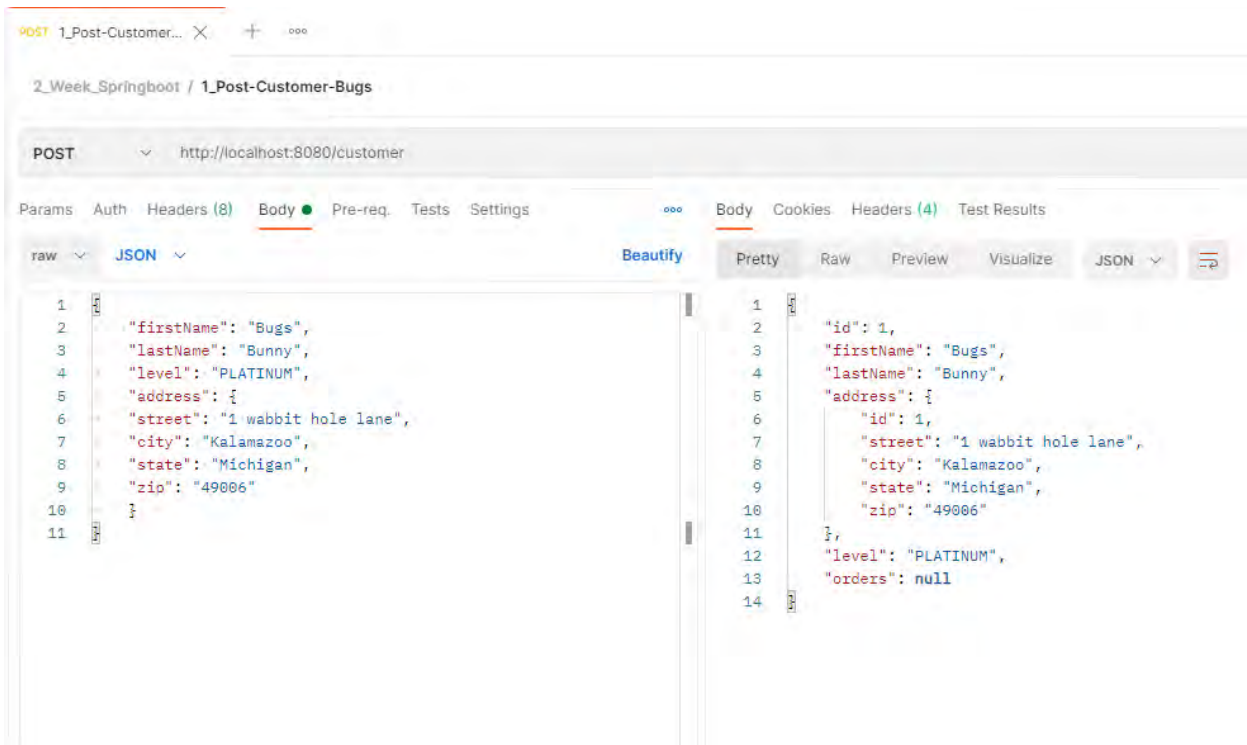


Screenshots of Running Application:

Text document titled “JeffBeck_Week_2_CodingAssignment_Postman_Tests” saved to Git details data used and steps for testing code in postman. Each step in text document corresponds with steps listed below:

Text document titled “CommandPrompt-mvn spring-boot run” is the CLI documentations showing the spring-boot API was able to make establish the necessary endpoints in order to carry out and test the API in Postman...

(1) POST (body/raw/JSON) http://localhost:8080/customer



(2) //2_Post-Customer-Elmer

POST http://localhost:8080/customer

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON Beautify

```
1 {
2   "firstName": "Elmer",
3   "lastName": "Fudd",
4   "level": "PLATINUM",
5   "address": {
6     "street": "123 Hunters Rd.",
7     "city": "Battle Creek",
8     "state": "Michigan",
9     "zip": "49015"
10  }
11 }
```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "firstName": "Elmer",
4   "lastName": "Fudd",
5   "address": {
6     "id": 2,
7     "street": "123 Hunters Rd.",
8     "city": "Battle Creek",
9     "state": "Michigan",
10    "zip": "49015"
11  },
12  "level": "PLATINUM",
13  "orders": null
14 }
```

(3) //3_Post-Customer-Wile

2_Week_Springboot / 3_Post-Customer-Wile

POST http://localhost:8080/customer

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON Beautify

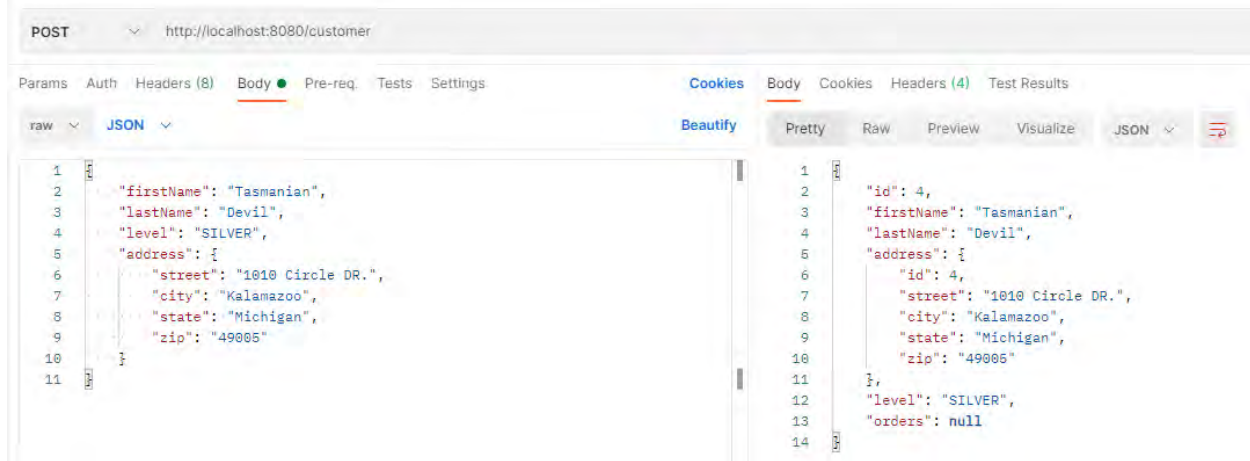
```
1 {
2   "firstName": "Wile E.",
3   "lastName": "Coyote",
4   "level": "SILVER",
5   "address": {
6     "street": "2020 Acme Lane",
7     "city": "Portage",
8     "state": "Michigan",
9     "zip": "49002"
10  }
11 }
```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 3,
3   "firstName": "Wile E.",
4   "lastName": "Coyote",
5   "address": {
6     "id": 3,
7     "street": "2020 Acme Lane",
8     "city": "Portage",
9     "state": "Michigan",
10    "zip": "49002"
11  },
12  "level": "SILVER",
13  "orders": null
14 }
```

(4) //4_Post-Customer-Tasmanian



(5) GET (body/raw/JSON) <http://localhost:8080/customer>



```
1  [
2    {
3      "id": 1,
4      "firstName": "Bugs",
5      "lastName": "Bunny",
6      "address": {
7        "id": 1,
8        "street": "1 wabbit hole lane",
9        "city": "Kalamazoo",
10       "state": "Michigan",
11       "zip": "49006"
12     },
13     "level": "PLATINUM",
14     "orders": []
15   },
16   {
17     "id": 2,
18     "firstName": "Elmer",
19     "lastName": "Fudd",
20     "address": {
21       "id": 2,
22       "street": "123 Hunters Rd.",
23       "city": "Battle Creek",
24       "state": "Michigan",
25       "zip": "49015"
26     },
27     "level": "PLATINUM",
28     "orders": []
29   },
30   {
31     "id": 3,
32     "firstName": "Wile E.",
33     "lastName": "Coyote",
34     "address": {
35       "id": 3,
36       "street": "2020 Acme Lane",
37       "city": "Portage",
38       "state": "Michigan",
39       "zip": "49002"
40     },
41     "level": "SILVER",
42     "orders": []
43   },
44   {
45     "id": 4,
46     "firstName": "Tasmanian",
47     "lastName": "Devil",
48     "address": {
49       "id": 4,
50       "street": "1010 Circle DR.",
51       "city": "Kalamazoo",
52       "state": "Michigan",
53       "zip": "49005"
54     },
55     "level": "SILVER",
56     "orders": []
57   }
58 ]
```

(6) PUT (body/raw/JSON) <http://localhost:8080/customer/2>

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/customer/2`. The request body is a JSON object with the following structure:

```
1 {  
2   "firstName": "Elmer",  
3   "lastName": "Fudd",  
4   "level": "GOLD",  
5   "address": {  
6     "street": "123 Hunters Rd.",  
7     "city": "Battle Creek",  
8     "state": "Michigan",  
9     "zip": "49015"  
10  }  
11 }
```

The response is also in JSON format, showing the server's acknowledgment of the PUT operation:

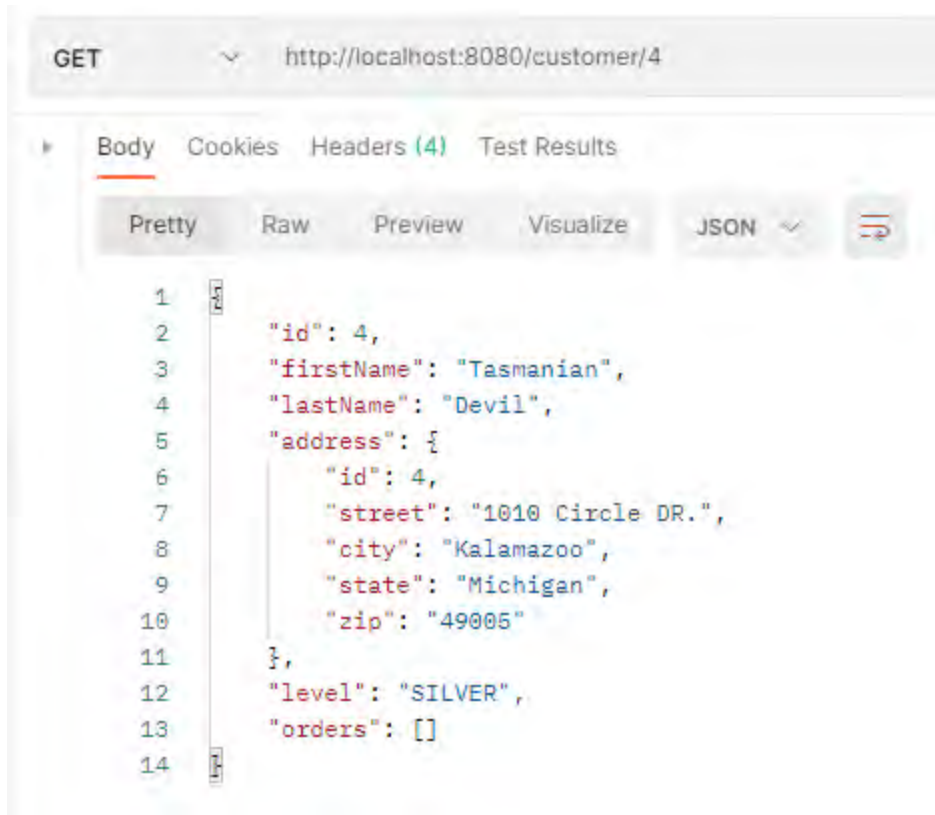
```
1 {  
2   "id": 2,  
3   "firstName": "Elmer",  
4   "lastName": "Fudd",  
5   "address": {  
6     "id": 5,  
7     "street": "123 Hunters Rd.",  
8     "city": "Battle Creek",  
9     "state": "Michigan",  
10    "zip": "49015"  
11  },  
12  "level": "GOLD",  
13  "orders": []  
14 }
```

(7) GET (body/raw/JSON) <http://localhost:8080/customer/2>

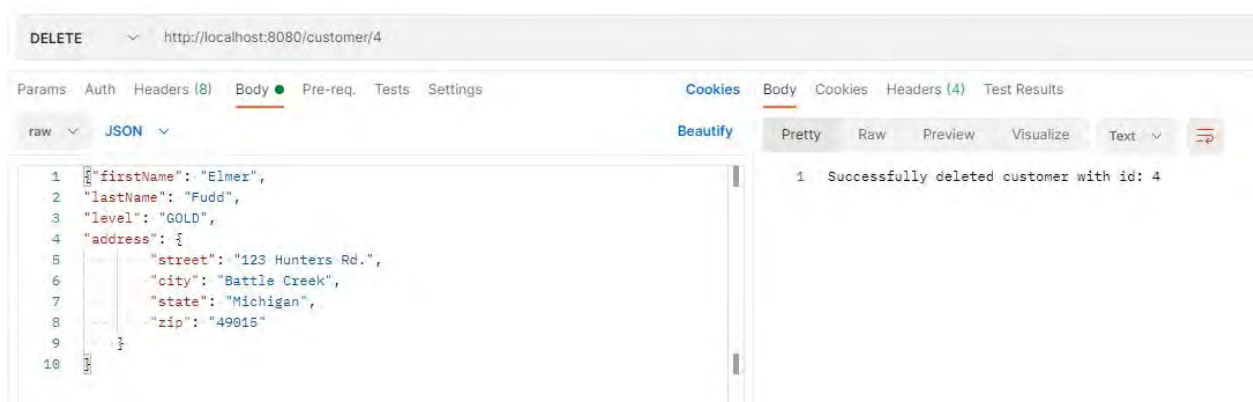
The screenshot shows a REST client interface with a GET request to `http://localhost:8080/customer/2`. The response body is a JSON object, identical to the one shown in the previous screenshot:

```
1 {  
2   "id": 2,  
3   "firstName": "Elmer",  
4   "lastName": "Fudd",  
5   "address": {  
6     "id": 5,  
7     "street": "123 Hunters Rd.",  
8     "city": "Battle Creek",  
9     "state": "Michigan",  
10    "zip": "49015"  
11  },  
12  "level": "GOLD",  
13  "orders": []  
14 }
```

(8) GET (body/raw/JSON) <http://localhost:8080/customer/4>



(9) DELETE (body/raw/JSON) <http://localhost:8080/customer/4>



(10) GET (body/raw/JSON) <http://localhost:8080/customer/>

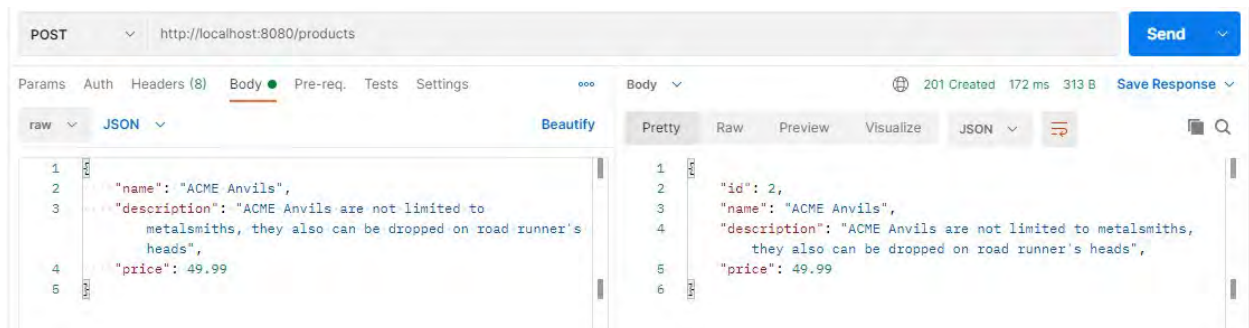
The screenshot shows a web browser's developer tools with the 'Network' tab selected. A GET request to <http://localhost:8080/customer/> is shown. The response body is displayed in JSON format, showing three customer objects. The first two are 'Bugs Bunny' and 'Elmer Fudd', and the third is 'Wile E. Coyote'.

```
1  [
2    {
3      "id": 1,
4      "firstName": "Bugs",
5      "lastName": "Bunny",
6      "address": {
7        "id": 1,
8        "street": "1 wabbit hole lane",
9        "city": "Kalamazoo",
10       "state": "Michigan",
11       "zip": "49006"
12     },
13     "level": "PLATINUM",
14     "orders": []
15   },
16   {
17     "id": 2,
18     "firstName": "Elmer",
19     "lastName": "Fudd",
20     "address": {
21       "id": 5,
22       "street": "123 Hunters Rd.",
23       "city": "Battle Creek",
24       "state": "Michigan",
25       "zip": "49015"
26     },
27     "level": "GOLD",
28     "orders": []
29   },
30   {
31     "id": 3,
32     "firstName": "Wile E.",
33     "lastName": "Coyote",
34     "address": {
35       "id": 3,
36       "street": "2020 Acme Lane",
37       "city": "Portage",
38       "state": "Michigan",
39       "zip": "49002"
40     },
41     "level": "SILVER",
42     "orders": []
43   }
44 ]
```


(11) POST (body/raw/JSON) <http://localhost:8080/products>



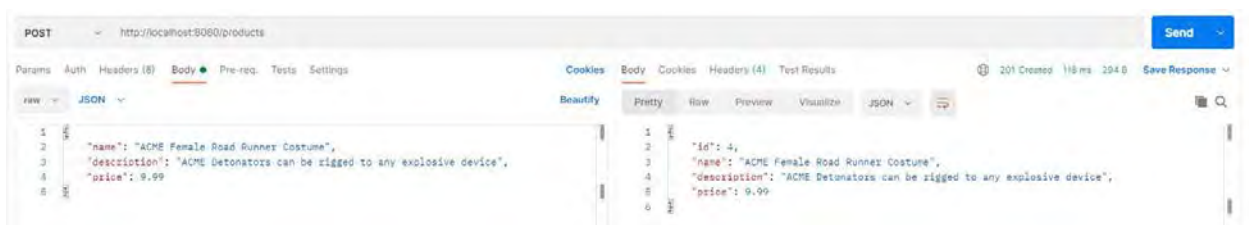
(12) // POST-Products-Anvils



(13) // POST-Products-Detonator



(14) // POST-Products-FemaleRoadRunnerCostume



(15) // POST-Products-BuckShot

POST http://localhost:8080/products

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON Beautify

```
1 {
2   "name": "ACME Buck Shot",
3   "description": "ACME Buck Shot can be used in any
4     situation where weapons are involved and is so tiny
5     even spiders can use it to catch elusive insects.",
6   "price": 14.75
7 }
```

Body 201 Created 99 ms 360 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 5,
3   "name": "ACME Buck Shot",
4   "description": "ACME Buck Shot can be used in any situation
5     where weapons are involved and is so tiny even spiders
6     can use it to catch elusive insects.",
7   "price": 14.75
8 }
```

(16) // GET (body/raw/JSON) <http://localhost:8080/products>

GET http://localhost:8080/products

Body Cookies Headers (4) Test Results Status: 200 OK Time: 40 ms

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "name": "ACME Boomerang",
5     "description": "The ACME boomerangs are guaranteed to always return to Wile E. Coyote when he is not looking...",
6     "price": 24.99
7   },
8   {
9     "id": 2,
10    "name": "ACME Anvils",
11    "description": "ACME Anvils are not limited to metalsmiths, they also can be dropped on road runner's heads",
12    "price": 49.99
13  },
14  {
15    "id": 3,
16    "name": "ACME Detonator",
17    "description": "ACME Detonators can be rigged to any explosive device",
18    "price": 9.99
19  },
20  {
21    "id": 5,
22    "name": "ACME Buck Shot",
23    "description": "ACME Buck Shot can be used in any situation where weapons are involved and is so tiny even spiders can use it to catch elusive insects.",
24    "price": 14.75
25  },
26  {
27    "id": 6,
28    "name": "ACME Female Road Runner Costume",
29    "description": "The ACME Female Road Runner Costume is used by cross dressing animal lovers everywhere",
30    "price": 79.99
31  }
32 ]
```

(17) //PUT /products/1

PUT http://localhost:8080/products/1

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON Beautify

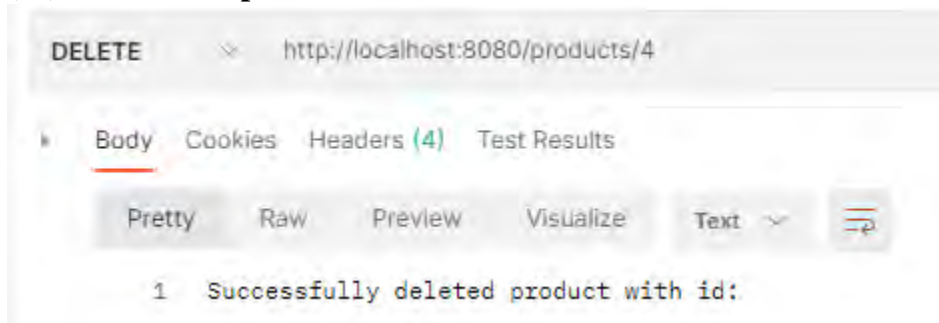
```
1 {
2   "name": "ACME Boomerang",
3   "description": "The ACME boomerangs are guaranteed to
4     always return to Wile E. Coyote when he is not
5     looking...",
6   "price": 24.99
7 }
```

Body 200 OK 95 ms 315 B Save Response

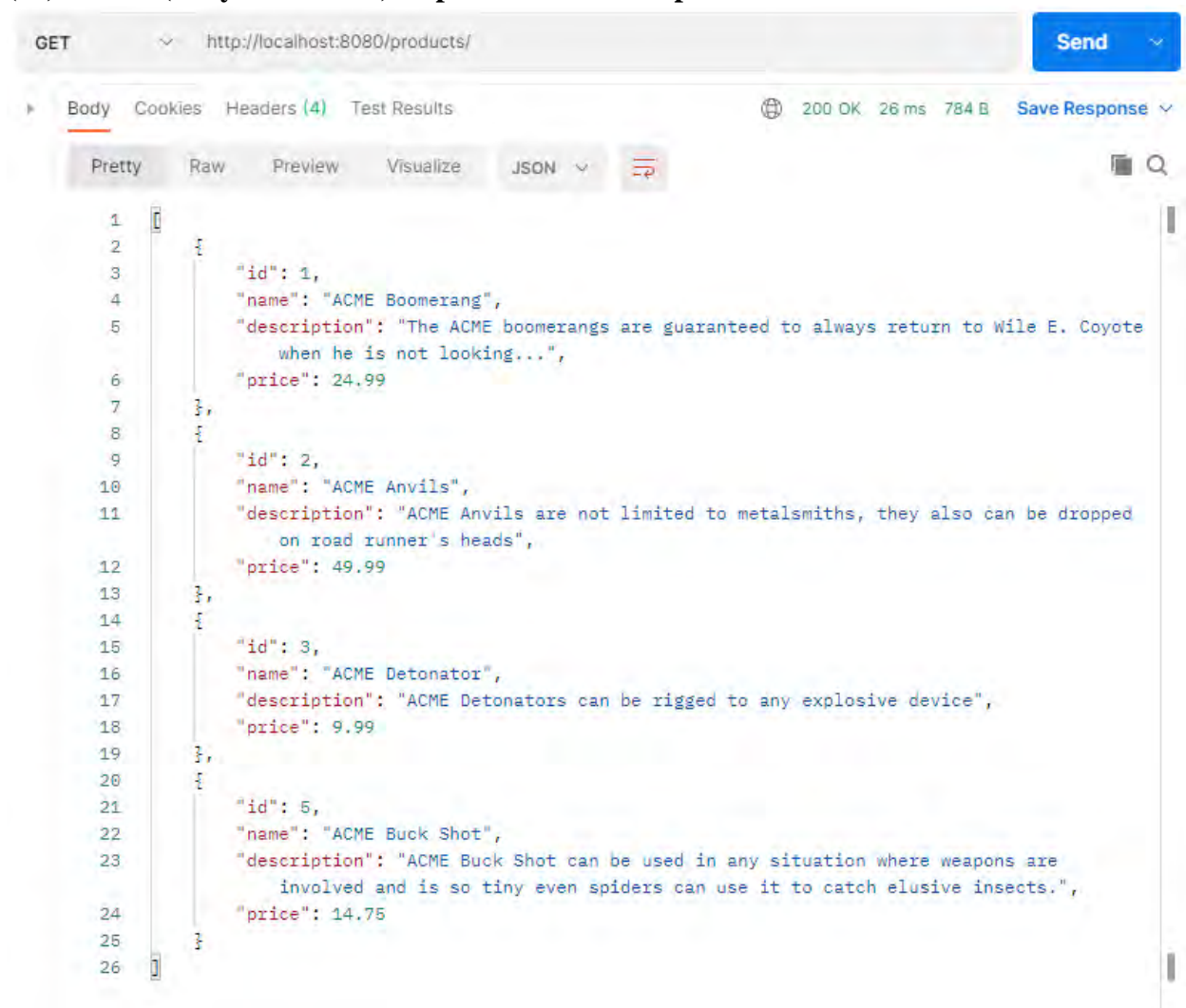
Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "ACME Boomerang",
4   "description": "The ACME boomerangs are guaranteed to
5     always return to Wile E. Coyote when he is not looking..
6   ",
7   "price": 24.99
8 }
```

(18) // DELETE /products/4



(18) // GET (body/raw/JSON) http://localhost:8080/products



(19) // POST /customers/3/orders -- Wile E. Coyote(3) - Detonator (3) –

POST http://localhost:8080/customer/3/orders

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

1 [3, 3]

POST http://localhost:8080/customer/3/orders

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "ordered": {
4     "year": 2021,
5     "month": "APRIL",
6     "era": "CE",
7     "dayOfMonth": 18,
8     "dayOfWeek": "SUNDAY",
9     "dayOfYear": 108,
10    "leapYear": false,
11    "monthValue": 4,
12    "chronology": {
13      "id": "ISO",
14      "calendarType": "iso8601"
15    }
16  },
17  "estimatedDelivery": {
18    "year": 2021,
19    "month": "APRIL",
20    "era": "CE",
21    "dayOfMonth": 25,
22    "dayOfWeek": "SUNDAY",
23    "dayOfYear": 115,
24    "leapYear": false,
25    "monthValue": 4,
26    "chronology": {
27      "id": "ISO",
28      "calendarType": "iso8601"
29    }
30  },
31  "delivered": null,
32  "invoiceAmount": 9.4905,
33  "status": "ORDERED",
34  "products": [
35    {
36      "id": 3,
37      "name": "ACME Detonator",
38      "description": "ACME Detonators can be rigged to any explosive device",
39      "price": 9.99
40    }
41  ]
42 }
```

(20) POST /customer/2/orders – Elmer Fudd (2) – Anvils (5)

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/customer/2/orders`. The request body is a JSON object. The response status is 201 Created.

Request Body:

```
1  {
2    "id": 3,
3    "ordered": {
4      "year": 2021,
5      "month": "APRIL",
6      "era": "CE",
7      "dayOfMonth": 18,
8      "dayOfWeek": "SUNDAY",
9      "dayOfYear": 108,
10     "leapYear": false,
11     "monthValue": 4,
12     "chronology": {
13       "id": "ISO",
14       "calendarType": "iso8601"
15     }
16   },
17   "estimatedDelivery": {
18     "year": 2021,
19     "month": "APRIL",
20     "era": "CE",
21     "dayOfMonth": 25,
22     "dayOfWeek": "SUNDAY",
23     "dayOfYear": 115,
24     "leapYear": false,
25     "monthValue": 4,
26     "chronology": {
27       "id": "ISO",
28       "calendarType": "iso8601"
29     }
30   },
31   "delivered": null,
32   "invoiceAmount": 58.266000000000005,
33   "status": "ORDERED",
34   "products": [
35     {
36       "id": 2,
37       "name": "ACME Anvils",
38       "description": "ACME Anvils are not limited to metalsmiths, they also can be
39         dropped on road runner's heads",
40       "price": 49.99
41     },
42     {
43       "id": 5,
44       "name": "ACME Buck Shot",
45       "description": "ACME Buck Shot can be used in any situation where weapons are
46         involved and is so tiny even soiders can use it to catch elusive insects.",
47       "price": 14.75
48     }
49   ]
50 }
```

Response: 201 Created, 335 ms, 989 B. Save Response.

(21) // PUT /customer/2/orders/1_CANCELED

The screenshot displays a REST client interface with a PUT request to `http://localhost:8080/customer/2/orders/1`. The request body is a JSON object with `"status": "CANCELED"`. The response body is a JSON object containing order details, including `"id": 1`, `"estimatedDelivery"`, `"delivered": null`, `"invoiceAmount": 9.4905`, `"status": "CANCELED"`, and `"products"` (an array with one product: `"ACME Detonator"`).

```
PUT http://localhost:8080/customer/2/orders/1

{"status": "CANCELED"}

PUT http://localhost:8080/customer/2/orders/1

{
  "id": 1,
  "ordered": {
    "year": 2021,
    "month": "APRIL",
    "era": "CE",
    "dayOfMonth": 18,
    "dayOfWeek": "SUNDAY",
    "dayOfYear": 108,
    "leapYear": false,
    "monthValue": 4,
    "chronology": {
      "id": "ISO",
      "calendarType": "iso8601"
    }
  },
  "estimatedDelivery": {
    "year": 2021,
    "month": "APRIL",
    "era": "CE",
    "dayOfMonth": 25,
    "dayOfWeek": "SUNDAY",
    "dayOfYear": 115,
    "leapYear": false,
    "monthValue": 4,
    "chronology": {
      "id": "ISO",
      "calendarType": "iso8601"
    }
  },
  "delivered": null,
  "invoiceAmount": 9.4905,
  "status": "CANCELED",
  "products": [
    {
      "id": 3,
      "name": "ACME Detonator",
      "description": "ACME Detonators can be rigged to any explosive device",
      "price": 9.99
    }
  ]
}
```


(22) // PUT /customers/2/orders/5_DELIVERED

The screenshot displays a REST client interface with two panels. The top panel shows a PUT request to `http://localhost:8080/customer/2/orders/2` with a JSON body: `{ "status": "DELIVERED" }`. The bottom panel shows the response, which is a JSON object containing order details, delivery information, and a list of products.

```
PUT http://localhost:8080/customer/2/orders/2
Body
raw JSON Beautify
1 {
2   "status": "DELIVERED"
3 }

PUT http://localhost:8080/customer/2/orders/2
Body Cookies Headers (4) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "id": 2,
3   "ordered": {
4     "year": 2021,
5     "month": "APRIL",
6     "era": "CE",
7     "dayOfMonth": 18,
8     "dayOfWeek": "SUNDAY",
9     "dayOfYear": 108,
10    "leapYear": false,
11    "monthValue": 4,
12    "chronology": {
13      "id": "ISO",
14      "calendarType": "iso8601"
15    }
16  },
17   "estimatedDelivery": {
18     "year": 2021,
19     "month": "APRIL",
20     "era": "CE",
21     "dayOfMonth": 26,
22     "dayOfWeek": "SUNDAY",
23     "dayOfYear": 116,
24     "leapYear": false,
25     "monthValue": 4,
26     "chronology": {
27       "id": "ISO",
28       "calendarType": "iso8601"
29     }
30   },
31   "delivered": null,
32   "invoiceAmount": 58.266000000000005,
33   "status": "DELIVERED",
34   "products": [
35     {
36       "id": 5,
37       "name": "ACME Buck Shot",
38       "description": "ACME Buck Shot can be used in any situation where weapons are
39         involved and is so tiny even spiders can use it to catch elusive insects.",
40       "price": 14.75
41     },
42     {
43       "id": 2,
44       "name": "ACME Anvils",
45       "description": "ACME Anvils are not limited to metalsmiths, they also can be
46         dropped on road runner's heads",
47       "price": 49.99
48     }
49   ]
50 }
```

URL to GitHub Repository: https://github.com/becje/Springboot_Coding_Assignments