

# API AUTOMATION WITH POSTMAN AND NEWMAN SCREENSHOTS

## 1) POST METHOD :

The screenshot shows the Postman interface for a POST request. The URL is `http://localhost:3000/products`. The request method is **POST**. The **Scripts** tab is active, showing the following test scripts:

```
1 pm.test("Status code is 201", function () {
2   pm.response.to.have.status(201);
3 });
4
5 pm.test("Response has correct name and price", function () {
6   const jsonData = pm.response.json();
7   pm.expect(jsonData.name).to.eql("Tablet");
8   pm.expect(jsonData.price).to.eql(300);
9 });
```

The **Body** tab is also visible, showing the JSON response:

```
1 {
2   "id": "3",
3   "name": "Tablet",
4   "price": 300
5 }
```

The screenshot shows a web browser window displaying the JSON response from `localhost:3000/products`. The response is a JSON array containing three product objects:

```
[
  {
    "id": "1",
    "name": "Laptop",
    "price": 1000
  },
  {
    "id": "2",
    "name": "Phone",
    "price": 500
  },
  {
    "id": "3",
    "name": "Tablet",
    "price": 300
  }
]
```

## 2) PATCH METHOD :

The image shows a Postman interface for a PATCH request and a web browser displaying the response.

**Postman Interface:**

- URL:** `http://localhost:3000/products/1`
- Method:** PATCH
- Scripts:**

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Updated price is correct", function () {
6   const jsonData = pm.response.json();
7   pm.expect(jsonData.price).to.eql(1100);
8 });
9
```
- Response:** 200 OK • 16 ms • 374 B
- Body (JSON):**

```
1 {
2   "id": "1",
3   "name": "Laptop",
4   "price": 1100
5 }
```

**Web Browser Interface:**

- Address Bar:** `localhost:3000/products/1`
- Content:** Pretty-print `{ "id": "1", "name": "Laptop", "price": 1100 }`

### 3) TERMINAL OUTPUT :

```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Users\amulya.manjunatha\OneDrive - ascension\Desktop>newman run "C:\Users\amulya.manjunatha\OneDrive - ascension\Desktop\ProductAPI_Localhost3000.postman_collection.json"
newman

CAPSTONE Automation

→ POST - Create a new product
POST http://localhost:3000/products [201 Created, 378B, 120ms]
✓ Status code is 201
✓ Response has correct name and price
✓ Response time is less than 2000ms

→ PATCH - Update existing product
PATCH http://localhost:3000/products/1 [200 OK, 374B, 26ms]
✓ Status code is 200
✓ Updated price is correct
✓ Response time is less than 2000ms
```

	executed	failed
iterations	1	0
requests	2	0
test-scripts	2	0
prerequisite-scripts	0	0
assertions	6	0

```
C:\WINDOWS\system32\cmd. X + v

→ POST - Create a new product
POST http://localhost:3000/products [201 Created, 378B, 120ms]
✓ Status code is 201
✓ Response has correct name and price
✓ Response time is less than 2000ms

→ PATCH - Update existing product
PATCH http://localhost:3000/products/1 [200 OK, 374B, 26ms]
✓ Status code is 200
✓ Updated price is correct
✓ Response time is less than 2000ms
```

	executed	failed
iterations	1	0
requests	2	0
test-scripts	2	0
prerequisite-scripts	0	0
assertions	6	0

total run duration: 429ms
total data received: 103B (approx)
average response time: 73ms [min: 26ms, max: 120ms, s.d.: 47ms]

```
C:\Users\amulya.manjunatha\OneDrive - ascension\Desktop>
```

### 3) NEWMAN OUTPUT

