

## Task 2

### Browser

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

*(There will be a simulated loading)*



### All Products

- Tomato Soup: \$1
- Yo-yo: \$3.75
- Hammer: \$16.99

### Search by ID

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

### Search by ID

Tomato Soup: \$1

In the case of error, client will reconnect and prompt to check.

### All Products

❌ Fail to get products. Reloading... (Check your Uri in the project)

### Search by ID

## 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:60125/api/v1/products/version>

Get API version.

GET

▼

http://localhost:60125/api/v1/products/version

Params

Authorization

Headers (9)

Body ●

Pre-request S

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Pretty

Raw

Preview

Visualize BETA

JSON ▼

1

2

3

4

5

[

"hello",

"version 1",

"1"

]

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.

The screenshot shows a REST client interface with the following components:

- Request Bar:** Method: GET, URL: `http://localhost:60125/api/v1/products/message?name1=ji&na...`, Send button.
- Params Tab:** Query Params table with columns KEY, VALUE, and DESCRIPTION.
- Response Bar:** Status: 200 OK, Time: 9ms, Size: 491 B, Save Response button.
- Body Tab:** Pretty view of the response body.

	KEY	VALUE	DESCRIPTION	...
<input checked="" type="checkbox"/>	name1	ji		
<input checked="" type="checkbox"/>	name2	jii1		
<input checked="" type="checkbox"/>	name3	ji3		
	Key	Value	Description	

```
1 <html>
2
3 <body>Hello World name1 =ji name2= jii1 name3=ji3 is provided in path parameter</body>
4
5 </html>
```

GET: api/v1/products

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

Get all products.

GET

http://localhost:60125/api/v1/products

Params

Authorization

Headers (9)

Body ●

Pre-request S

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Pretty

Raw

Preview

Visualize BETA

JSON

```
1  [
2    {
3      "Id": 1,
4      "Name": "Tomato Soup",
5      "Category": "Groceries",
6      "Price": 1.0
7    },
8    {
9      "Id": 2,
10     "Name": "Yo-yo",
11     "Category": "Toys",
12     "Price": 3.75
13   },
14   {
15     "Id": 3,
16     "Name": "Hammer",
17     "Category": "Hardware",
18     "Price": 16.99
19   }
20 ]
```

GET: api/v1/products/{id}

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

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

GET ▼ http://localhost:60125/api/v1/products/1

Params Authorization Headers (9) Body ● Pre-request S

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (10) Test Results Status: 200 OK

Pretty Raw Preview Visualize BETA JSON ▼ ↺

```
1 {  
2   "Id": 1,  
3   "Name": "Tomato Soup",  
4   "Category": "Groceries",  
5   "Price": 1.0  
6 }
```

Providing invalid id will lead to 404 Not Found.

GET ▼ http://localhost:60125/api/v1/products/5

Params Authorization Headers (9) Body ● Pre-request S

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (9) Test Results Status: 404 Not Found

Pretty Raw Preview Visualize BETA Text ▼ ↺

```
1 
```

GET: api/v2/products

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

Get all products.

GET

http://localhost:60125/api/v2/products

Params

Authorization

Headers (9)

Body ●

Pre-request S

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Pretty

Raw

Preview

Visualize BETA

JSON ▼

```
1  [
2    {
3      "Id": 1,
4      "Name": "Tomato soup",
5      "Category": "Groceries",
6      "Price": 1.39
7    },
8    {
9      "Id": 2,
10     "Name": "Yo-yo",
11     "Category": "Toys",
12     "Price": 3.75
13   },
14   {
15     "Id": 3,
16     "Name": "Hammer",
17     "Category": "Hardware",
18     "Price": 16.99
19   }
20 ]
```

GET: api/v2/products/{id}

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

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

GET ▼ http://localhost:60125/api/v2/products/1

Params Authorization Headers (9) Body ● Pre-request

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (10) Test Results Status: 200 OK

Pretty Raw Preview Visualize BETA JSON ▼

```
1 {  
2   "Id": 1,  
3   "Name": "Tomato soup",  
4   "Category": "Groceries",  
5   "Price": 1.39  
6 }
```

Providing invalid id will lead to 400 Bad Request

GET ▼ http://localhost:60125/api/v2/products/5

Params Authorization Headers (9) Body ● Pre-request S

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (10) Test Results Status: 400 Bad Request

Pretty Raw Preview Visualize BETA Text ▼

```
1 No Product Id Found: 5
```



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

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

Get products by Category as parameter.

GET ▼ http://localhost:60125/api/v2/products?Category=Toys

Params ● Authorization Headers (9) Body ● Pre-request

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	Category	Toys
	Key	Value

Body Cookies Headers (10) Test Results Status: 200 OK T

Pretty Raw Preview Visualize BETA JSON ▼

```
1 [
2   {
3     "Id": 2,
4     "Name": "Yo-yo",
5     "Category": "Toys",
6     "Price": 3.75
7   }
8 ]
```

POST: api/v2/products

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

Create product using Post.

The screenshot displays a REST client interface with the following components:

- Request Section:**
  - Method: POST
  - URL: http://localhost:60125/api/v2/products
  - Body Type: raw (selected)
  - Body Content:

```
1 {  
2   "Name": "Potato soup",  
3   "Category": "Groceries",  
4   "Price": 2.00  
5 }
```
- Response Section:**
  - Status: 201 Created
  - Body Type: Pretty (selected)
  - Body Content:

```
1 {  
2   "Id": 4,  
3   "Name": "Potato soup",  
4   "Category": "Groceries",  
5   "Price": 2.00  
6 }
```

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)

The screenshot displays a REST client interface with the following components:

- Request Section:**
  - Method: PUT
  - URL: http://localhost:60125/api/v2/products/2
  - Body Type: raw (selected)
  - Body Content:

```
1 {  
2   "Name": "Car",  
3   "Category": "Toy",  
4   "Price": 100.00  
5 }
```
- Response Section:**
  - Status: 200 OK
  - Body Type: JSON (selected)
  - Body Content:

```
1 {  
2   "Id": 2,  
3   "Name": "Car",  
4   "Category": "Toy",  
5   "Price": 100.00  
6 }
```

Providing invalid id will lead to 400 Bad Request

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://localhost:60125/api/v2/products/5
- Body Tab:** Selected, showing a JSON object:

```
{  "Name": "Car",  "Category": "Toy",  "Price": 100.00}
```
- Response Tab:** Selected, showing a status of 400 Bad Request and the message: "No Product Id Found: 5"

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)

DELETE

http://localhost:60125/api/v2/products/2

Params

Authorization

Headers (9)

Body ●

Pre-request

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Pretty

Raw

Preview

Visualize BETA

JSON

```
1  [
2    {
3      "Id": 1,
4      "Name": "Tomato soup",
5      "Category": "Groceries",
6      "Price": 1.39
7    },
8    {
9      "Id": 3,
10     "Name": "Hammer",
11     "Category": "Hardware",
12     "Price": 16.99
13   }
14 ]
```

Providing invalid id will lead to 400 Bad Request

DELETE

http://localhost:60125/api/v2/products/5

Params

Authorization

Headers (9)

Body ●

Pre-request S

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (10)

Test Results

Status: 400 Bad Request

Pretty

Raw

Preview

Visualize BETA

Text

1

No Product Id Found: 5

GET: api/v3/products

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

Get all products.

GET

http://localhost:60125/api/v3/products

Params

Authorization

Headers (9)

Body ●

Pre-request

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Pretty

Raw

Preview

Visualize BETA

JSON

1

[

2

{

3

"Id": 1,

4

"Name": "Tomato soup",

5

"Category": "Groceries",

6

"Price": 1.39

7

},

8

{

9

"Id": 2,

10

"Name": "Yo-yo",

11

"Category": "Toys",

12

"Price": 3.75

13

},

14

{

15

"Id": 3,

16

"Name": "Hammer",

17

"Category": "Hardware",

18

"Price": 16.99

19

}

20

]

GET: api/v3/products/{id}

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

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

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

Params Authorization Headers (9) Body ● Pre-request

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (10) Test Results Status: 200 OK

Pretty Raw Preview Visualize BETA JSON ↺

```
1 {
2   "Id": 3,
3   "Name": "Hammer",
4   "Category": "Hardware",
5   "Price": 16.99
6 }
```

Providing invalid id will lead to 400 Bad Request

GET <http://localhost:60125/api/v3/products/5>

Params Authorization Headers (9) Body ● Pre-request

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (10) Test Results Status: 400 Bad Request

Pretty Raw Preview Visualize BETA Text ↺

```
1 No Product Id Found: 5
```



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

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

Get products by Category as parameter.

GET

▼

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

Params ●

Authorization

Headers (9)

Body ●

Pre-request Script

Query Params

	KEY	VALUE	DE
✓	Category	Groceries	
	Key	Value	D

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 25ms

Pretty

Raw

Preview

Visualize BETA

JSON ▼

≡

1

2

3

4

5

6

7

8

[

{

"Id": 1,

"Name": "Tomato soup",

"Category": "Groceries",

"Price": 1.39

}

]

POST: api/v3/products

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

Create product using Post.

The screenshot displays a REST client interface with the following components:

- Request Section:**
  - Method: POST
  - URL: http://localhost:60125/api/v3/products
  - Body tab selected, showing raw JSON: 

```
{  "Name": "ABC",  "Category": "Toys",  "Price": 12}
```
- Response Section:**
  - Status: 201 Created
  - Body tab selected, showing pretty-printed JSON: 

```
{  "Id": 4,  "Name": "ABC",  "Category": "Toys",  "Price": 12.0}
```

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

*\*Price must be between 0 to 100*

POST http://localhost:60125/api/v3/products

Params Authorization Headers (9) Body ● Pre-request Script Test

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● G

```
1 {  
2   "Category": "Toys",  
3   "Price": -12  
4 }
```

Body Cookies Headers (10) Test Results Status: 400 Bad Request Time: 14ms

Pretty Raw Preview Visualize BETA JSON ▼

```
1 {  
2   "Message": "The request is invalid.",  
3   "ModelState": {  
4     "item.Name": [  
5       "The Name field is required."  
6     ],  
7     "item.Price": [  
8       "The field Price must be between 0 and 100."  
9     ]  
10  }  
11 }
```

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

The screenshot shows a REST client interface. At the top, a POST request is configured for the URL `http://localhost:60125/api/v3/products`. The 'Body' tab is selected, showing a JSON payload: 

```
{  "Name": "ABC",  "Category": "Toys",  "Price": 12,  "Discount": 5}
```

. Below the request, the response is displayed with a status of '201 Created'. The response body is shown in 'Pretty' JSON format: 

```
{  "Id": 5,  "Name": "ABC",  "Category": "Toys",  "Price": 12.0}
```

POST `http://localhost:60125/api/v3/products`

Params Authorization Headers (9) **Body** Pre-request S

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ b

```
1 {
2   "Name": "ABC",
3   "Category": "Toys",
4   "Price": 12,
5   "Discount": 5
6 }
```

Body Cookies Headers (11) Test Results Status: 201 Created

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "Id": 5,
3   "Name": "ABC",
4   "Category": "Toys",
5   "Price": 12.0
6 }
```

^ “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)

The screenshot shows a REST client interface with a PUT request to `http://localhost:60125/api/v3/products/3`. The request body is a JSON object: `{ "Name": "Drill", "Category": "Hardware", "Price": 25.00 }`. The response status is `200 OK`, and the response body is a JSON object: `{ "Id": 3, "Name": "Drill", "Category": "Hardware", "Price": 25.00 }`.

PUT `http://localhost:60125/api/v3/products/3`

Params Authorization Headers (9) **Body** Pre-request

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ text

```
1 {
2   "Name": "Drill",
3   "Category": "Hardware",
4   "Price": 25.00
5 }
```

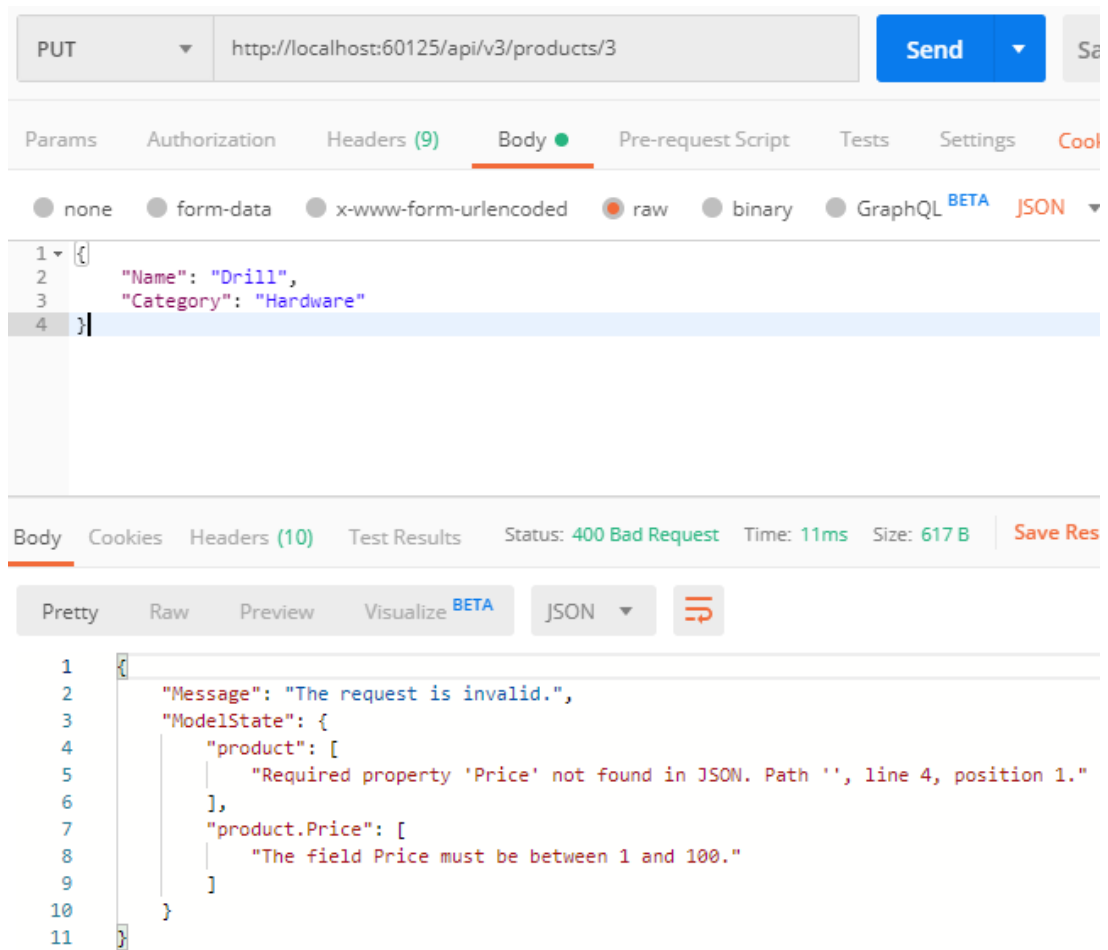
Body Cookies Headers (11) Test Results Status: 200 OK

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "Id": 3,
3   "Name": "Drill",
4   "Category": "Hardware",
5   "Price": 25.00
6 }
```

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

*\*Price must be between 0 to 100*



PUT http://localhost:60125/api/v3/products/3

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cool

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON

```
1 {
2   "Name": "Drill",
3   "Category": "Hardware"
4 }
```

Body Cookies Headers (10) Test Results Status: 400 Bad Request Time: 11ms Size: 617 B Save Res

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "Message": "The request is invalid.",
3   "ModelState": {
4     "product": [
5       "Required property 'Price' not found in JSON. Path '', line 4, position 1."
6     ],
7     "product.Price": [
8       "The field Price must be between 1 and 100."
9     ]
10  }
11 }
```

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

The screenshot shows a REST client interface. At the top, a PUT request is configured for the URL `http://localhost:60125/api/v3/products/3`. The 'Body' tab is selected, showing a JSON payload: 

```
{  "Name": "Drill",  "Category": "Hardware",  "Price": 25.00,  "Discount": 10}
```

. Below the request, the 'Response' tab is active, displaying a status of '200 OK' and a JSON response: 

```
{  "Id": 3,  "Name": "Drill",  "Category": "Hardware",  "Price": 25.00}
```

. The response JSON is formatted in 'Pretty' view. The 'Discount' field is present in the request but absent in the response.

^ “Discount” is excluded during creation

Providing invalid id will lead to 400 Bad Request

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://localhost:60125/api/v3/products/5
- Body:** A JSON object with the following fields:

```
{  "Name": "Drill",  "Category": "Hardware",  "Price": 25.00}
```
- Status:** 400 Bad Request
- Response Body:** No Product Id Found: 5



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)

DELETE

http://localhost:60125/api/v3/products/1

Params

Authorization

Headers (9)

Body ●

Pre-request

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Pretty

Raw

Preview

Visualize BETA

JSON

1

[

2

{

3

"Id": 2,

4

"Name": "Yo-yo",

5

"Category": "Toys",

6

"Price": 3.75

7

},

8

{

9

"Id": 3,

10

"Name": "Drill",

11

"Category": "Hardware",

12

"Price": 25.00

13

}

14

]

Providing invalid id will lead to 400 Bad Request

DELETE

http://localhost:60125/api/v3/products/5

Params

Authorization

Headers (9)

Body ●

Pre-request S

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (10)

Test Results

Status: 400 Bad Request

Pretty

Raw

Preview

Visualize BETA

Text

1

No Product Id Found: 5