

Github Link : <https://github.com/febbrystbl/java-enterprise-uas>

The screenshot shows the Postman REST client interface. On the left, a sidebar lists collections, with 'Java Enterprise' expanded to show a list of endpoints including 'POST Add Category'. The main workspace displays the details for the 'POST Add Category' endpoint at 'localhost:8080/api/categories'. The request method is 'POST', and the body is a JSON object:

```
{
  "name": "fakultas teknologi infomedia"
}
```

. The response tab is selected, showing a '200 OK' status with a response time of 21 ms and a body of

```
{
  "id": 3,
  "name": "fakultas teknologi infomedia",
  "books": []
}
```

. The bottom status bar indicates the application is running on 'Postman'.

The screenshot displays the Java Enterprise IDE interface. On the left, a sidebar shows a collection of endpoints under 'Java Enterprise'. The main area is titled 'Java Enterprise | Get Category'. It features a request configuration section with a 'GET' method and a URL 'localhost8080/api/categories'. Below this, there's a 'Query Params' table with columns 'Key' and 'Value'. The response view shows a '200 OK' status, a response time of '61 ms', and a response size of '46.4 KB'. The response body is a JSON array of two category objects, each with 'id', 'name', and 'books' fields.

```

{
  "id": 1,
  "name": "Fakultas teknologi informasi",
  "books": []
},
{
  "id": 2,
  "name": "Fakultas teknologi infomedia",
  "books": [
    {
      "id": 1,
      "title": "bagaimana membuat media trending",
      "author": "dede inoen",
      "category": {
        "id": 2,
        "name": "Fakultas teknologi infomedia",
        "books": [
          {
            "id": 1,
            "title": "bagaimana membuat media trending",
            "author": "dede inoen",
            "category": {
              "id": 1,
              "name": "Fakultas teknologi informasi",
              "books": []
            }
          }
        ]
      }
    }
  ]
}

```

3. Update Category

The screenshot shows the Postman interface for a PUT request to `localhost:8080/api/categories/2`. The request is successful, returning a 200 OK status. The response body is a JSON object representing a category with its associated books.

Request:

- Method: PUT
- URL: `localhost:8080/api/categories/2`

Query Params:

Key	Value	Description
Key	Value	Description

Response Body (JSON):

```
1 {
2   "id": 2,
3   "name": "fakultas teknologi infomedia revisi 1",
4   "books": [
5     {
6       "id": 1,
7       "title": "bagaimana membuat media trending",
8       "author": "dede inoen",
9       "category": {
10        "id": 2,
11        "name": "fakultas teknologi infomedia revisi 1",
12        "books": [
13          {
14            "id": 1,
15            "title": "bagaimana membuat media trending",
16            "author": "dede inoen",
17            "category": {
18              "id": 2,
```

4. Get Category by ID

The screenshot shows the Postman interface for a GET request to `localhost:8080/api/categories/2`. The request is successful, returning a 200 OK status. The response body is a JSON object representing a category with its associated books.

Request:

- Method: GET
- URL: `localhost:8080/api/categories/2`

Query Params:

Key	Value	Description
Key	Value	Description

Response Body (JSON):

```
1 {
2   "id": 2,
3   "name": "fakultas teknologi infomedia revisi 1",
4   "books": [
5     {
6       "id": 1,
7       "title": "bagaimana membuat media trending topic revisi 1",
8       "author": "dede inoen",
9       "category": {
10        "id": 2,
11        "name": "fakultas teknologi infomedia revisi 1",
12        "books": [
13          {
14            "id": 1,
15            "title": "bagaimana membuat media trending topic revisi 1",
16            "author": "dede inoen",
17            "category": {
18              "id": 2,
19              "name": "fakultas teknologi infomedia revisi 1",
20              "books": [
21                {
22                  "id": 1,
23                  "title": "bagaimana membuat media trending topic revisi 1",
24                  "author": "dede inoen",
```

5. Delete Category

The screenshot shows the Postman interface for a DELETE request. The request is sent to `localhost:8080/api/categories/2`. The response is a `200 OK` status with a response time of 15 ms and a body size of 123 B. The response body is empty, indicating a successful deletion.

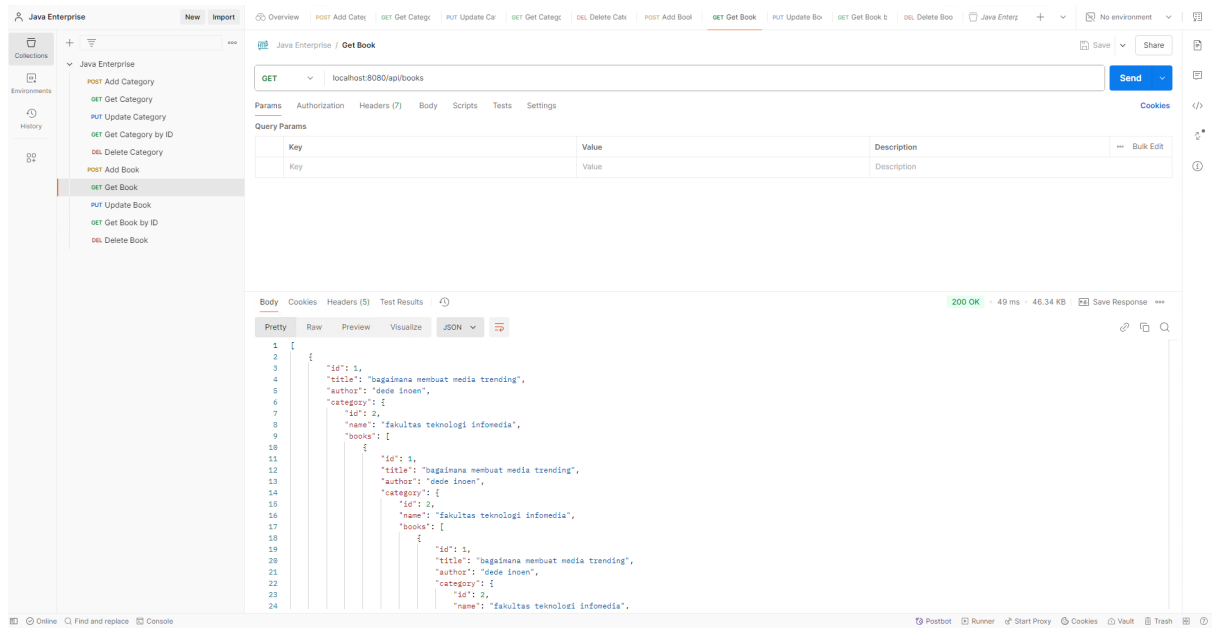
Key	Value	Description
Key	Value	Description

6. Add Book

The screenshot shows the Postman interface for a POST request. The request is sent to `localhost:8080/api/books`. The response is a `200 OK` status with a response time of 23 ms and a body size of 280 B. The response body is a JSON object representing a book.

```
1 {
2   "id": 3,
3   "title": "bagaimana membuat media trending",
4   "author": "dede inoem",
5   "category": {
6     "id": 2,
7     "name": null,
8     "books": []
9   }
10 }
```

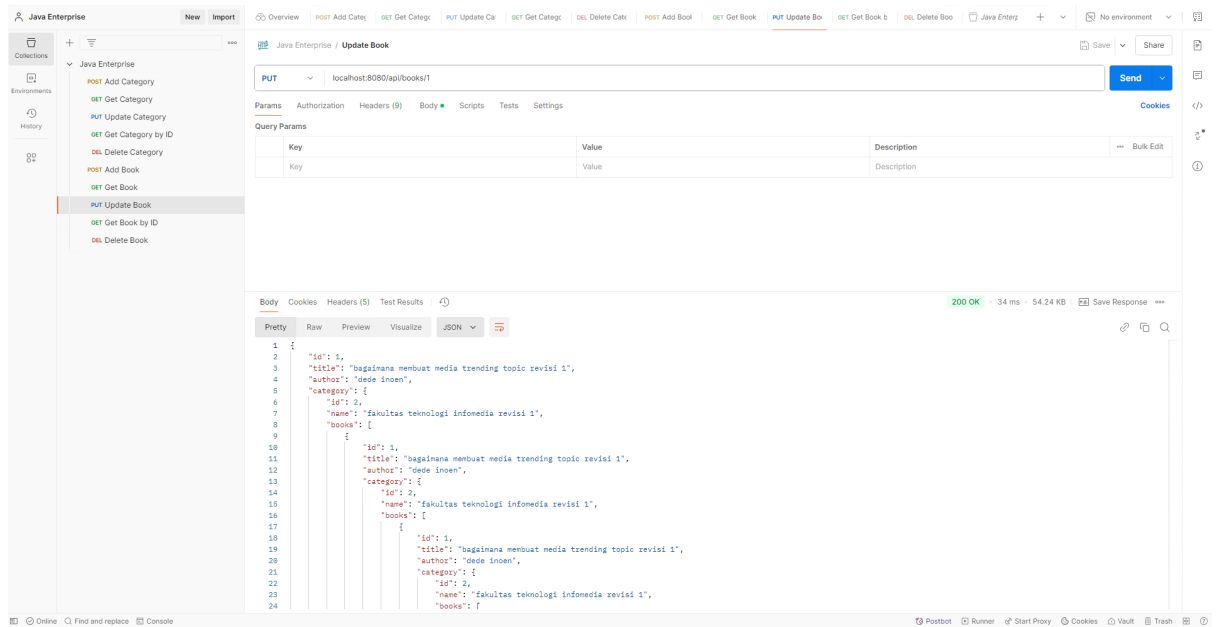
7. Get Book



The screenshot shows the Postman interface for a GET request to `localhost:8080/api/books`. The response is a 200 OK status with a 49 ms response time and a 46.34 KB body. The response body is displayed in JSON format, showing an array of book objects.

```
1 {
2   {
3     "id": 1,
4     "title": "bagaimana membuat media trending",
5     "author": "dede inoen",
6     "category": {
7       "id": 2,
8       "name": "fakultas teknologi infomedia",
9       "books": [
10        {
11          "id": 1,
12          "title": "bagaimana membuat media trending",
13          "author": "dede inoen",
14          "category": {
15            "id": 2,
16            "name": "fakultas teknologi infomedia",
17            "books": [
18              {
19                "id": 1,
20                "title": "bagaimana membuat media trending",
21                "author": "dede inoen",
22                "category": {
23                  "id": 2,
24                  "name": "fakultas teknologi infomedia",
```

8. Update Book



The screenshot shows the Postman interface for a PUT request to `localhost:8080/api/books/1`. The response is a 200 OK status with a 34 ms response time and a 54.24 KB body. The response body is displayed in JSON format, showing an array of book objects, where the first book is updated.

```
1 {
2   {
3     "id": 1,
4     "title": "bagaimana membuat media trending topic revisi 1",
5     "author": "dede inoen",
6     "category": {
7       "id": 2,
8       "name": "fakultas teknologi infomedia revisi 1",
9       "books": [
10        {
11          "id": 1,
12          "title": "bagaimana membuat media trending topic revisi 1",
13          "author": "dede inoen",
14          "category": {
15            "id": 2,
16            "name": "fakultas teknologi infomedia revisi 1",
17            "books": [
18              {
19                "id": 1,
20                "title": "bagaimana membuat media trending topic revisi 1",
21                "author": "dede inoen",
22                "category": {
23                  "id": 2,
24                  "name": "fakultas teknologi infomedia revisi 1",
```

9. Get Book by ID

The screenshot shows the Postman interface for a REST client. The collection is named "Java Enterprise". The selected request is "GET Get Book by ID". The URL is "localhost:8080/api/books/1". The response status is "200 OK" with a response time of 27 ms and a size of 54.24 KB. The response body is displayed in JSON format, showing a list of books. The first book in the list has an ID of 1, a title of "bagaimana membuat media trending topic revisi 1", an author of "dede inoen", and a category of "fakultas teknologi infomedia revisi 1".

GET localhost:8080/api/books/1

Params Authorization Headers (7) Body Scripts Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results

200 OK · 27 ms · 54.24 KB

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "title": "bagaimana membuat media trending topic revisi 1",
4   "author": "dede inoen",
5   "category": {
6     "id": 2,
7     "name": "fakultas teknologi infomedia revisi 1",
8     "books": [
9       {
10        "id": 1,
11        "title": "bagaimana membuat media trending topic revisi 1",
12        "author": "dede inoen",
13        "category": {
14          "id": 2,
15          "name": "fakultas teknologi infomedia revisi 1",
16          "books": [
17            {
18              "id": 1,
19              "title": "bagaimana membuat media trending topic revisi 1",
20              "author": "dede inoen",
21              "category": {
22                "id": 2,
23                "name": "fakultas teknologi infomedia revisi 1",
24                "books": [
```

10. Delete Book

The screenshot shows the Postman interface for a REST client. The collection is named "Java Enterprise". The selected request is "DELETE Delete Book". The URL is "localhost:8080/api/books/1". The response status is "200 OK" with a response time of 41 ms and a size of 123 B. The response body is displayed in JSON format, showing a single object with an ID of 1.

DELETE localhost:8080/api/books/1

Params Authorization Headers (7) Body Scripts Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (4) Test Results

200 OK · 41 ms · 123 B

Pretty Raw Preview Visualize Text

```
1 {
```