

# Day-24 Assignment (API)

## 1. Employees

(a) - get all employees

GET  Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 2.48 s Size: 331 B Save as example

Pretty Raw Preview Visualize JSON Copy Search

```
1 [
2   {
3     "id": 1,
4     "name": "John Doe",
5     "city": "New York"
6   },
7   {
8     "id": 2,
9     "name": "Jane Smith",
10    "city": "Los Angeles"
11  }
```

Activate Windows  
Go to Settings to activate Windows.

(b) Get specific employee using id

GET  Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 9 ms Size: 279 B Save as example

Pretty Raw Preview Visualize JSON Copy Search

```
1 {
2   "id": 1,
3   "name": "John Doe",
4   "city": "New York"
5 }
```

### (c) Creating an employee

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

- Method:** POST
- URL:** http://localhost:4000/employees
- Body:** A JSON object: 

```
{  "id": 3,  "name": "Faizan Nawaz",  "city": "New Delhi"}
```
- Status:** 201 Created
- Time:** 52 ms
- Size:** 289 B
- Response Body:** The same JSON object as the request body, displayed in a pretty-printed format.

### (d) Update employee using id

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

- Method:** PUT
- URL:** http://localhost:4000/employees/3
- Body:** A JSON object: 

```
{  "id": 3,  "name": "Faizan Anwer",  "city": "New Delhi"}
```
- Status:** 200 OK
- Time:** 9 ms
- Size:** 284 B
- Response Body:** The same JSON object as the request body, displayed in a pretty-printed format.

### (e) Deleting employee using id

DELETE ▼ http://localhost:4000/employees/3 Send ▼

Params Authorization Headers (9) **Body** ● Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1 {
2   "id": 3,
3   "name": "Faizan Anwer",
4   "city": "New Delhi"
5 }
```

Body Cookies Headers (7) Test Results ⊕ Status: 200 OK Time: 242 ms Size: 286 B 📄 Save as example ⋮

Pretty Raw Preview Visualize **JSON** ▼ 🔍

```
1 [
2   {
3     "id": 3,
4     "name": "Faizan Anwer",
5     "city": "New Delhi"
6   }
7 ]
```

Activate Windows

### (f) Verification

GET ▼ http://localhost:4000/employees Send ▼

Params Authorization Headers (7) **Body** **Scripts** ● Settings Cookies

Pre-request

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

Post-response \*

```
6 pm.test("Response has the required fields - id, name, and city",
7   function () {
8     const responseData = pm.response.json();
9     pm.expect(responseData).to.be.an('array');
10
11    responseData.forEach(function(employee) {
```

🔍 Response headers: Content-Type header check  
Response time is less than 200ms  
Status code: Successful POST request  
Status code: Code name has string  
Response body: Convert XML body to a JSON Object  
Use Tiny Validator for JSON data

Body Cookies Headers (7) **Test Results (5/5)** ⊕ Status: 200 OK Time: 8 ms Size: 331 B 📄 Save as example ⋮

ℹ️ Perform API tests faster with templates for integration testing, regression testing, and more. View Templates ✕

All Passed Skipped Failed 🔄

**PASS** Response status code is 200

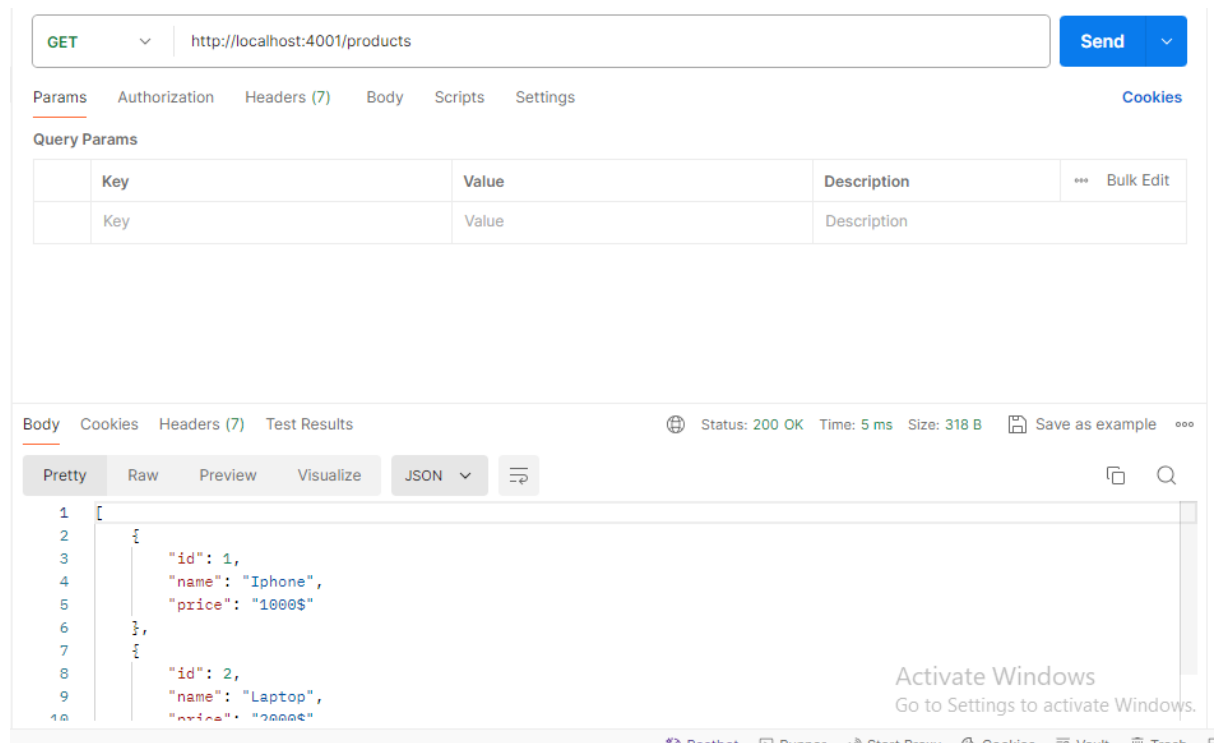
**PASS** Response has the required fields - id, name, and city

**PASS** Each employee object has non-negative integer id, non-empty string name, and city

Activate Windows  
Go to Settings to activate Windows.

## 2. Products

### (a) Get all products



GET ▼ | http://localhost:4001/products Send ▼

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

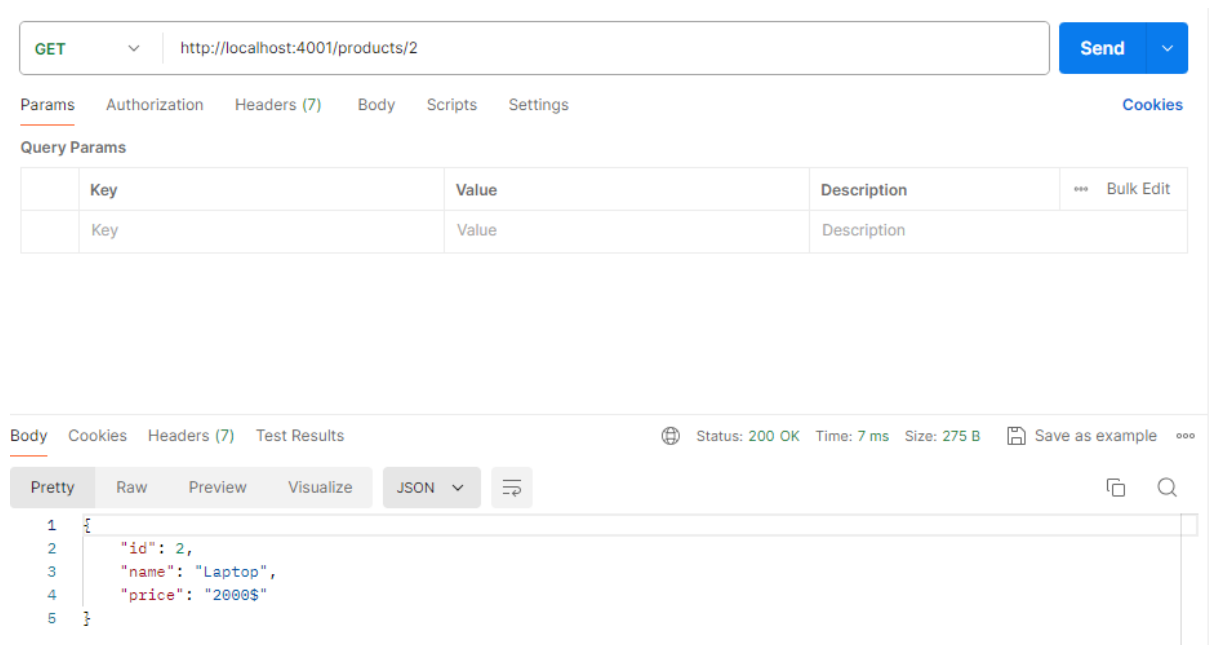
Body Cookies Headers (7) Test Results 🌐 Status: 200 OK Time: 5 ms Size: 318 B 💾 Save as example ...

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 [
2   {
3     "id": 1,
4     "name": "Iphone",
5     "price": "1000$"
6   },
7   {
8     "id": 2,
9     "name": "Laptop",
10    "price": "2000$"
```

Activate Windows  
Go to Settings to activate Windows.

### (b) Get a specific product



GET ▼ | http://localhost:4001/products/2 Send ▼

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results 🌐 Status: 200 OK Time: 7 ms Size: 275 B 💾 Save as example ...

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 {
2   "id": 2,
3   "name": "Laptop",
4   "price": "2000$"
5 }
```

### (c) Create a new product

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

- Method:** POST
- URL:** http://localhost:4001/products
- Body:** A JSON object: 

```
{ 1: { 2:   "name": "Shoe", 3:   "price": "500$" 4: }
```
- Response:** Status: 201 Created, Time: 7 ms, Size: 277 B. The response body is a JSON object: 

```
1 { 2:   "id": 3, 3:   "name": "Shoe", 4:   "price": "500$" 5: }
```

### (d) Update a product using id

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

- Method:** PUT
- URL:** http://localhost:4001/products/3
- Body:** A JSON object: 

```
1 { 2:   "name": "Shoe", 3:   "price": "700$" 4: }
```
- Response:** Status: 200 OK, Time: 6 ms, Size: 272 B. The response body is a JSON object: 

```
1 { 2:   "id": 3, 3:   "name": "Shoe", 4:   "price": "700$" 5: }
```

### (e) Deleting a specific product

DELETE ▼ http://localhost:4001/products/2 Send ▼

Params Authorization Headers (7) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

1

Body Cookies Headers (7) Test Results ⊕ Status: 200 OK Time: 5 ms Size: 277 B 📄 Save as example ⋮

Pretty Raw Preview Visualize **JSON** ▼ ≡ 📄 🔍

```
1 [
2   {
3     "id": 2,
4     "name": "Laptop",
5     "price": "2000$"
6   }
7 ]
```

### (f) Verification

GET ▼ http://localhost:4001/products Send ▼

Params Authorization Headers (7) **Body** **Scripts** ● Settings Cookies

Pre-request

Post-response \*

```
1 pm.test("Response status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5
6 pm.test("Response time is less than 200ms", function () {
7   pm.expect(pm.response.responseTime).to.be.below(200);
8 });
9
10
11 pm.test("Response has the required fields - id, name, and price",
    function () {
```

📦 Packages ➤  
[Open package library](#) →

Snippets

- Get an environment variable
- Get a global variable
- Get a variable
- Get a collection variable

Body Cookies Headers (7) **Test Results (4/5)** ⊕ Status: 200 OK Time: 24 ms Size: 277 B 📄 Save as example ⋮

All Passed Skipped Failed 🔄

- PASS** Response status code is 200
- PASS** Response time is less than 200ms
- PASS** Response has the required fields - id, name, and price
- PASS** ID is a non-negative integer

Activate Windows  
Go to Settings to activate Windows.

### 3. Students

#### (a) Get all students

GET ▼ | http://localhost:4001/students Send ▼

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 6 ms Size: 310 B Save as example ...

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 [
2   {
3     "id": 1,
4     "name": "Faizan",
5     "grade": "A"
6   },
7   {
8     "id": 2,
9     "name": "Mariya",
10    "grade": "B"
11  }
12 ]
```

Activate Windows  
Go to Settings to activate Windows.

#### (b) Get a specific student

GET ▼ | http://localhost:4001/students/1 Send ▼

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 14 ms Size: 271 B Save as example ...

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 {
2   "id": 1,
3   "name": "Faizan",
4   "grade": "A"
5 }
```

### (c) Creating a new student

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

- Method:** POST
- URL:** http://localhost:4001/students
- Body:** A JSON object: 

```
{ 1: { 2:   "name": "Sameera", 3:   "grade": "A" 4: }
```
- Response:** Status 201 Created, Time: 35 ms, Size: 277 B. The response body is shown in the "Pretty" view: 

```
1 { 2   "id": 3, 3   "name": "Sameera", 4   "grade": "A" 5 }
```

### (d) Update a student using id

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

- Method:** PUT
- URL:** http://localhost:4001/students/3
- Body:** A JSON object: 

```
1 { 2   "name": "Sameera khan", 3   "grade": "A" 4 }
```
- Response:** Status 200 OK, Time: 5 ms, Size: 277 B. The response body is shown in the "Pretty" view: 

```
1 { 2   "id": 3, 3   "name": "Sameera khan", 4   "grade": "A" 5 }
```



### (e) Deleting a student using id

DELETE ▼ http://localhost:4001/students/3 Send ▼

Params Authorization Headers (7) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

1

Body Cookies Headers (7) Test Results ⊕ Status: 200 OK Time: 6 ms Size: 279 B 📄 Save as example ⋮

Pretty Raw Preview Visualize **JSON** ▼ 🔍

```
1 [
2   {
3     "id": 3,
4     "name": "Sameera khan",
5     "grade": "A"
6   }
7 ]
```

### (f) Verification

DELETE ▼ http://localhost:4001/students/3 Send ▼

Params Authorization Headers (7) **Body** **Scripts** ● Settings Cookies

Pre-request

Post-response

```
1 pm.test("Response status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5
6 pm.test("Response time is less than 200ms", function () {
7   pm.expect(pm.response.responseTime).to.be.below(200);
8 });
9
10
```

📦 Packages ➤  
[Open package library](#)

**Snippets**  
Get an environment variable  
Get a global variable  
Get a variable

Body Cookies Headers (7) **Test Results (4/4)** ⊕ Status: 200 OK Time: 6 ms Size: 279 B 📄 Save as example ⋮

All Passed Skipped Failed 🔄

**PASS** Response status code is 200

**PASS** Response time is less than 200ms

**PASS** Response has the required fields - id, name, and grade

**PASS** Id is a non-negative integer, name is a non-empty string, and grade is a non-empty string

Activate Windows