

## Create Blog Data using POST:

The screenshot shows the Postman interface with a POST request to `https://localhost:7003/api/Blogs`. The request body is set to `JSON` and contains the following JSON:

```
1 {
2   "title": "My First Blog",
3   "content": "Learning ASP.NET Core is fun"
4 }
```

The response status is `201 Created` with a response time of `370 ms` and a size of `288 B`.

## Read Blog Data using GET:

The screenshot shows the Postman interface with a GET request to `https://localhost:7003/api/Blogs`. The request body is set to `JSON` and contains the same JSON as the POST request:

```
1 {
2   "title": "My First Blog",
3   "content": "Learning ASP.NET Core is fun"
4 }
```

The response status is `200 OK` with a response time of `122 ms` and a size of `239 B`.

## Read Blog Data by ID using GET:

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** <https://localhost:7003/api/Blogs/2>
- Body:** Raw JSON response (Beautified):

```
1 {
2   "title": "My Second Blog",
3   "content": "Learning ASP.NET Core Basics"
4 }
```

- Test Results:** Status: 200 OK, Time: 148 ms, Size: 238 B
- Body (Raw JSON):** Raw JSON response (Beautified):

```
1 {
2   "id": 2,
3   "title": "My Second Blog",
4   "content": "Learning ASP.NET Core Basics",
5   "comments": null
6 }
```

## Delete Blog Data:

The screenshot shows the Postman interface with the following details:

- Method:** DELETE
- URL:** <https://localhost:7003/api/Blogs/2>
- Body:** Raw JSON response (Beautified):

```
1 √{
2   "title": "My Second Blog",
3   "content": "Learning ASP.NET Core Basics"
4 }
```

- Test Results:** Status: 204 No Content, Time: 130 ms, Size: 81 B
- Body (Raw JSON):** Raw JSON response (Beautified):

```
1
```

## Create a comment:

The screenshot shows the Postman interface for creating a comment. The URL is <https://localhost:7003/api/Comments>. The method is set to **POST**. The **Body** tab is selected, showing the following JSON payload:

```
1 {
2   "message": "Great article",
3   "author": "Abhi",
4   "blogId": 1
5 }
```

The response status is **201 Created** with a response time of 84 ms and a body size of 263 B. The response JSON is identical to the sent payload.

## Show all comment:

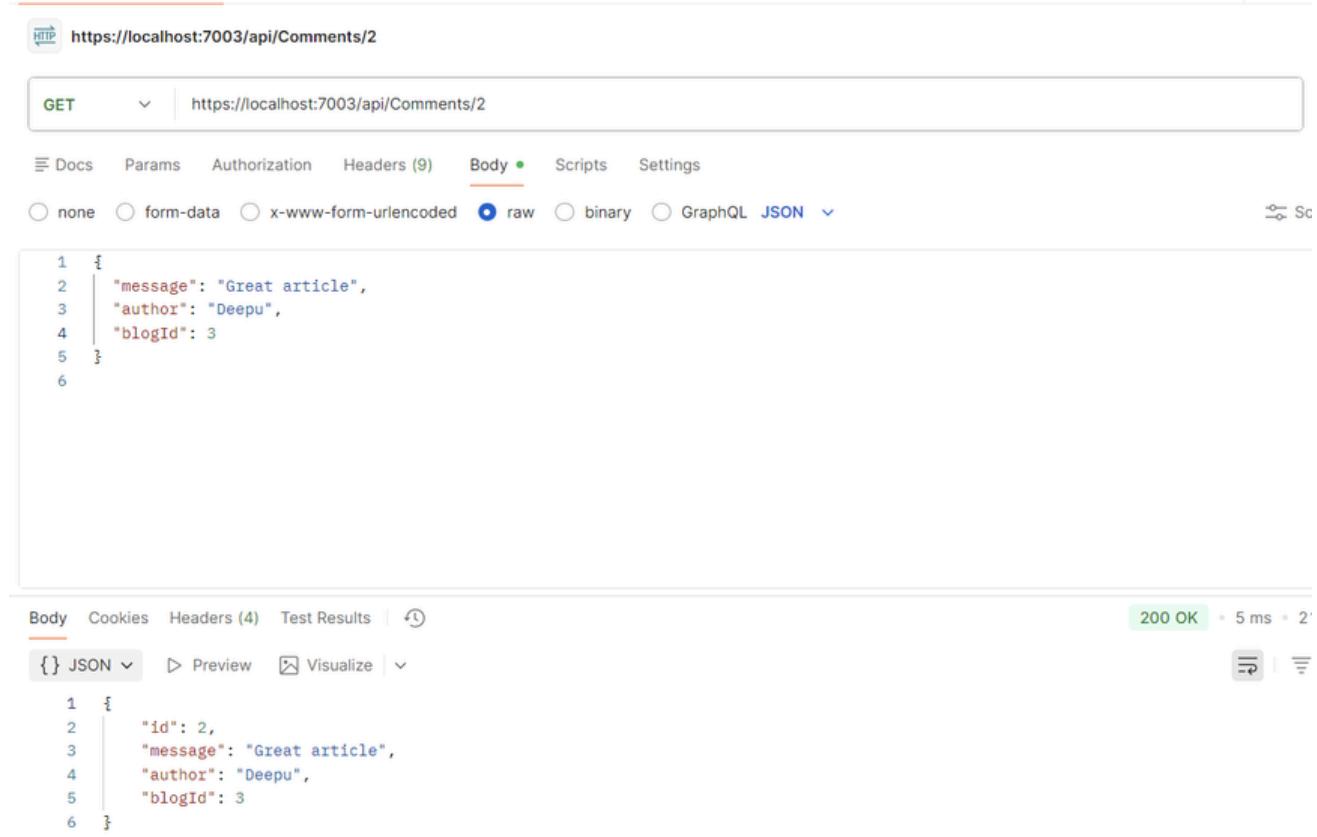
The screenshot shows the Postman interface for listing comments. The URL is <https://localhost:7003/api/Comments>. The method is set to **GET**. The **Body** tab is selected, showing the following JSON payload:

```
1 {
2   "message": "Great article",
3   "author": "Deepu",
4   "blogId": 3
5 }
```

The response status is **200 OK** with a response time of 28 ms and a body size of 274 B. The response JSON contains two comments:

```
1 [
2   {
3     "id": 1,
4     "message": "Great article",
5     "author": "Abhi",
6     "blogId": 1
7   },
8   {
9     "id": 2,
10    "message": "Great article",
11    "author": "Deepu",
12    "blogId": 3
13  }
]
```

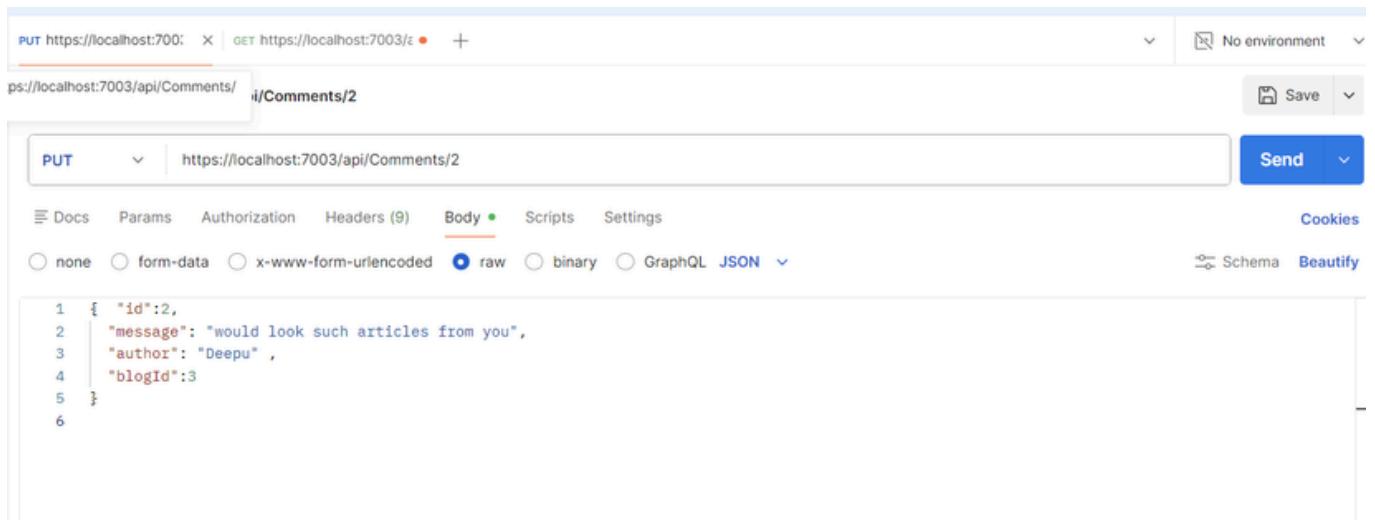
## Show comment by ID:



The screenshot shows a Postman request to `https://localhost:7003/api/Comments/2`. The method is `GET`. The response status is `200 OK` with a response time of 5 ms and a total duration of 2'. The response body is a JSON object:

```
1 {  
2   "id": 2,  
3   "message": "Great article",  
4   "author": "Deepu",  
5   "blogId": 3  
6 }
```

## Update comment by put:



The screenshot shows a Postman request to `https://localhost:7003/api/Comments/2` using the `PUT` method. The response status is `200 OK` with a response time of 5 ms and a total duration of 2'. The response body is a JSON object:

```
1 {  
2   "id": 2,  
3   "message": "would look such articles from you",  
4   "author": "Deepu",  
5   "blogId": 3  
6 }
```

Delete comment:

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

- HTTP Method:** `DELETE`
- URL:** `https://localhost:7003/api/Comments/2`
- Headers:** (7) (selected tab)
- Body:** (selected tab) contains the value `1`.
- Body Content Type:** `raw` (selected)
- Response Status:** `204 No Content`
- Response Time:** `13 ms`
- Response Size:** `81 B`
- Body Content:** `1`