

Categories

1.POST

The screenshot shows the Postman interface with a POST request to `https://localhost:7271/api/Categories`. The request body is a JSON object:

```
1 {  
2   "id": 0,  
3   "name": "Thriller"  
4 }
```

The response status is `201 Created`, and the response body is:

```
{ } JSON ▾
```

```
1 {  
2   "id": 1,  
3   "name": "Thriller",  
4   "books": null  
5 }
```

2.GET

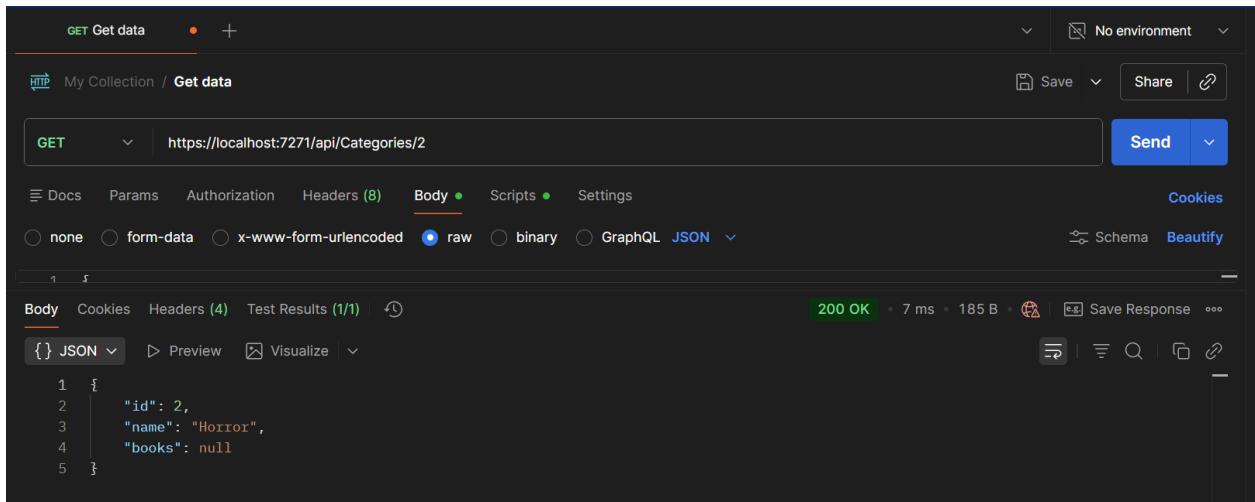
The screenshot shows the Postman interface with a GET request to `https://localhost:7271/api/Categories`. The response status is `200 OK`, and the response body is:

```
200 OK ▾ 11 ms ▾ 227 B ▾
```

```
{} JSON ▾
```

```
1 [  
2   {  
3     "id": 1,  
4     "name": "Thriller",  
5     "books": null  
6   },  
7   {  
8     "id": 2,  
9     "name": "Horror",  
10    "books": null  
11  }  
12 ]
```

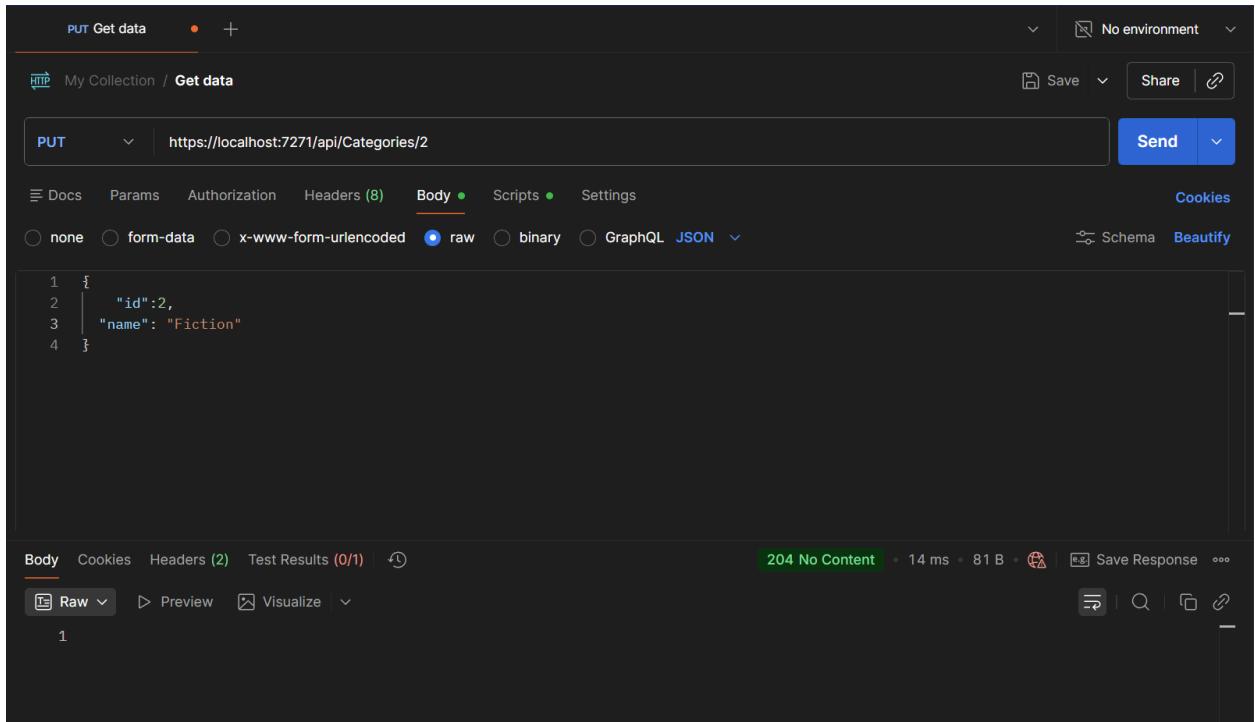
3. GET BY ID



The screenshot shows the Postman interface with a collection named "My Collection". A new request is being made to the URL `https://localhost:7271/api/Categories/2`. The method is set to GET. The request body is currently empty, with the option "raw" selected. The response status is 200 OK, indicating success. The response body contains the following JSON data:

```
1 {  
2     "id": 2,  
3     "name": "Horror",  
4     "books": null  
5 }
```

4.PUT



The screenshot shows the Postman interface with a collection named "My Collection". A new request is being made to the URL `https://localhost:7271/api/Categories/2`. The method is set to PUT. The request body contains the following JSON data:

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

The response status is 204 No Content, indicating that the data was successfully updated. The response body is empty.

5.DELETE

The screenshot shows the Postman interface with a DELETE request to `https://localhost:7271/api/Categories/2`. The response is a 204 No Content status with a duration of 153 ms and a size of 81 B. The raw response body is empty.

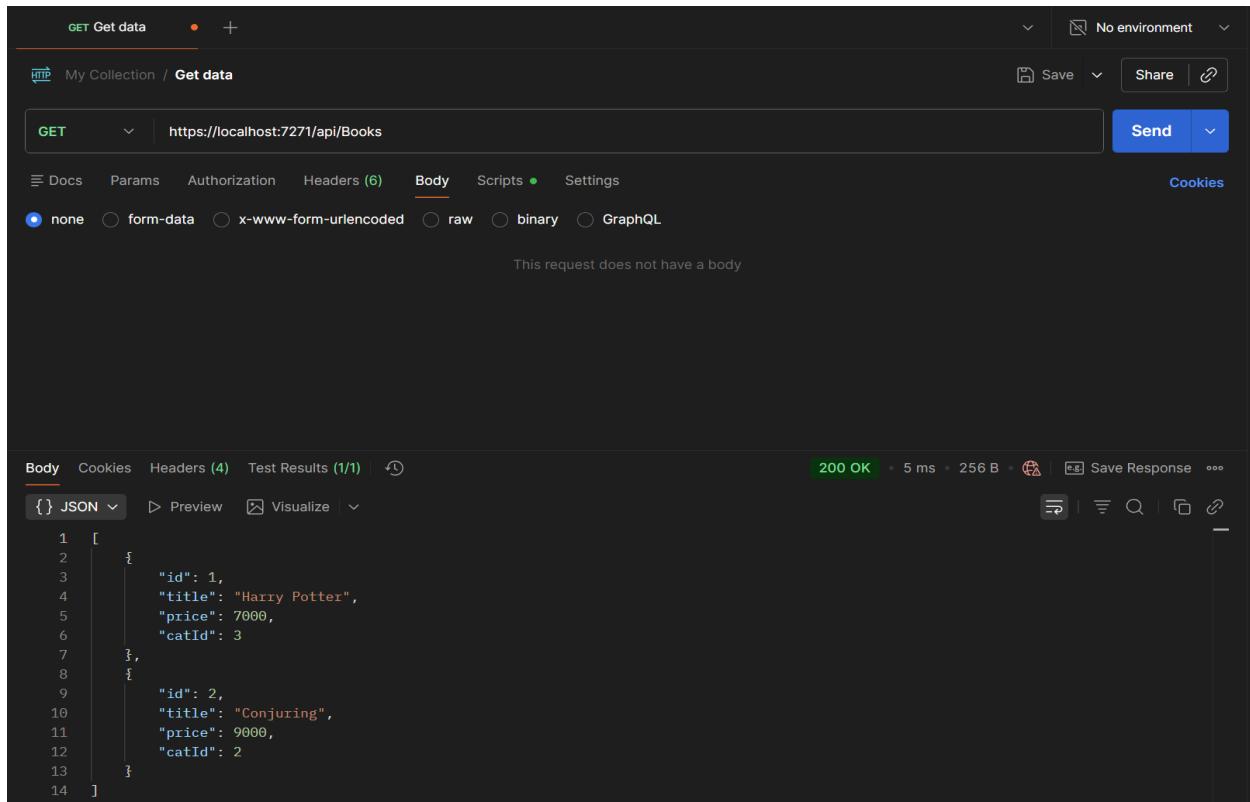
Books

1.POST

The screenshot shows the Postman interface with a POST request to `https://localhost:7271/api/Books`. The response is a 201 Created status with a duration of 70 ms and a size of 253 B. The raw response body is a JSON object representing a new book entry.

```
[{"id": 1, "title": "Harry Potter", "price": 7000, "catId": 3}]
```

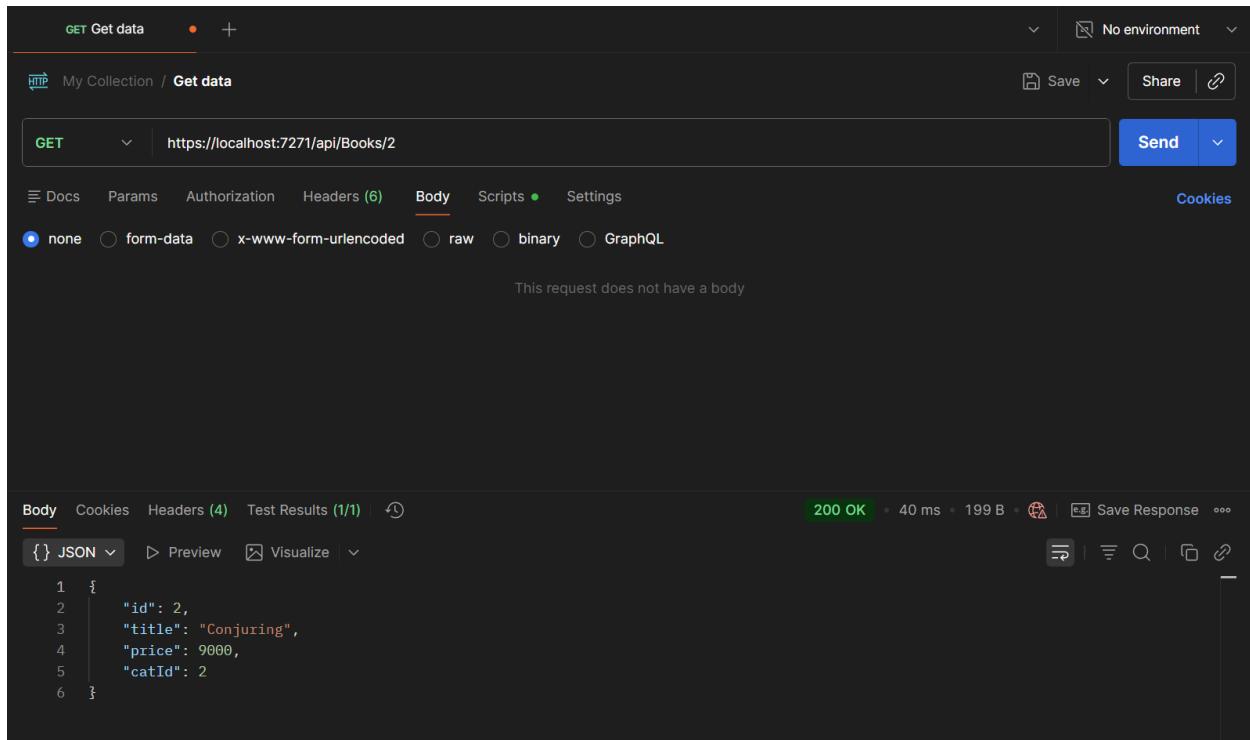
2.GET



The screenshot shows the Postman interface with a collection named "My Collection" and a "Get data" endpoint. The request URL is `https://localhost:7271/api/Books`. The response status is 200 OK, with a response time of 5 ms, a response size of 256 B, and a save response option. The response body is displayed as JSON:

```
1 [  
2   {  
3     "id": 1,  
4     "title": "Harry Potter",  
5     "price": 7000,  
6     "catId": 3  
7   },  
8   {  
9     "id": 2,  
10    "title": "Conjuring",  
11    "price": 9000,  
12    "catId": 2  
13  }  
14 ]
```

3.GET BY ID



The screenshot shows the Postman interface with a collection named "My Collection" and a "Get data" endpoint. The request URL is `https://localhost:7271/api/Books/2`. The response status is 200 OK, with a response time of 40 ms, a response size of 199 B, and a save response option. The response body is displayed as JSON:

```
1 {  
2   "id": 2,  
3   "title": "Conjuring",  
4   "price": 9000,  
5   "catId": 2  
6 }
```

4.PUT

The screenshot shows the Postman interface with a PUT request to `https://localhost:7271/api/Books/2`. The request body is a JSON object:

```
1 {
2   "id": 2,
3   "title": "Conjuring-2",
4   "price": 9000,
5   "catId": 2
6 }
```

The response status is 204 No Content, with a response time of 30 ms and a response size of 81 B.

5.GET (Fetching Grouped-by-category)

The screenshot shows the Postman interface with a GET request to `https://localhost:7271/api/Books/grouped-by-category`. The response body is a JSON object:

```
1 [
2   {
3     "category": "Horror",
4     "books": [
5       {
6         "id": 2,
7         "title": "Conjuring-2",
8         "price": 9000
9       },
10      {
11        "id": 3,
12        "title": "Anabelle",
13        "price": 9500
14      }
15    ],
16    {
17      "category": "Thriller",
18      "books": [
19        {
20          "id": 4,
21          "title": "The Silent Patient",
22          "price": 9500
23        }
24      ]
25    }
]
```

The response status is 200 OK, with a response time of 12 ms and a response size of 351 B.

5.DELETE

The screenshot shows the Postman application interface. At the top, there's a header bar with tabs for "DEL Get data" and a "+" button. To the right of the header are buttons for "Save", "Share", and a copy icon. Below the header, the URL "https://localhost:7271/api/Books/4" is entered into the "DELETE" method field. The main workspace is titled "Get data". On the left, there are tabs for "Docs", "Params", "Authorization", "Headers (8)", "Body", "Scripts", and "Settings". The "Body" tab is currently selected and has a green dot next to it. Below the tabs, there are radio buttons for "none", "form-data", "x-www-form-urlencoded", "raw" (which is selected), "binary", "GraphQL", and "JSON". To the right of these buttons are "Cookies", "Schema", and "Beautify" buttons. At the bottom of the workspace, there are buttons for "Raw", "Preview", and "Visualize". The status bar at the bottom right shows "204 No Content", "13 ms", "81 B", and a refresh icon. There are also "Save Response" and "..." buttons.