

VKR Book Store

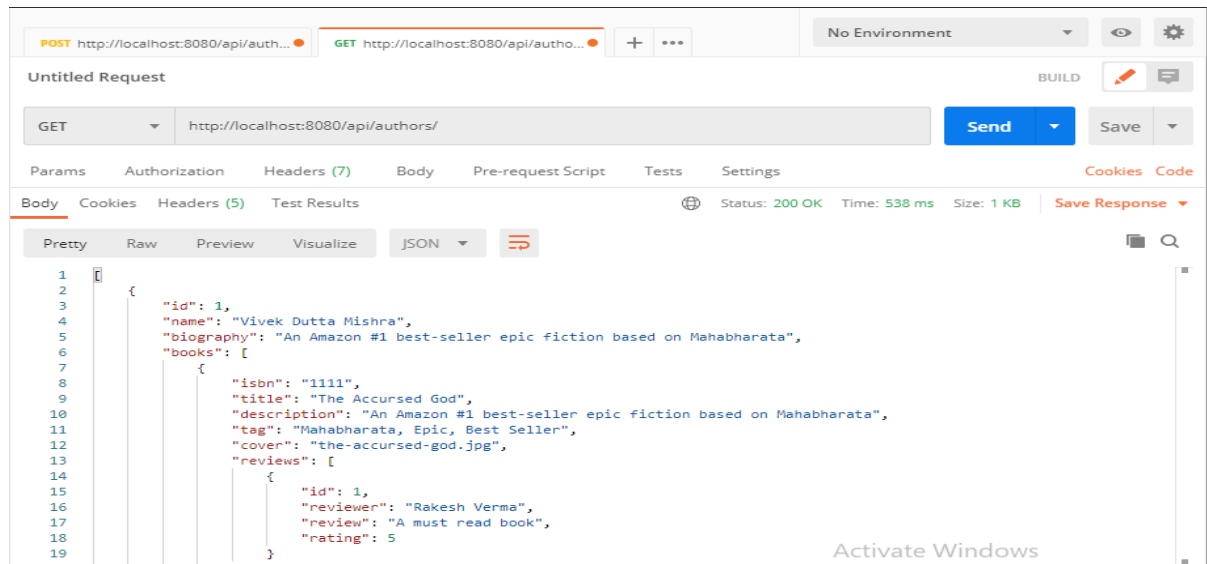
1. Below are the APIs to manage Authors

i. List of all Author(<http://localhost:8080/api/authors/>)

Response:{

```
"id": 1,
"name": "Vivek Dutta Mishra",
"biography": "An Amazon #1 best-seller epic fiction based on Mahabharata",
"books": [
  {
    "isbn": "1111",
    "title": "The Accursed God",
    "description": "An Amazon #1 best-
seller epic fiction based on Mahabharata",
    "tag": "Mahabharata, Epic, Best Seller",
    "cover": "the-accursed-god.jpg",
    "price": 250.0,
    "reviews": [
      {
        "id": 1,
        "reviewer": "Rakesh Verma",
        "review": "A must read book",
        "rating": 5
      }
    ]
  }
],
"photograph": "the-accursed-god.jpg"
},
{
  "id": 2,
  "name": "Mahatma Gandhi",
  "biography": "An auto biography of the greatest man of the century",
  "books": [
    {
      "isbn": "2222",
      "title": "My Experiments with Truth",
      "description": "An auto biography of the greatest man of the century",
      "tag": "Mahatma Gandhi, autobiography",
      "cover": "my-experiments-with-truth.jpg",
      "price": 350.0,
      "reviews": [
        {
          "id": 2,
          "reviewer": "Manish Pal",
          "review": "Amazing book",
          "rating": 4
        }
      ]
    }
  ],
  "photograph": "my-experiments-with-truth.jpg"
}
```

]



ii. To create an Author(<http://localhost:8080/api/authors/>)

Request body: {

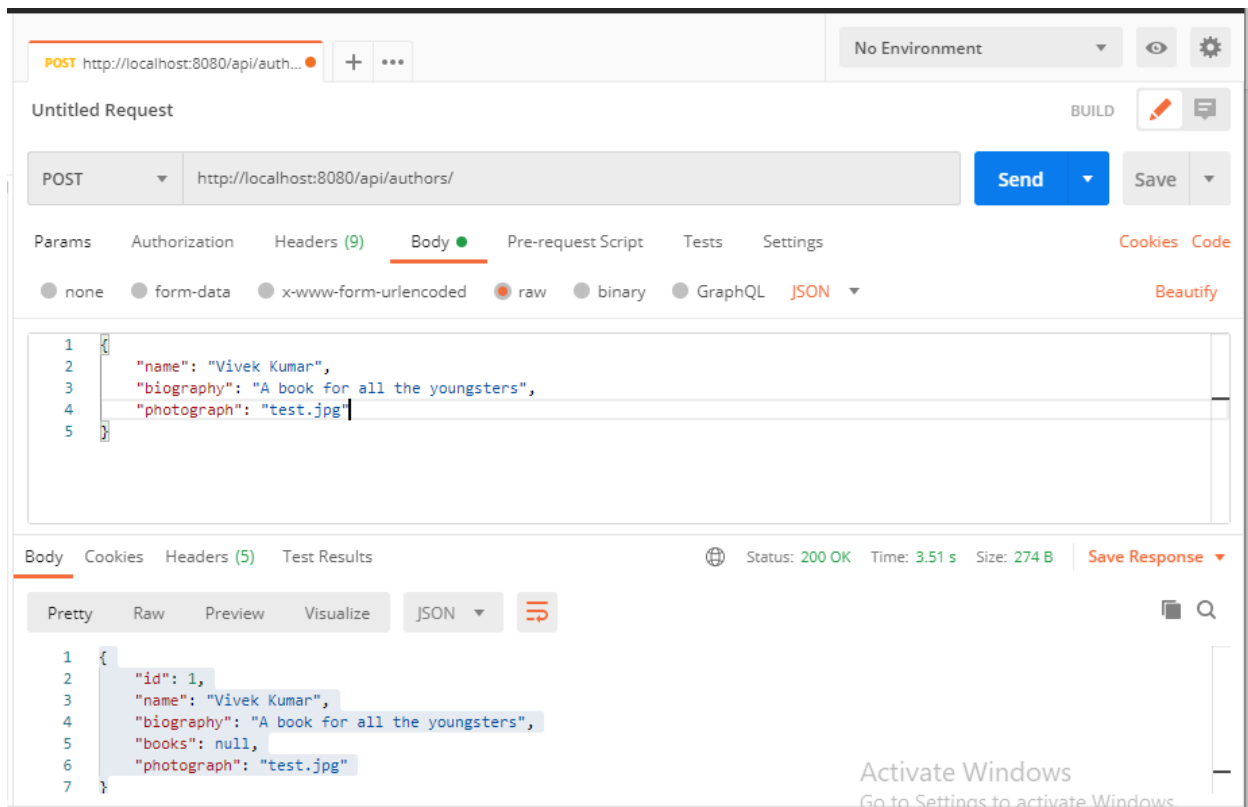
```
"name": "Vivek Kumar",
"biography": "A book for all the youngsters",
"photograph": "test.jpg"
```

}

Response: {

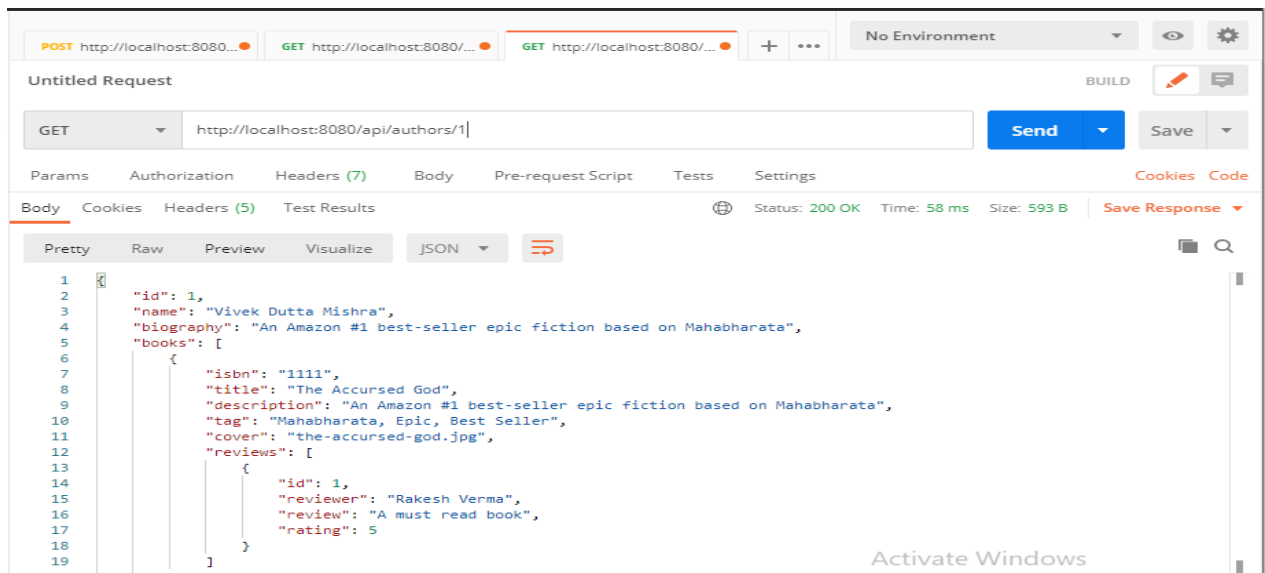
```
"id": 1,
"name": "Vivek Kumar",
"biography": "A book for all the youngsters",
"books": null,
"photograph": "test.jpg"
```

}



iii. Author by id(<http://localhost:8080/api/authors/1>)

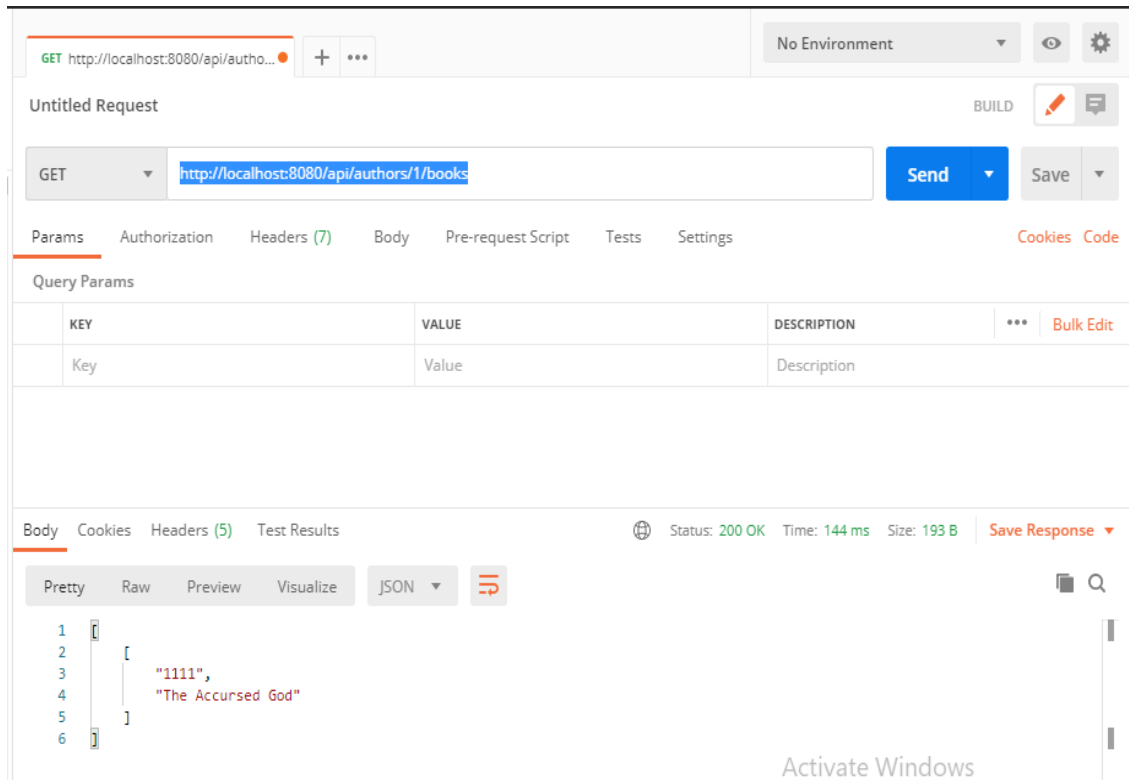
```
Response: {
  "id": 1,
  "name": "Vivek Dutta Mishra",
  "biography": "An Amazon #1 best-seller epic fiction based on Mahabharata",
  "books": [
    {
      "isbn": "1111",
      "title": "The Accursed God",
      "description": "An Amazon #1 best-
seller epic fiction based on Mahabharata",
      "tag": "Mahabharata, Epic, Best Seller",
      "cover": "the-accursed-god.jpg",
      "price": 250.0,
      "reviews": [
        {
          "id": 1,
          "reviewer": "Rakesh Verma",
          "review": "A must read book",
          "rating": 5
        }
      ]
    }
  ],
  "photograph": "the-accursed-god.jpg"
}
```



iv. All book by current Author(<http://localhost:8080/api/authors/1/books>)

Response: [

```
[
  "1111",
  "The Accursed God"
]
```



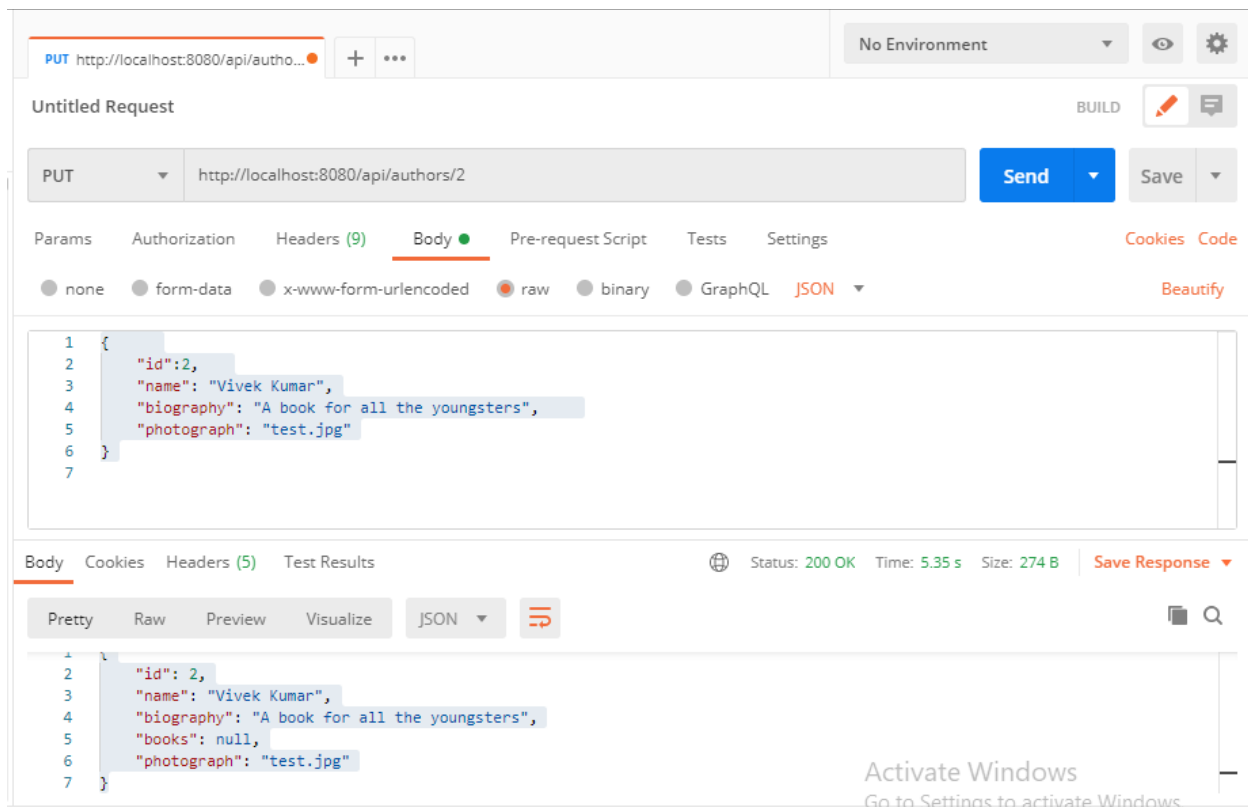
v. Update Author(<http://localhost:8080/api/authors/2>)

Request Body: {

```
"id": 2,  
"name": "Vivek Kumar",  
"biography": "A book for all the youngsters",  
"photograph": "test.jpg"  
}
```

Response: {

```
"id": 2,  
"name": "Vivek Kumar",  
"biography": "A book for all the youngsters",  
"books": null,  
"photograph": "test.jpg"  
}
```



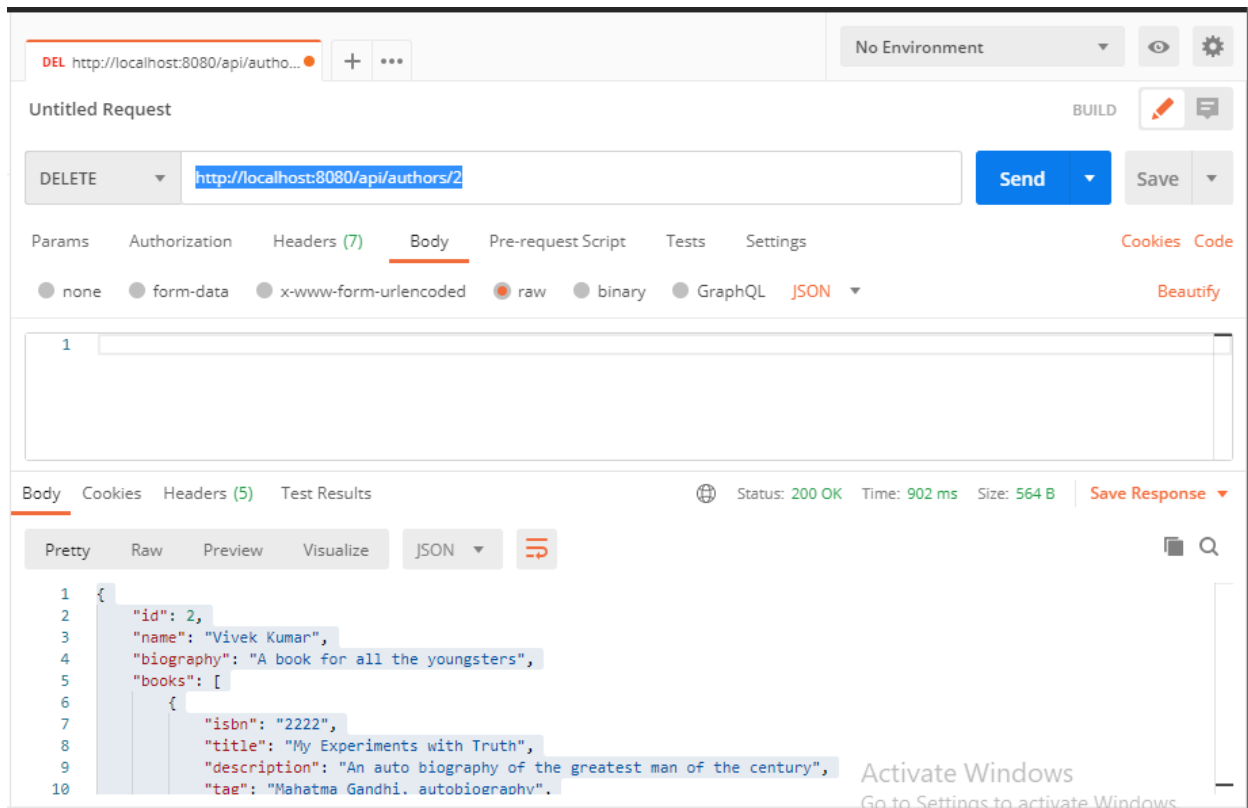
vi. Delete Author(<http://localhost:8080/api/authors/2>)

Response: {

```

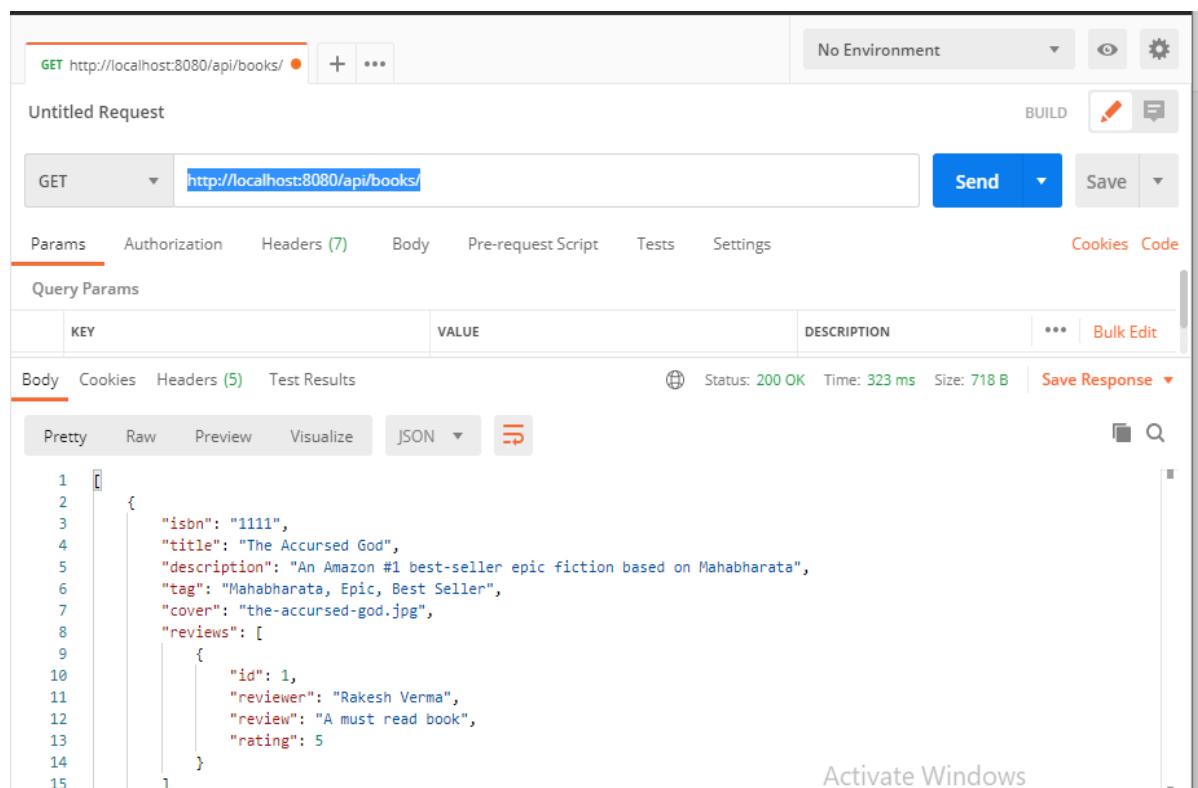
  "id": 2,
  "name": "Vivek Kumar",
  "biography": "A book for all the youngsters",
  "books": [
    {
      "isbn": "2222",
      "title": "My Experiments with Truth",
      "description": "An auto biography of the greatest man of the century",
      "tag": "Mahatma Gandhi, autobiography",
      "cover": "my-experiments-with-truth.jpg",
      "price": 350.0,
      "reviews": [
        {
          "id": 2,
          "reviewer": "Manish Pal",
          "review": "Amazing book",
          "rating": 4
        }
      ]
    }
  ],
  "photograph": "test.jpg"
}

```



2. Below are the APIs to manage Books

i. List all books(<http://localhost:8080/api/books/>)



ii. Create new book(<http://localhost:8080/api/books/>)

Request body: {

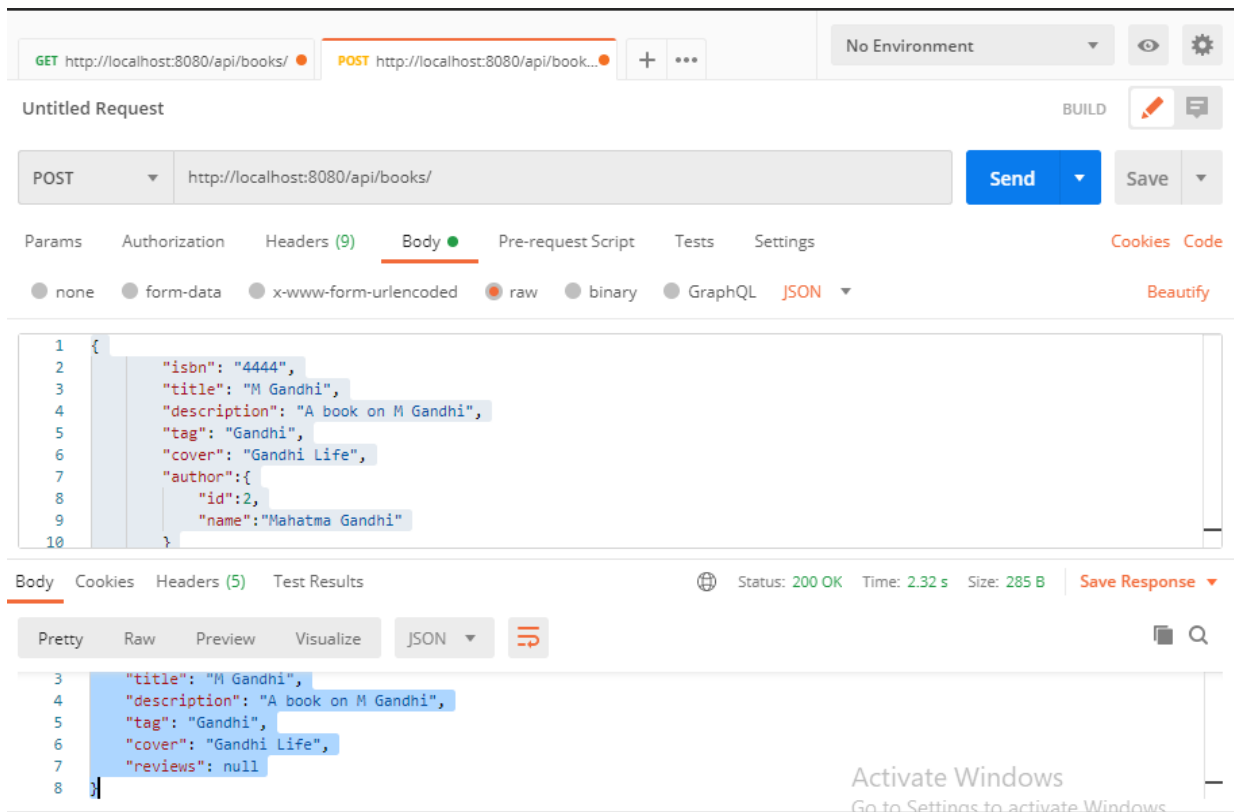
```
"isbn": "4444",
"title": "M Gandhi",
"description": "A book on M Gandhi",
"tag": "Gandhi",
"cover": "Gandhi Life",
"author": {
  "id": 2,
  "name": "Mahatma Gandhi"
}
```

}

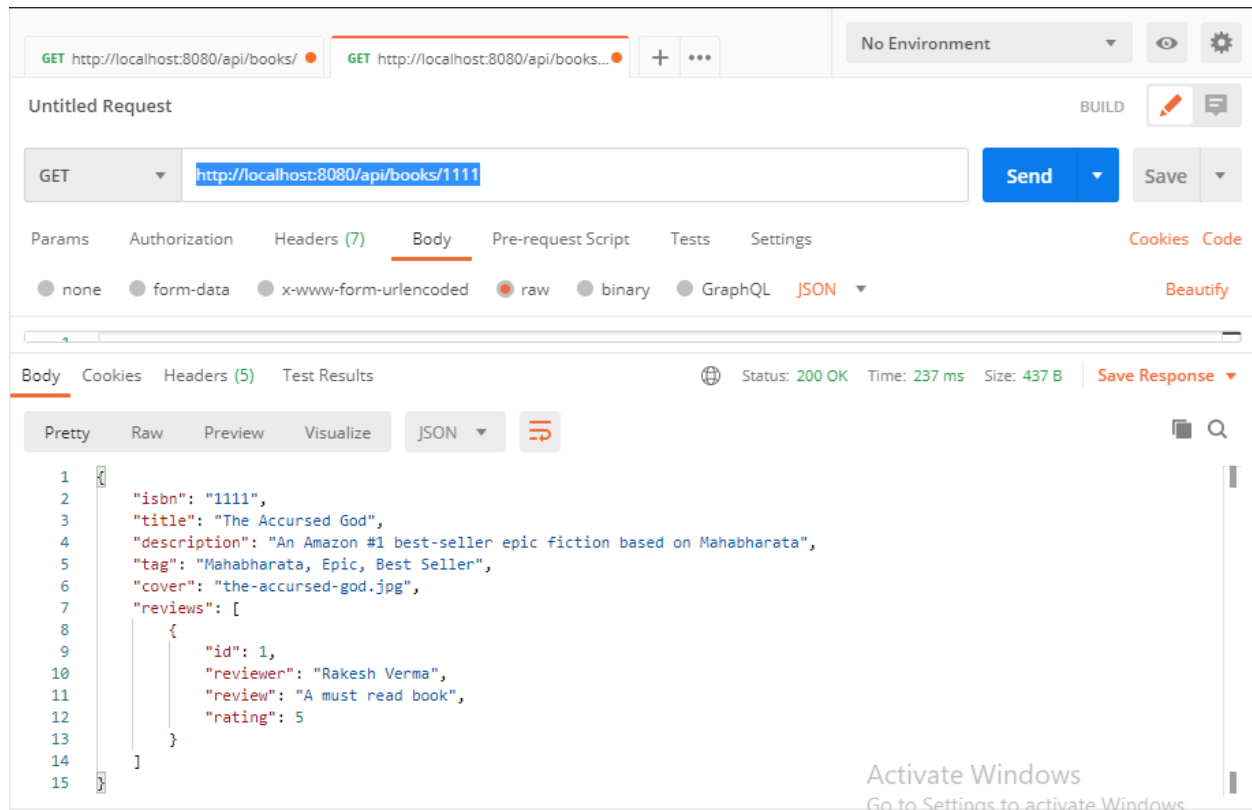
Response: {

```
"isbn": "4444",
"title": "M Gandhi",
"description": "A book on M Gandhi",
"tag": "Gandhi",
"cover": "Gandhi Life",
"reviews": null
```

}



iii. Book by isbn(<http://localhost:8080/api/books/1111>)



iv. Update book(`http://localhost:8080/api/books/1111`)

Request body: {

```

  "isbn": "1111",
  "title": "The Accursed God-2",
  "description": "An Amazon #1 best-seller epic fiction based on Mahabharata",
  "tag": "Mahabharata, Epic, Best Seller",
  "cover": "the-accursed-god.jpg",
  "author": {
    "id": 2,
    "name": "Mahatma Gandhi"
  }
}

```

Response: {

```

  "isbn": "1111",
  "title": "The Accursed God-2",
  "description": "An Amazon #1 best-seller epic fiction based on Mahabharata",
  "tag": "Mahabharata, Epic, Best Seller",
  "cover": "the-accursed-god.jpg",
  "reviews": null
}

```

GET http://localhost:8080/api/books/ PUT http://localhost:8080/api/books/ No Environment

Untitled Request BUILD

PUT http://localhost:8080/api/books/1111 Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "isbn": "1111",
3   "title": "The Accursed God-2",
4   "description": "An Amazon #1 best-seller epic fiction based on Mahabharata",
5   "tag": "Mahabharata, Epic, Best Seller",
6   "cover": "the-accursed-god.jpg",
7   "author": "Mahatma Gandhi",
8   "id": 2,
9   "name": "Mahatma Gandhi"
10 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 3.88 s Size: 368 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "isbn": "1111",
3   "title": "The Accursed God-2",
4   "description": "An Amazon #1 best-seller epic fiction based on Mahabharata",
5   "tag": "Mahabharata, Epic, Best Seller",
6   "cover": "the-accursed-god.jpg",
```

v. Delete book(<http://localhost:8080/api/books/4444>)

DEL http://localhost:8080/api/books/ No Environment

Untitled Request BUILD

DELETE <http://localhost:8080/api/books/4444> Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 1323 ms Size: 283 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "isbn": "4444",
3   "title": "M Gandhi",
4   "description": "A book on M Gandhi",
5   "tag": "Gandhi",
6   "cover": "Gandhi Life",
```

vi. Get book by matching author name part (<http://localhost:8080/api/books/by/Mahatma Gandhi>)

Response: [

```
{
  "isbn": "2222",
  "title": "My Experiments with Truth",
  "description": "An auto biography of the greatest man of the century",
  "tag": "Mahatma Gandhi, autobiography",
  "cover": "my-experiments-with-truth.jpg",
  "price": 350.0,
  "reviews": [
    {
      "id": 2,
      "reviewer": "Manish Pal",
      "review": "Amazing book",
      "rating": 4
    }
  ]
}
```

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

- Request:** GET <http://localhost:8080/api/books/by/Mahatma Gandhi>
- Response:** Status: 200 OK, Time: 1163 ms, Size: 720 B
- Response Body (JSON):**

```
[
  {
    "isbn": "1111",
    "title": "The Accursed God-2",
    "description": "An Amazon #1 best-seller epic fiction based on Mahabharata",
    "tag": "Mahabharata, Epic, Best Seller",
    "cover": "the-accursed-god.jpg",
    "reviews": [
      {
        "id": 1,
        "reviewer": "Rakesh Verma",
        "review": "A must read book",
        "rating": 5
      }
    ]
  }
]
```

vii. Get book in price range(<http://localhost:8080/api/books/price/between/250/and/300>)

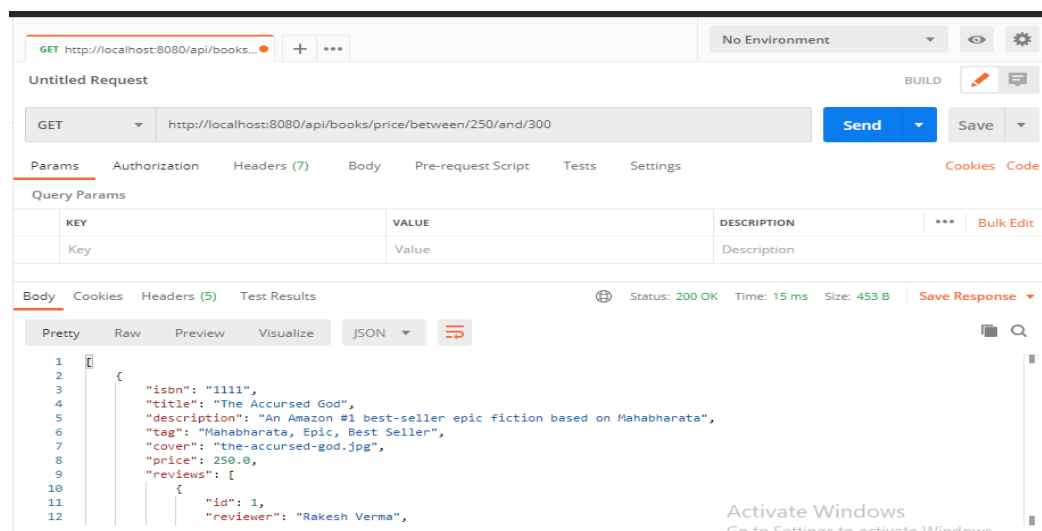
Response: [

```
{
  "isbn": "1111",
  "title": "The Accursed God",
  "description": "An Amazon #1 best-seller epic fiction based on Mahabharata",
  "tag": "Mahabharata, Epic, Best Seller",
  "cover": "the-accursed-god.jpg",
  "price": 250.0,
  "reviews": [
    {
      "id": 1,
      "reviewer": "Rakesh Verma",
      "review": "A must read book",
      "rating": 5
    }
  ]
}
```

```

    "tag": "Mahabharata, Epic, Best Seller",
    "cover": "the-accursed-god.jpg",
    "price": 250.0,
    "reviews": [
      {
        "id": 1,
        "reviewer": "Rakesh Verma",
        "review": "A must read book",
        "rating": 5
      }
    ]
  }
}
]

```



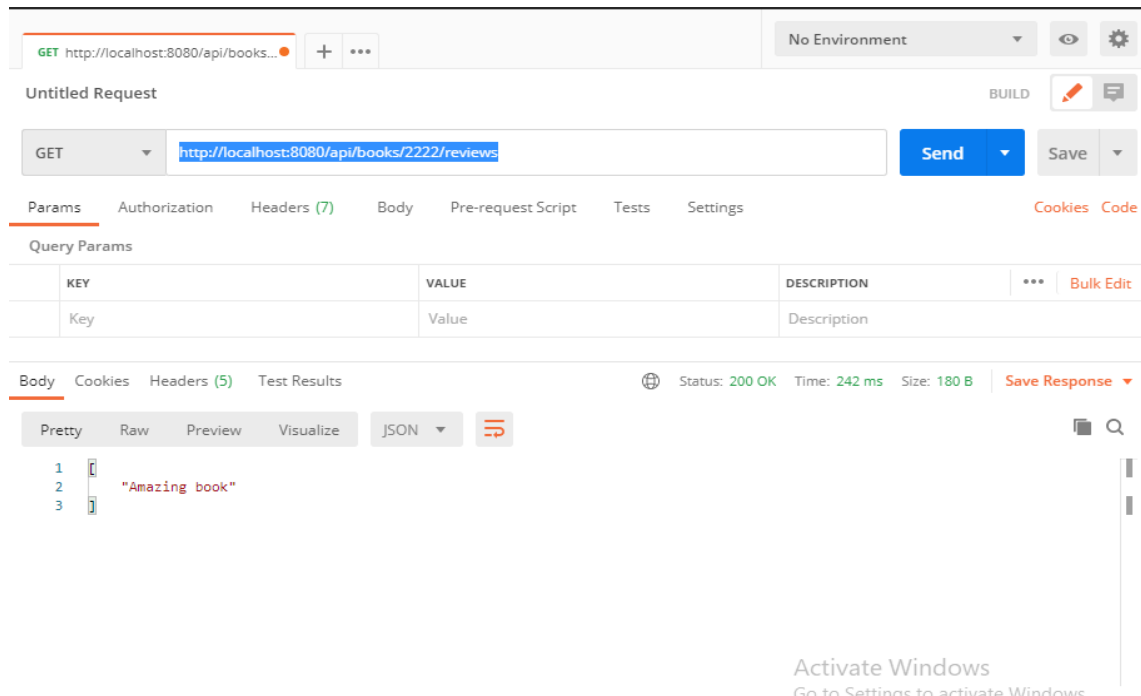
viii. Get all reviews for the given isbn(<http://localhost:8080/api/books/2222/reviews>)

Response: [

```

  "Amazing book"
]

```



3. Below are the APIs to manage Reviews

- i. Get all reviews(<http://localhost:8080/api/reviews/>)

Response: [

```
{
  "id": 1,
  "reviewer": "Rakesh Verma",
  "review": "A must read book",
  "rating": 5
},
{
  "id": 2,
  "reviewer": "Manish Pal",
  "review": "Amazing book",
  "rating": 4
}
]
```

GET http://localhost:8080/api/reviews/ No Environment

Untitled Request BUILD

GET http://localhost:8080/api/reviews/ Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 483 ms Size: 396 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "reviewer": "Rakesh Verma",
5     "review": "A must read book",
6     "rating": 5
7   },
8   {
9     "id": 2,
10    "reviewer": "Manish Pal",
11    "review": "Amazing book",
12    "rating": 4
13  }
14 ]
```

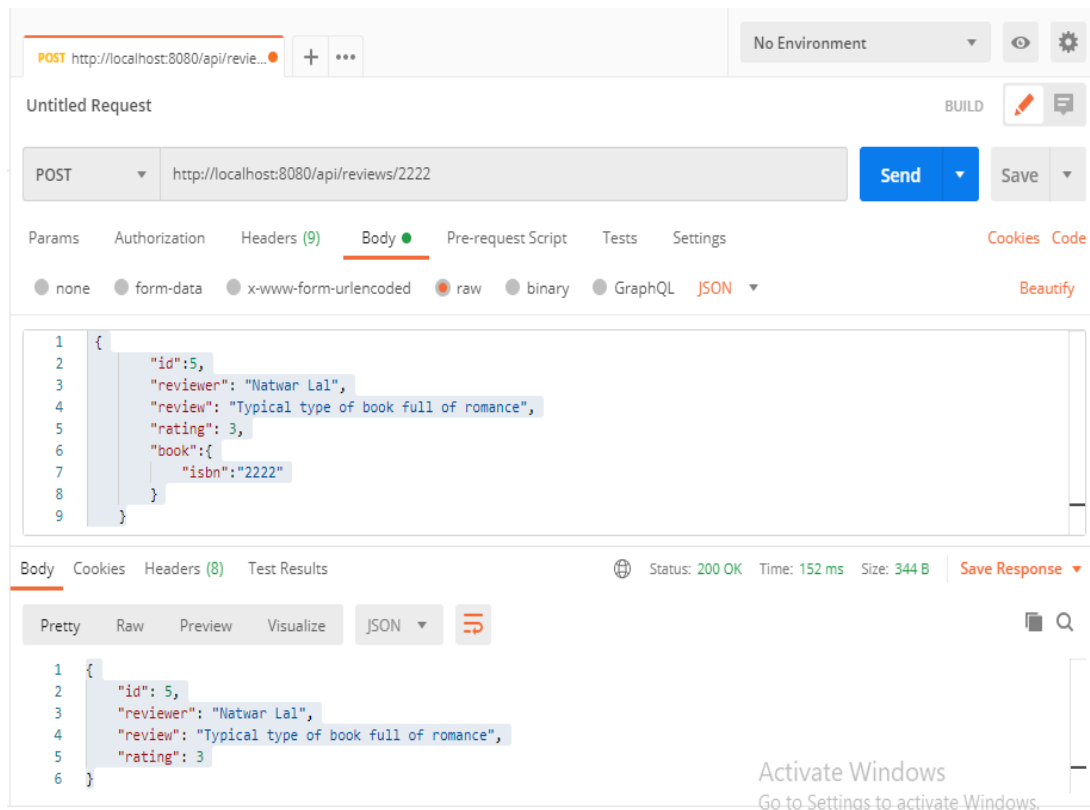
ii. ADD NEW REVIEW FOR GIVEN BOOK(<http://localhost:8080/api/reviews/2222>)

Request body:

```
{
  "id": 5,
  "reviewer": "Natwar Lal",
  "review": "Typical type of book full of romance",
  "rating": 3,
  "book": {
    "isbn": "2222"
  }
}
```

Response:

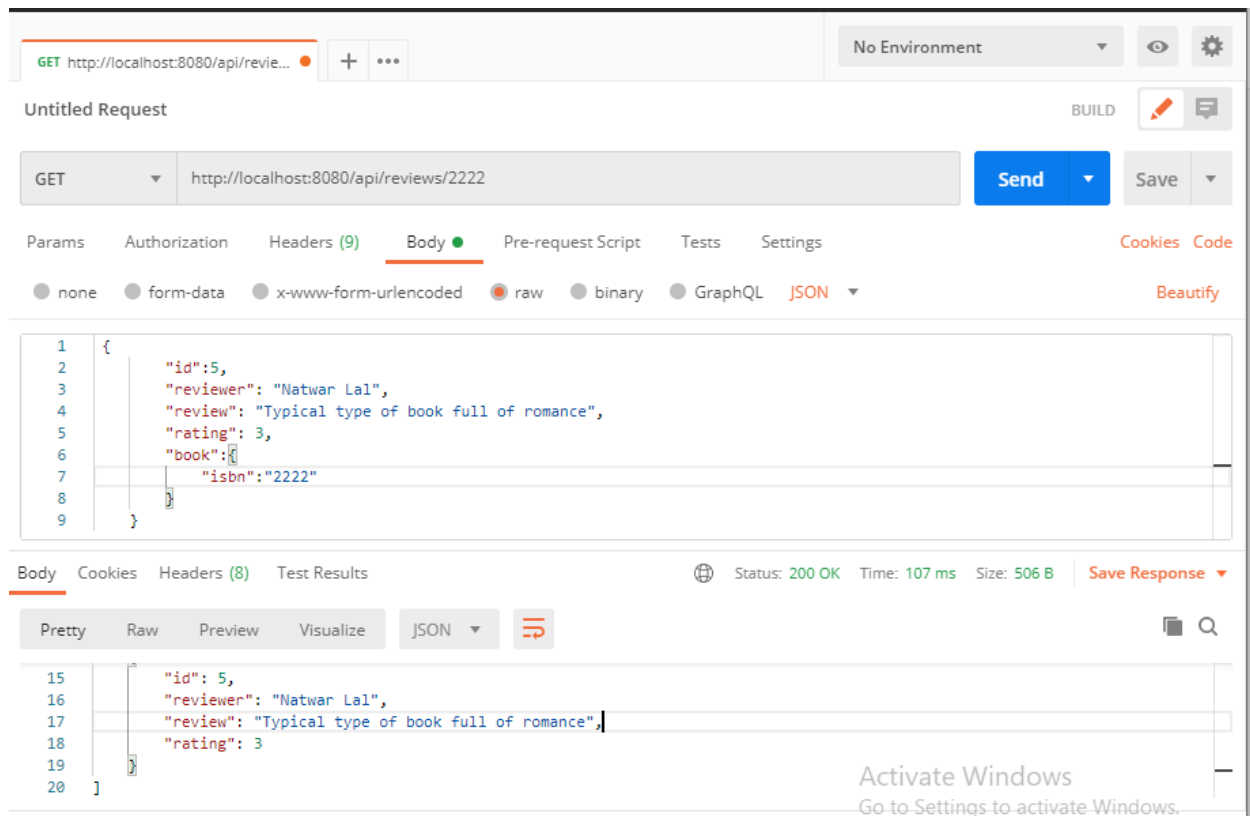
```
{
  "id": 5,
  "reviewer": "Natwar Lal",
  "review": "Typical type of book full of romance",
  "rating": 3
}
```



iii. Get all reviews for given isbn(<http://localhost:8080/api/reviews/2222>)

Response: [

```
{
  "id": 2,
  "reviewer": "Manish Pal",
  "review": "Amazing book",
  "rating": 4
},
{
  "id": 5,
  "reviewer": "Natwar Lal",
  "review": "Typical type of book full of romance",
  "rating": 3
}
]
```



iv. Get All reviews in min-max range(<http://localhost:8080/api/reviews/rating/between/2/and/4>)

Response: [

```
{
  "id": 2,
  "reviewer": "Manish Pal",
  "review": "Amazing book",
  "rating": 4
}
```

]

GET http://localhost:8080/api/reviews/rating/between/2/and/4

Send Save

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

1

Body Cookies Headers (8) Test Results Status: 200 OK Time: 18 ms Size: 322 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 2,
4     "reviewer": "Manish Pal",
5     "review": "Amazing book",
6     "rating": 4
7   }
8 ]
```

v. Return the average rating(<http://localhost:8080/api/reviews/2222/average/rating>)

GET http://localhost:8080/api/reviews/2222/average/rating

Send Save

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings Cookies Coc

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

1

Body Cookies Headers (8) Test Results Status: 200 OK Time: 8 ms Size: 254 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "average": 4
3 }
```

vi. Get all reviews containing text(<http://localhost:8080/api/reviews/containing/amazing>)
Response:

```
[
  {
    "id": 2,
    "reviewer": "Manish Pal",
    "review": "Amazing book",
    "rating": 4
  }
]
```

The screenshot displays a REST client interface. At the top, a GET request is configured for the URL `http://localhost:8080/api/reviews/containing/amazing`. The request body is set to raw JSON, containing an array with one object: `{ "id": 2, "reviewer": "Manish Pal", "review": "Amazing book", "rating": 4 }`. The response status is `200 OK` with a time of `18 ms` and a size of `322 B`. The response body is shown in a pretty-printed JSON format, matching the request body. A watermark for 'Activate Windows' is visible in the bottom right corner.

GET `http://localhost:8080/api/reviews/containing/amazing` Send Save

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings Cookies Code

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON Beautify

1

Body Cookies Headers (8) Test Results Status: 200 OK Time: 18 ms Size: 322 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 2,
4     "reviewer": "Manish Pal",
5     "review": "Amazing book",
6     "rating": 4
7   }
8 ]
```

Activate Windows
Go to Settings to activate Windows

-----End-----