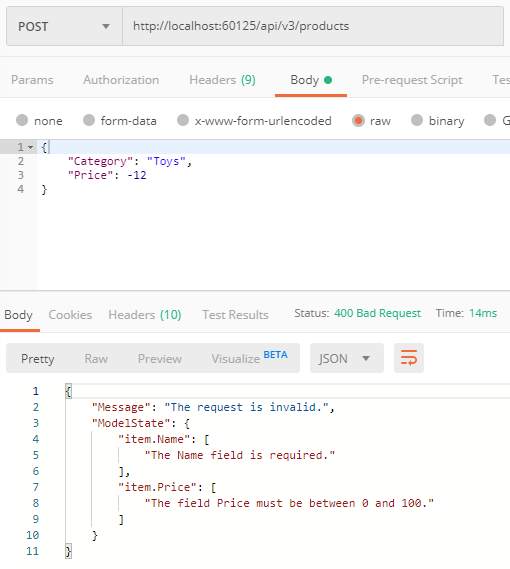
Link: <http://localhost:60125/api/v3/products>

Create product using Post.

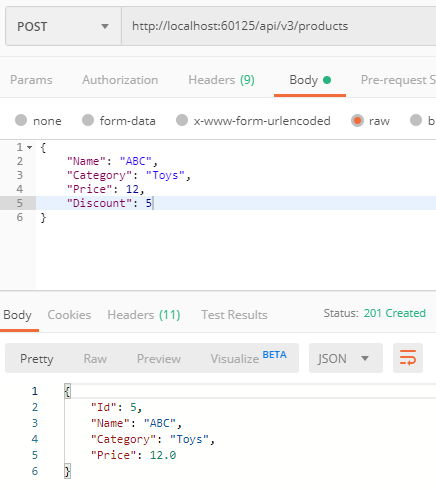


“Under-posting” will lead to 400 Bad Request. (Required: Name, Price\*)

*\*Price must be between 0 to 100*



“Over-posting” will lead to 201 Created. (“Over-post” values are excluded in creation)

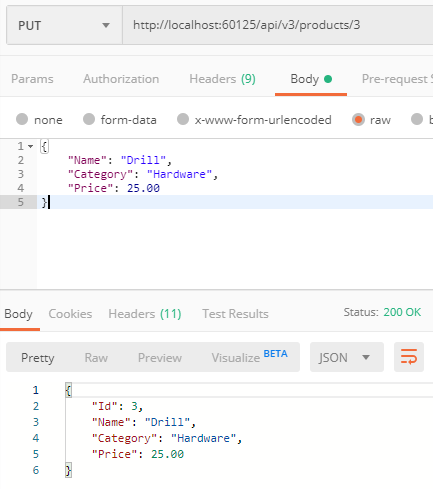


^ ”Discount” is excluded during creation

PUT: api/v3/products/{**id**}

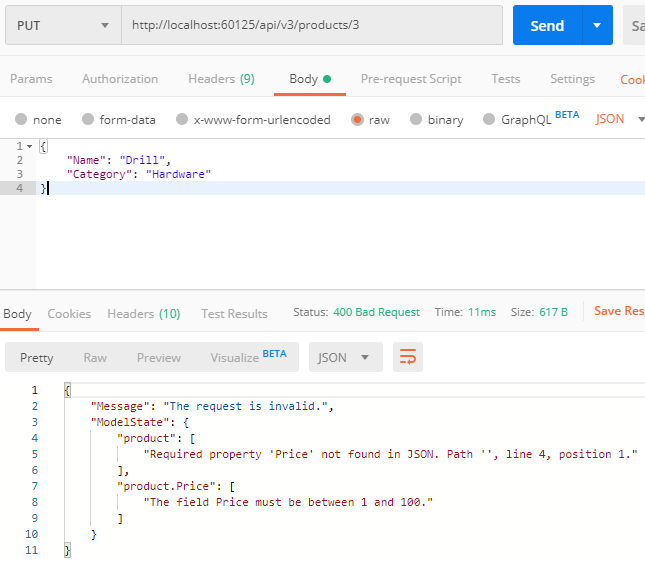
Link: <http://localhost:60125/api/v3/products/3>

Change product details based on id using Put. (Available Id between 1 to 3)

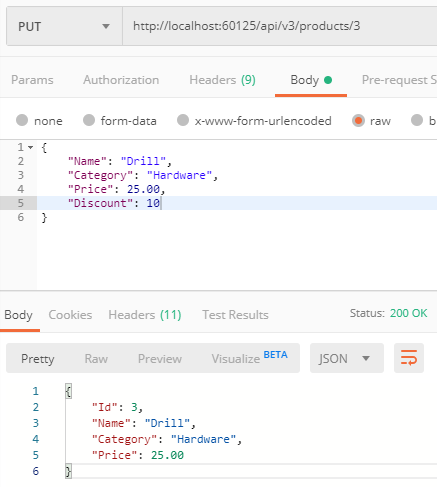


“Under-posting” will lead to 400 Bad Request. (Required: Name, Price\*)

*\*Price must be between 0 to 100*

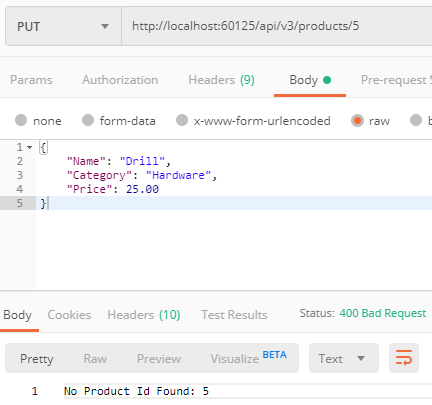


“Over-posting” will lead to 200 OK. (“Over-post” values are excluded in creation)



^ ”Discount” is excluded during creation

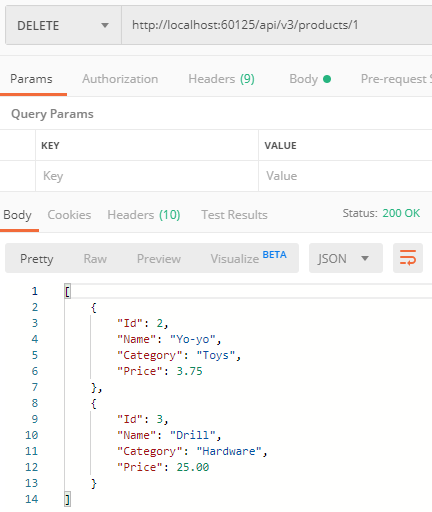
Providing invalid id will lead to 400 Bad Request



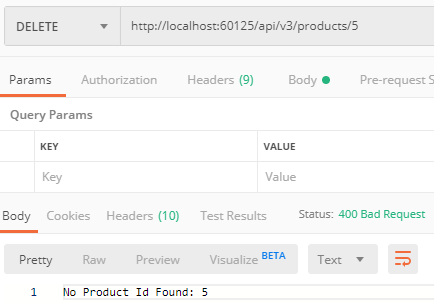
DELETE: api/v3/products/{**id**}

Link: <http://localhost:60125/api/v1/products/1>

Remove product based on id using Delete. (Available Id between 1 to 3)



Providing invalid id will lead to 400 Bad Request



Task 3

sth

Task 4

sth

Task 5

sth

Task 6

sth

Task 7

sth