

Create Blog Data using POST:

The screenshot shows a REST client interface with the URL `https://localhost:7003/api/Blogs`. The method is set to **POST**. The request body is a JSON object:

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

. The response status is **201 Created** with a response time of 370 ms and a body size of 288 B. The response body is a JSON object:

```
{  "id": 1,  "title": "My First Blog",  "content": "Learning ASP.NET Core is fun",  "comments": null}
```

URL: `https://localhost:7003/api/Blogs`

Method: **POST**

Body (raw):

```
1 {
2   "title": "My First Blog",
3   "content": "Learning ASP.NET Core is fun"
4 }
5
```

Response: **201 Created** • 370 ms • 288 B

Body (JSON):

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

Read Blog Data using GET:

The screenshot shows a REST client interface with the URL `https://localhost:7003/api/Blogs`. The method is set to **GET**. The response status is **200 OK** with a response time of 122 ms and a body size of 239 B. The response body is a JSON array containing one object:

```
[  {    "id": 1,    "title": "My First Blog",    "content": "Learning ASP.NET Core is fun",    "comments": null  }]
```

URL: `https://localhost:7003/api/Blogs`

Method: **GET**

Body (raw):

```
1 {
2   "title": "My First Blog",
3   "content": "Learning ASP.NET Core is fun"
4 }
5
```

Response: **200 OK** • 122 ms • 239 B

Body (JSON):

```
1 [
2   {
3     "id": 1,
4     "title": "My First Blog",
5     "content": "Learning ASP.NET Core is fun",
6     "comments": null
7   }
8 ]
```

Read Blog Data by ID using GET:

Postman interface showing a GET request to `https://localhost:7003/api/Blogs/2`. The response is a JSON object:

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

The status bar indicates a **200 OK** response with a time of 148 ms and a body size of 238 B.

Below the main response, the **Body** tab shows the JSON data:

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

Delete Blog Data:

Postman interface showing a DELETE request to `https://localhost:7003/api/Blogs/2`. The response is **204 No Content** with a time of 130 ms and a body size of 81 B.

The **Body** tab shows the raw response:

```
1
```

Create a comment:

Postman interface showing a POST request to `https://localhost:7003/api/Comments`. The request is configured with the following details:

- Method: POST
- URL: `https://localhost:7003/api/Comments`
- Body: raw (selected)
- Body content (JSON):

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

The response is displayed in the Body tab, showing a 201 Created status with a response time of 84 ms and a body size of 263 B. The response body is a JSON object:

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

Show all comment:

Postman interface showing a GET request to `https://localhost:7003/api/Comments`. The request is configured with the following details:

- Method: GET
- URL: `https://localhost:7003/api/Comments`
- Body: raw (selected)

The response is displayed in the Body tab, showing a 200 OK status with a response time of 28 ms and a body size of 274 B. The response body is a JSON array of two comment objects:

```
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  }
14 ]
```

Show comment by ID:

https://localhost:7003/api/Comments/2

GEThttps://localhost:7003/api/Comments/2

DocsParamsAuthorizationHeaders (9)BodyScriptsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

Sc

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

BodyCookiesHeaders (4)Test Results

200 OK5 ms2'

{ }JSONPreviewVisualize

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

Update comment by put:

PUT https://localhost:7003/api/Comments/2

PUThttps://localhost:7003/api/Comments/2

DocsParamsAuthorizationHeaders (9)BodyScriptsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

Schema Beautify

```
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:

- URL:** `https://localhost:7003/api/Comments/2`
- Method:** `DELETE`
- Body:** Empty (indicated by a tab labeled "Body" and a text area with a single line of text: `1 | ⌘+Alt+P to Ask AI`)
- Response:** `204 No Content` (13 ms, 81 B)
- Headers:** (7) (tab labeled "Headers (7)")
- Params:** (tab labeled "Params")
- Authorization:** (tab labeled "Authorization")
- Scripts:** (tab labeled "Scripts")
- Settings:** (tab labeled "Settings")
- Body Type:** `raw` (selected), `none`, `form-data`, `x-www-form-urlencoded`, `binary`, `GraphQL`, `JSON`
- Schema:** (tab labeled "Schema")
- Raw:** (tab labeled "Raw")
- Preview:** (tab labeled "Preview")
- Visualize:** (tab labeled "Visualize")