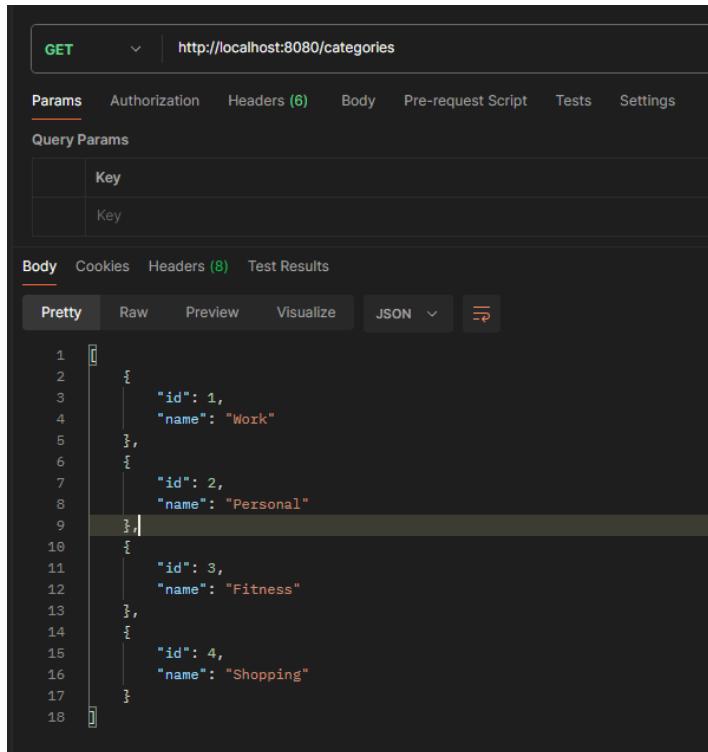


Java test

Categories API

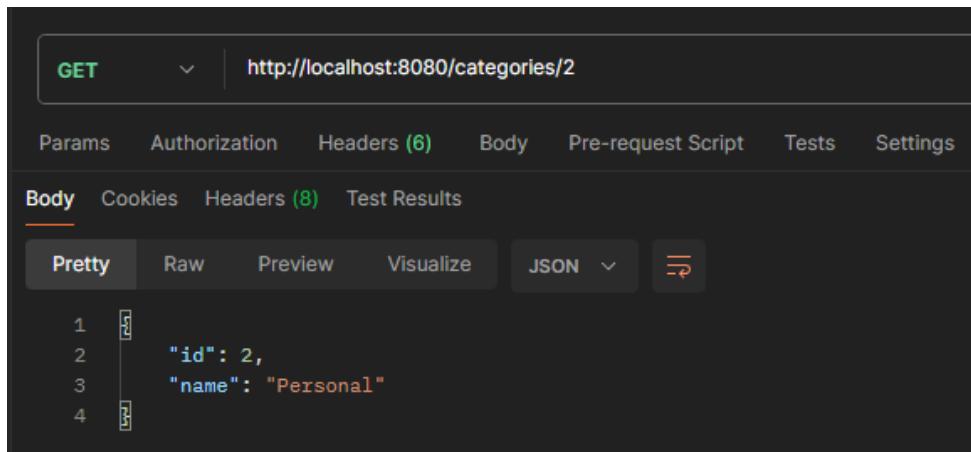
Get all categories



The screenshot shows the Postman application interface. A GET request is made to `http://localhost:8080/categories`. The response body is displayed in Pretty JSON format, showing a list of four categories:

```
1 [ ]  
2 {  
3     "id": 1,  
4     "name": "Work"  
5 },  
6 {  
7     "id": 2,  
8     "name": "Personal"  
9 },  
10 {  
11     "id": 3,  
12     "name": "Fitness"  
13 },  
14 {  
15     "id": 4,  
16     "name": "Shopping"  
17 }  
18 ]
```

Get by Id



GET <http://localhost:8080/categories/2>

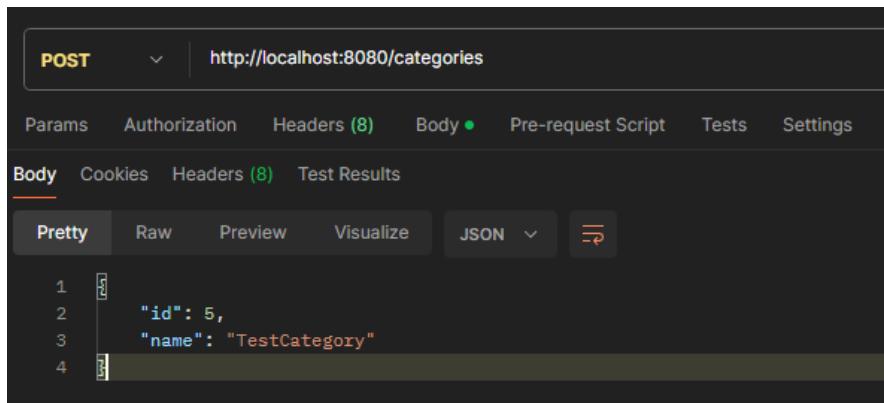
Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1  "id": 2,
2  "name": "Personal"
3
4
```

Create category



POST <http://localhost:8080/categories>

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1  "id": 5,
2  "name": "TestCategory"
3
4
```

Update category

The screenshot shows a POSTMAN interface with the following details:

- Method:** PUT
- URL:** <http://localhost:8080/categories/5>
- Body:** **JSON** (selected)
- Body Content:**

```
1 {  
2   "name": "changingNameTest"  
3 }
```
- Response Body:** **Pretty** (selected)
- Response Content:**

```
1 {  
2   "id": 5,  
3   "name": "changingNameTest"  
4 }
```

Delete category

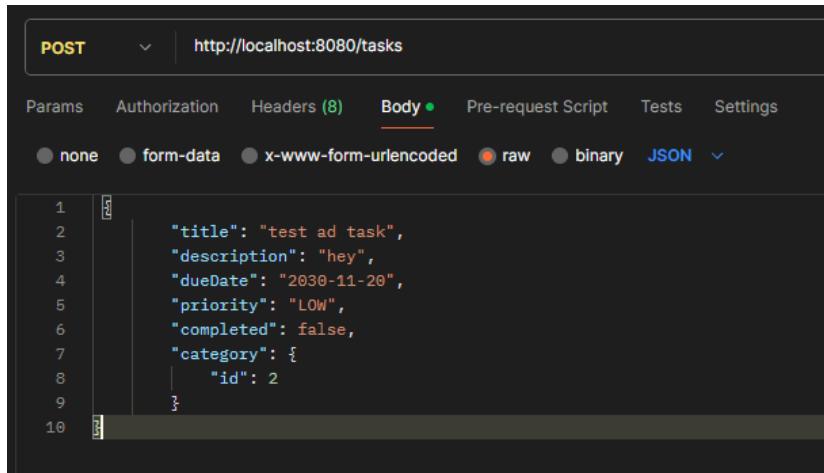
The screenshot shows a POSTMAN interface with the following details:

- Method:** DELETE
- URL:** <http://localhost:8080/categories/5>
- Body:** **Text** (selected)
- Body Content:**

```
1
```

Task API

Create new task



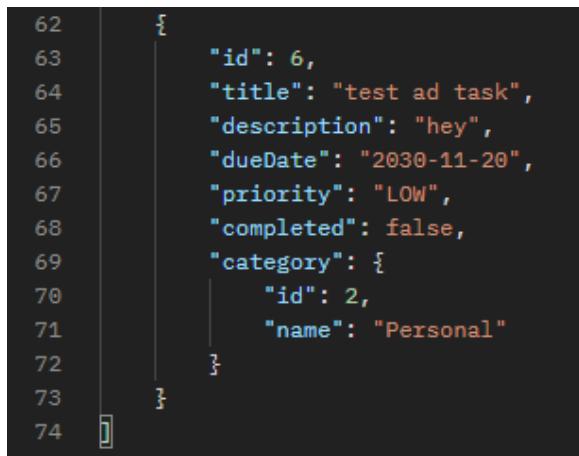
POST <http://localhost:8080/tasks>

Params Authorization Headers (8) **Body** • Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary **JSON**

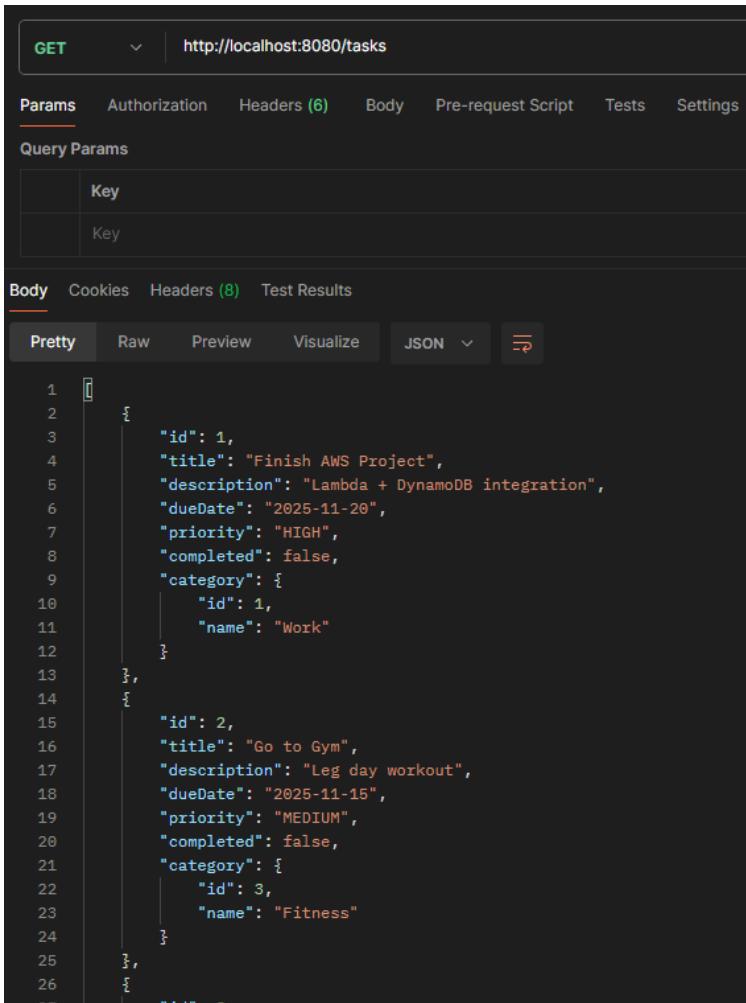
```
1 | {  
2 |     "title": "test ad task",  
3 |     "description": "hey",  
4 |     "dueDate": "2030-11-20",  
5 |     "priority": "LOW",  
6 |     "completed": false,  
7 |     "category": {  
8 |         "id": 2  
9 |     }  
10| }
```

Found using GET



```
62 | {  
63 |     "id": 6,  
64 |     "title": "test ad task",  
65 |     "description": "hey",  
66 |     "dueDate": "2030-11-20",  
67 |     "priority": "LOW",  
68 |     "completed": false,  
69 |     "category": {  
70 |         "id": 2,  
71 |         "name": "Personal"  
72 |     }  
73 | }  
74 | ]
```

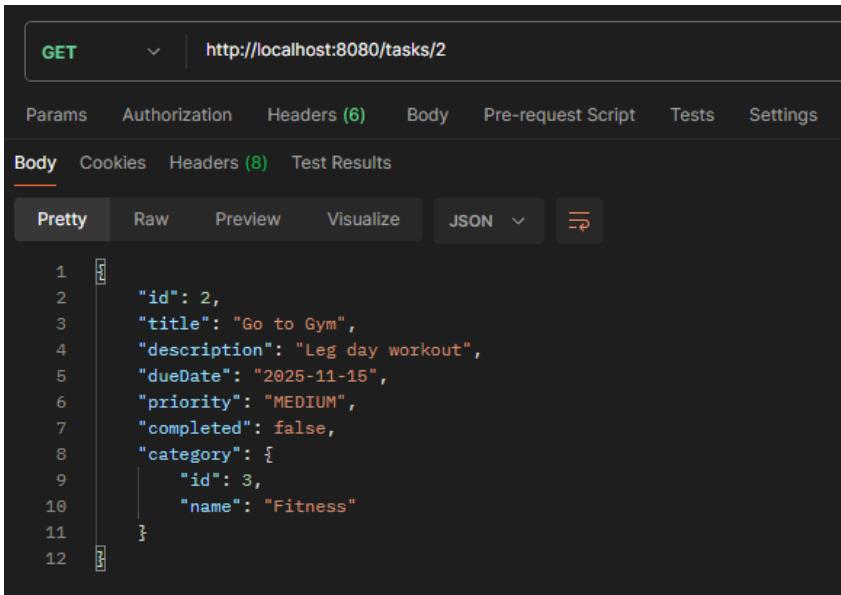
Get all tasks



The screenshot shows the Postman application interface. At the top, there is a header bar with a 'GET' button and the URL 'http://localhost:8080/tasks'. Below the header, there are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is currently selected. Under the 'Body' tab, there are sections for 'Query Params' and 'Body'. In the 'Query Params' section, there are two entries: 'Key' and 'Key'. In the 'Body' section, there are tabs for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. The 'Pretty' tab is selected. The response body is displayed as a JSON array of two objects:

```
1 [ { 2   "id": 1, 3     "title": "Finish AWS Project", 4     "description": "Lambda + DynamoDB integration", 5     "dueDate": "2025-11-20", 6     "priority": "HIGH", 7     "completed": false, 8     "category": { 9       "id": 1, 10      "name": "Work" 11    } 12  }, 13  { 14    "id": 2, 15    "title": "Go to Gym", 16    "description": "Leg day workout", 17    "dueDate": "2025-11-15", 18    "priority": "MEDIUM", 19    "completed": false, 20    "category": { 21      "id": 3, 22      "name": "Fitness" 23    } 24  }, 25  { 26    "id": 3, 27    "name": "Workout Session" 28  } ]
```

Get task by ID



GET <http://localhost:8080/tasks/2>

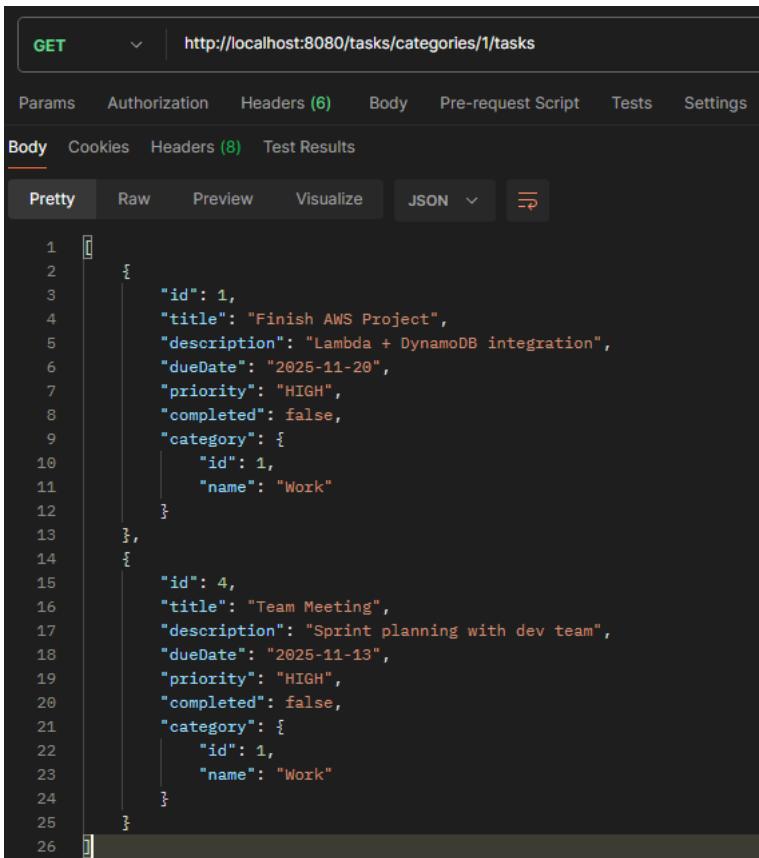
Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "title": "Go to Gym",
4   "description": "Leg day workout",
5   "dueDate": "2025-11-15",
6   "priority": "MEDIUM",
7   "completed": false,
8   "category": {
9     "id": 3,
10    "name": "Fitness"
11  }
12 }
```

Get tasks by category



GET <http://localhost:8080/tasks/categories/1/tasks>

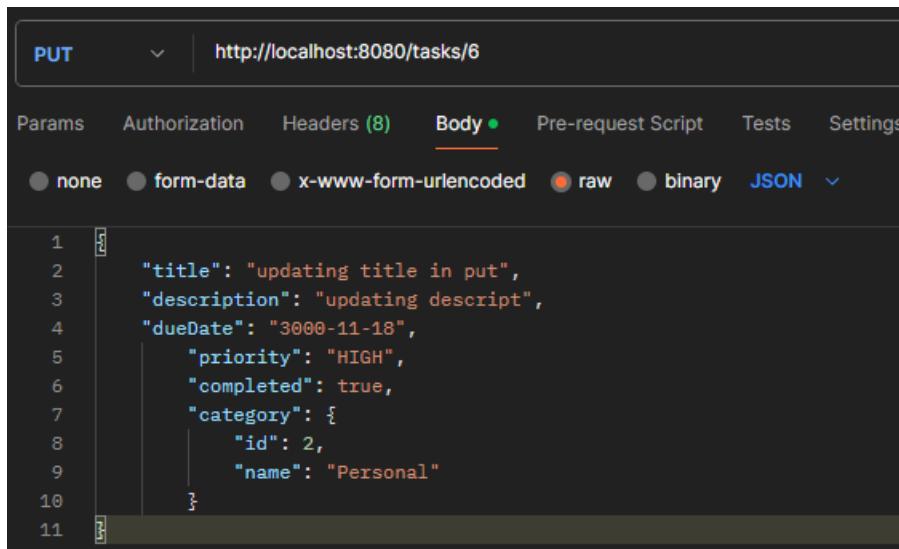
Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "title": "Finish AWS Project",
5     "description": "Lambda + DynamoDB integration",
6     "dueDate": "2025-11-20",
7     "priority": "HIGH",
8     "completed": false,
9     "category": {
10       "id": 1,
11       "name": "Work"
12     }
13   },
14   {
15     "id": 4,
16     "title": "Team Meeting",
17     "description": "Sprint planning with dev team",
18     "dueDate": "2025-11-13",
19     "priority": "HIGH",
20     "completed": false,
21     "category": {
22       "id": 1,
23       "name": "Work"
24     }
25   }
26 ]
```

Update task



A screenshot of the Postman application interface. The request method is set to **PUT** and the URL is **http://localhost:8080/tasks/6**. The **Body** tab is selected, showing a JSON payload with the following content:

```
1
2   "title": "updating title in put",
3   "description": "updating descript",
4   "dueDate": "3000-11-18",
5     "priority": "HIGH",
6     "completed": true,
7     "category": {
8       "id": 2,
9       "name": "Personal"
10    }
11 }
```

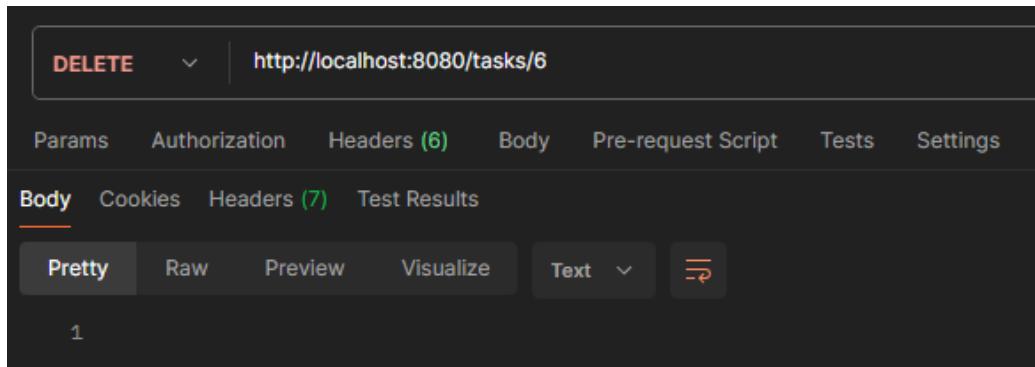
Then using GET



A screenshot of the Postman application interface. The request method is set to **GET** and the URL is **http://localhost:8080/tasks/6**. The response body is displayed as follows:

```
{
  "id": 6,
  "title": "updating title in put",
  "description": "updating descript",
  "dueDate": "3000-11-18",
  "priority": "HIGH",
  "completed": true,
  "category": {
    "id": 2,
    "name": "Personal"
  }
}
```

Delete task



A screenshot of the Postman application interface. The request method is set to **DELETE** and the URL is **http://localhost:8080/tasks/6**. The response body is empty, containing only the number **1**.

Its gone 😊

Scala test

Categories API

GET ALL - ✓
Get by id - ✓
Create - ✓
Update - ✓
Delete - ✓

Task API

Get all - ✓
Get by id - ✓
Create - ✓
Update - ✓
Delete - ✓
Get tasks by category - ✓