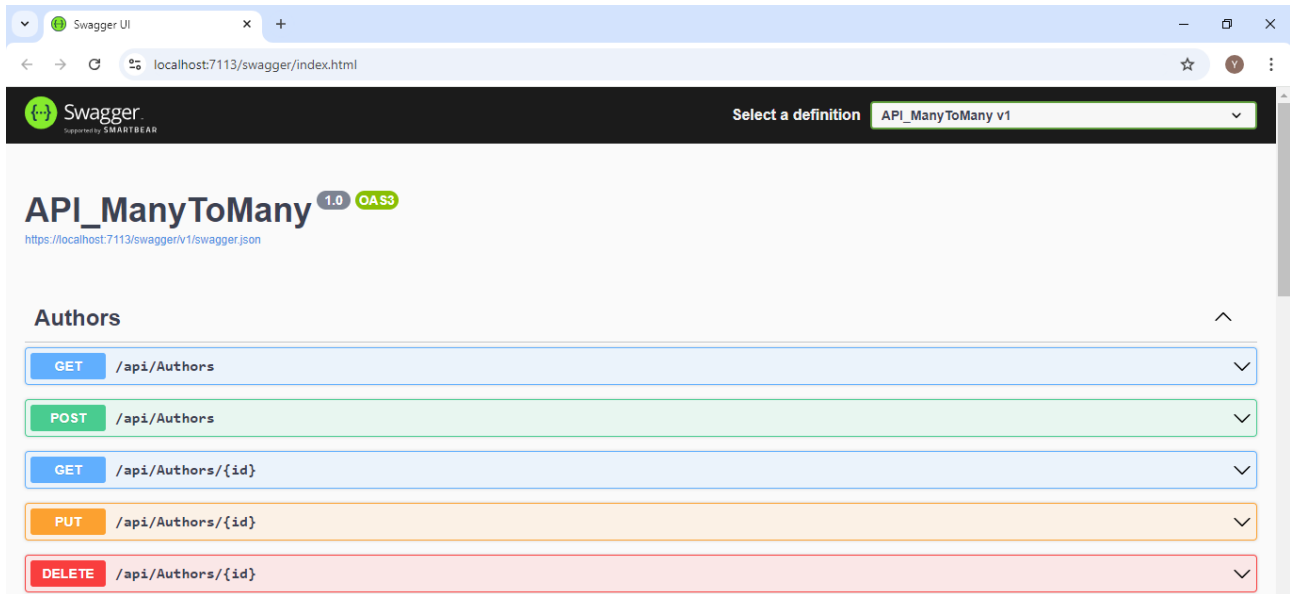


DATE : 04/09/2024

## API – Many to Many

OUTPUT:



The screenshot shows the Swagger UI for the API\_ManyToMany v1. The browser address bar indicates the URL is localhost:7113/swagger/index.html. The Swagger logo and version 1.0 OAS3 are visible. The API title is API\_ManyToMany. The Authors section is expanded, showing five endpoints:

- GET /api/Authors
- POST /api/Authors
- GET /api/Authors/{id}
- PUT /api/Authors/{id}
- DELETE /api/Authors/{id}



The screenshot shows the Swagger UI for the API\_ManyToMany v1, displaying the BookAuthors and Books sections. The browser address bar indicates the URL is localhost:7113/swagger/index.html. The Swagger logo and version 1.0 OAS3 are visible. The API title is API\_ManyToMany. The BookAuthors section is expanded, showing five endpoints:

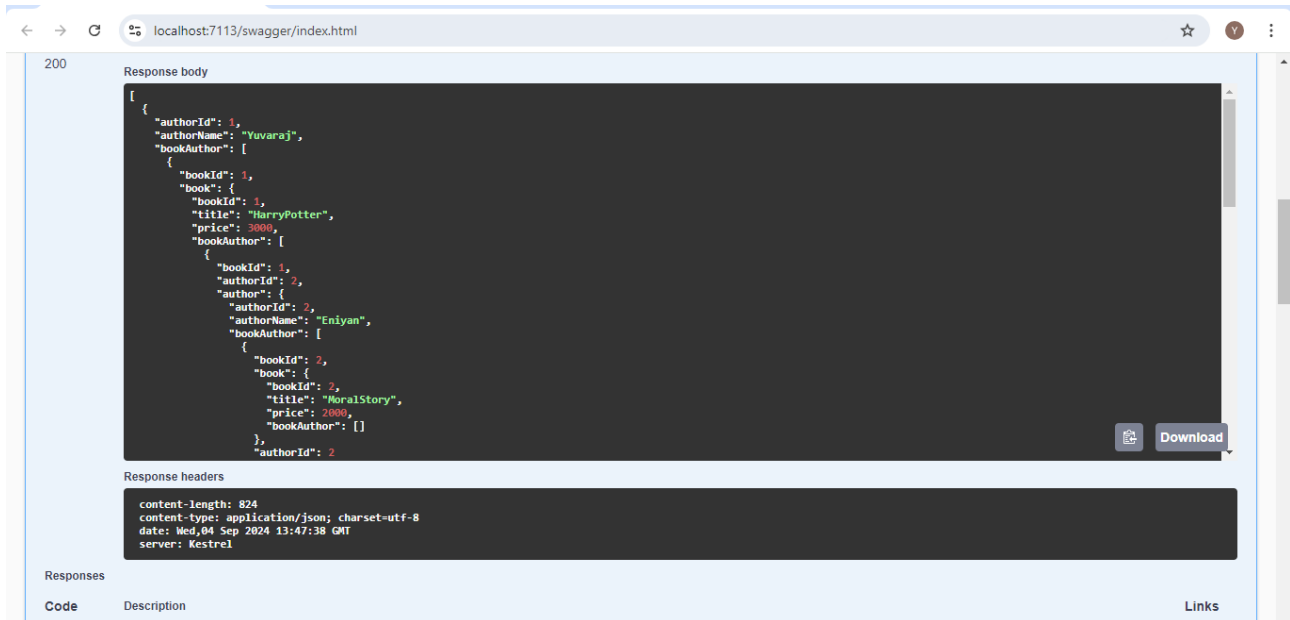
- GET /api/BookAuthors
- POST /api/BookAuthors
- GET /api/BookAuthors/{id}
- PUT /api/BookAuthors/{id}
- DELETE /api/BookAuthors/{bookId}/{authorId}

The Books section is also expanded, showing five endpoints:

- GET /api/Books
- POST /api/Books
- GET /api/Books/{id}
- PUT /api/Books/{id}
- DELETE /api/Books/{id}

Author:

Get:



The image shows a Swagger UI response view for the GET /api/Authors endpoint. The status code is 200. The response body is a JSON array containing two author objects. The first author has an id of 1, name 'Yuvaraj', and a list of books including 'Harry Potter'. The second author has an id of 2, name 'Eniyan', and a list of books including 'Moral Story'. The response headers show content-length: 824, content-type: application/json; charset=utf-8, date: Wed, 04 Sep 2024 13:47:38 GMT, and server: Kestrel.

```
200
Response body
[
  {
    "authorId": 1,
    "authorName": "Yuvaraj",
    "bookAuthor": [
      {
        "bookId": 1,
        "book": {
          "bookId": 1,
          "title": "Harry Potter",
          "price": 3600,
          "bookAuthor": [
            {
              "bookId": 1,
              "authorId": 2,
              "author": {
                "authorId": 2,
                "authorName": "Eniyan",
                "bookAuthor": [
                  {
                    "bookId": 2,
                    "book": {
                      "bookId": 2,
                      "title": "Moral Story",
                      "price": 2000,
                      "bookAuthor": []
                    },
                    "authorId": 2
                  }
                ]
              }
            }
          ]
        }
      }
    ]
  }
]
```

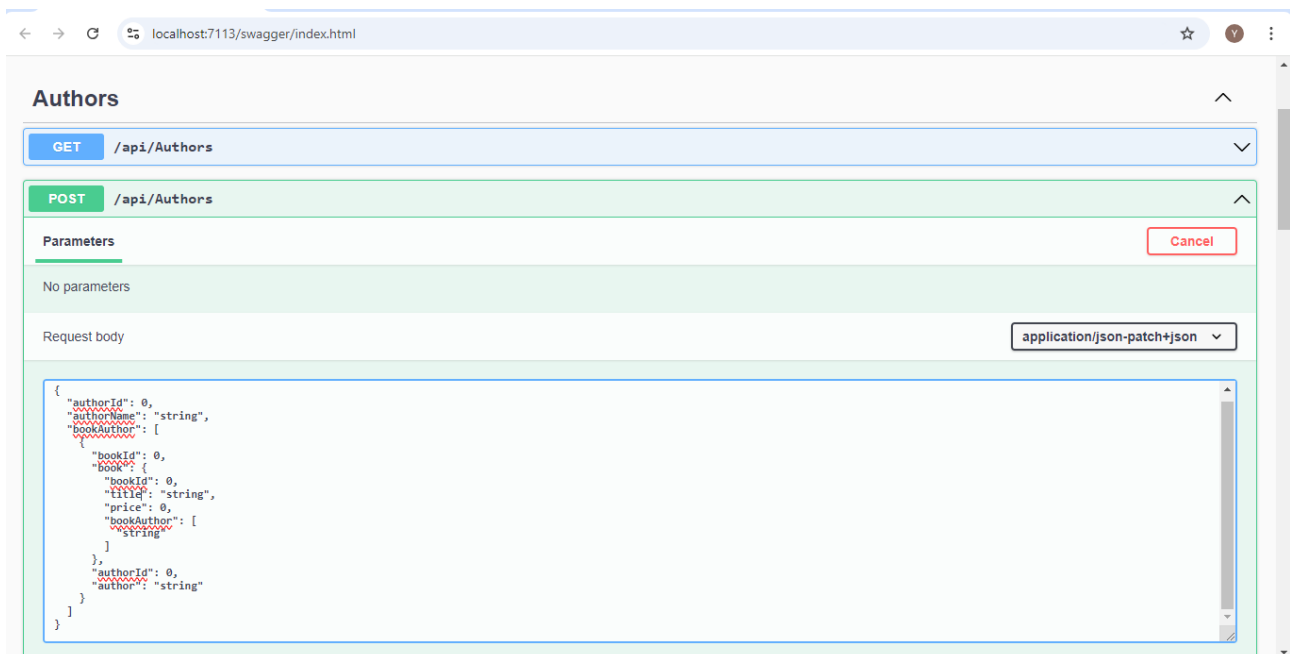
Response headers

```
content-length: 824
content-type: application/json; charset=utf-8
date: Wed, 04 Sep 2024 13:47:38 GMT
server: Kestrel
```

Responses

Code	Description	Links
------	-------------	-------

Post:



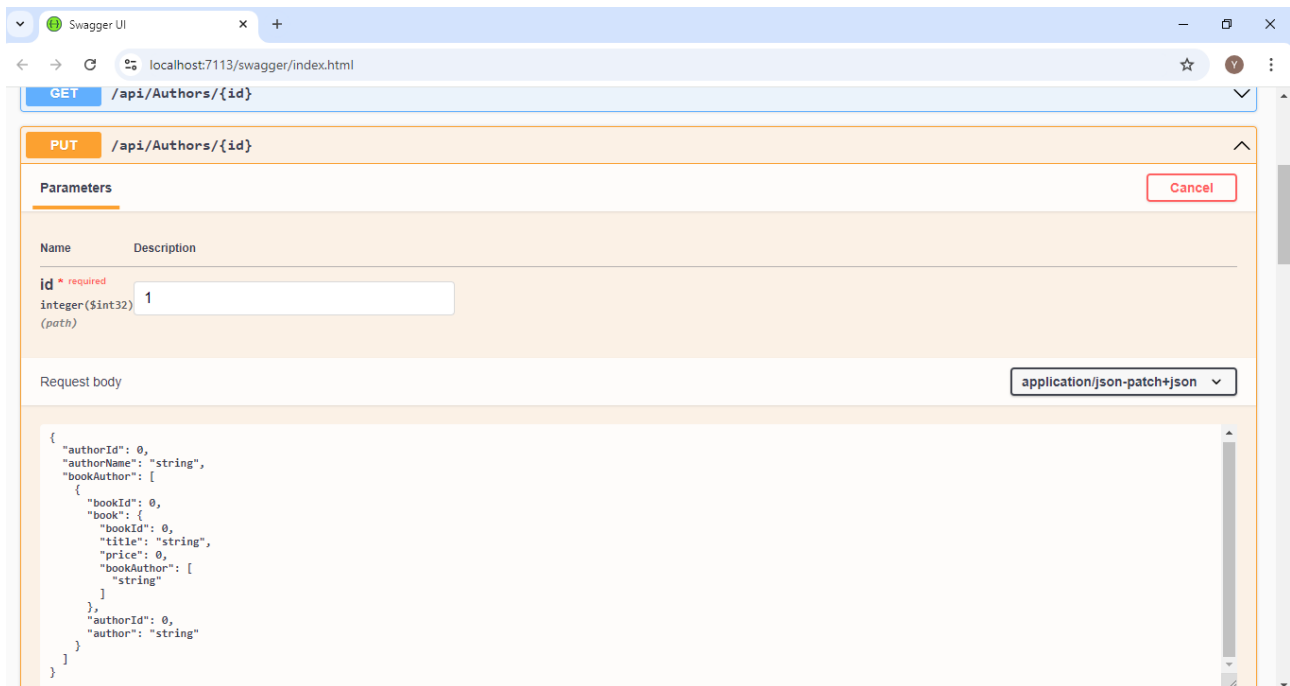
The image shows the Swagger UI POST view for the /api/Authors endpoint. The method is POST. There are no parameters. The request body is a JSON object with fields for authorId, authorName, and bookAuthor. The media type is set to application/json-patch+json.

```
POST /api/Authors

Parameters
No parameters

Request body
application/json-patch+json
{
  "authorId": 0,
  "authorName": "string",
  "bookAuthor": [
    {
      "bookId": 0,
      "book": {
        "bookId": 0,
        "title": "string",
        "price": 0,
        "bookAuthor": [
          "string"
        ]
      }
    },
    {
      "authorId": 0,
      "author": "string"
    }
  ]
}
```

Put:



The image shows the Swagger UI interface for a PUT endpoint at `/api/Authors/{id}`. The endpoint is highlighted in orange. The parameters section shows a required integer path parameter `id` with a value of `1`. The request body is set to `application/json-patch+json` and contains a JSON patch document. A red 'Cancel' button is visible in the top right of the parameters section.

PUT `/api/Authors/{id}`

Parameters

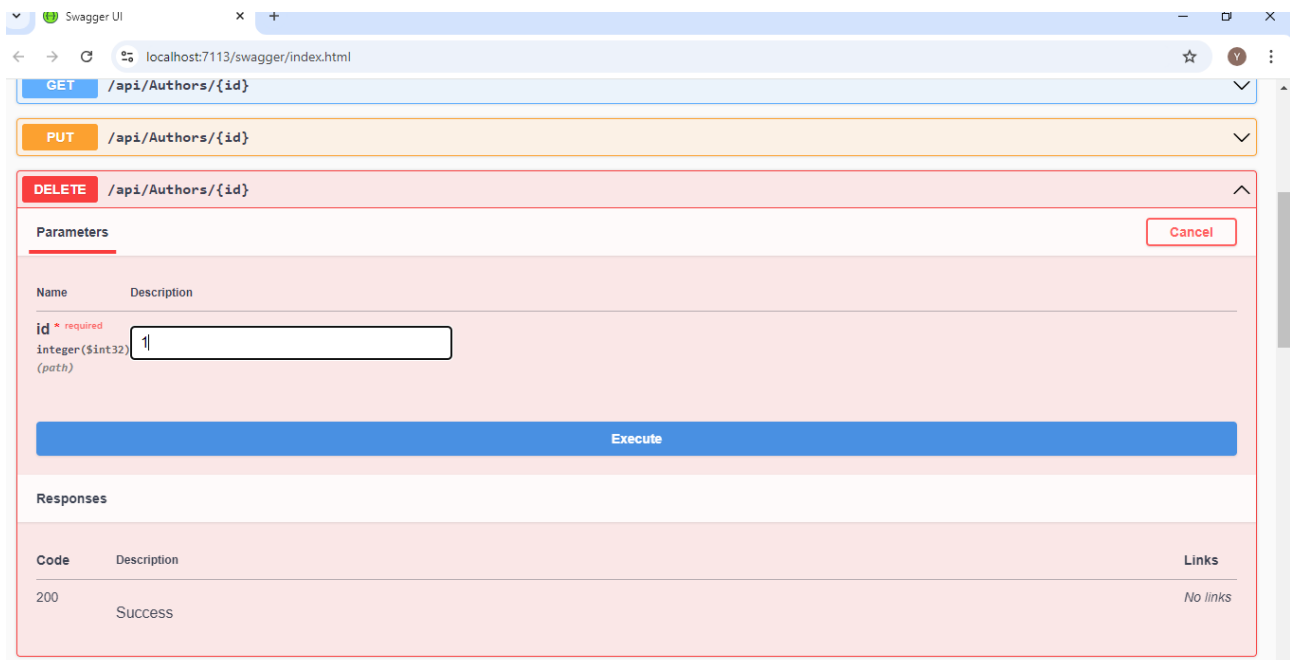
Name	Description
<code>id</code> * required <code>integer(\$int32)</code> (path)	1

Request body

application/json-patch+json

```
{
  "authorId": 0,
  "authorName": "string",
  "bookAuthor": [
    {
      "bookId": 0,
      "book": {
        "bookId": 0,
        "title": "string",
        "price": 0,
        "bookAuthor": [
          "string"
        ]
      },
      "authorId": 0,
      "author": "string"
    }
  ]
}
```

Delete:



The image shows the Swagger UI interface for a DELETE endpoint at `/api/Authors/{id}`. The endpoint is highlighted in red. The parameters section shows a required integer path parameter `id` with a value of `1`. A blue 'Execute' button is visible below the parameters section. The responses section shows a 200 status code with a 'Success' description and 'No links'.

DELETE `/api/Authors/{id}`

Parameters

Name	Description
<code>id</code> * required <code>integer(\$int32)</code> (path)	1

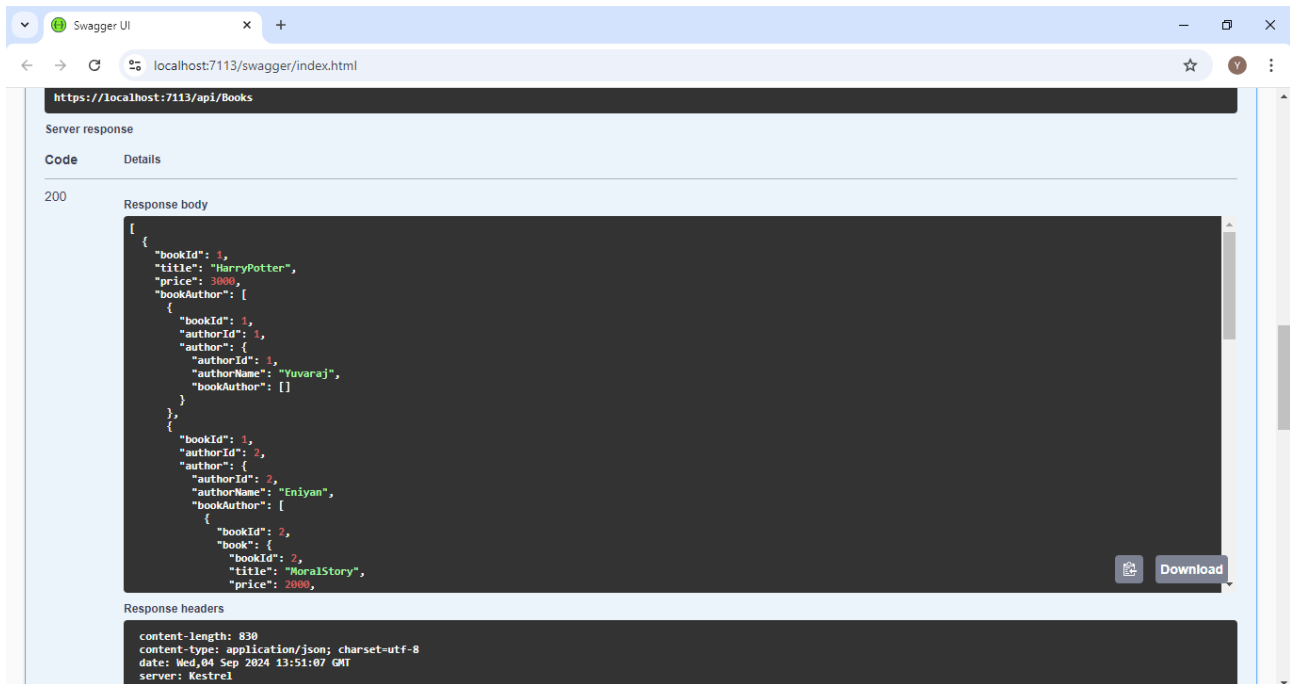
Execute

Responses

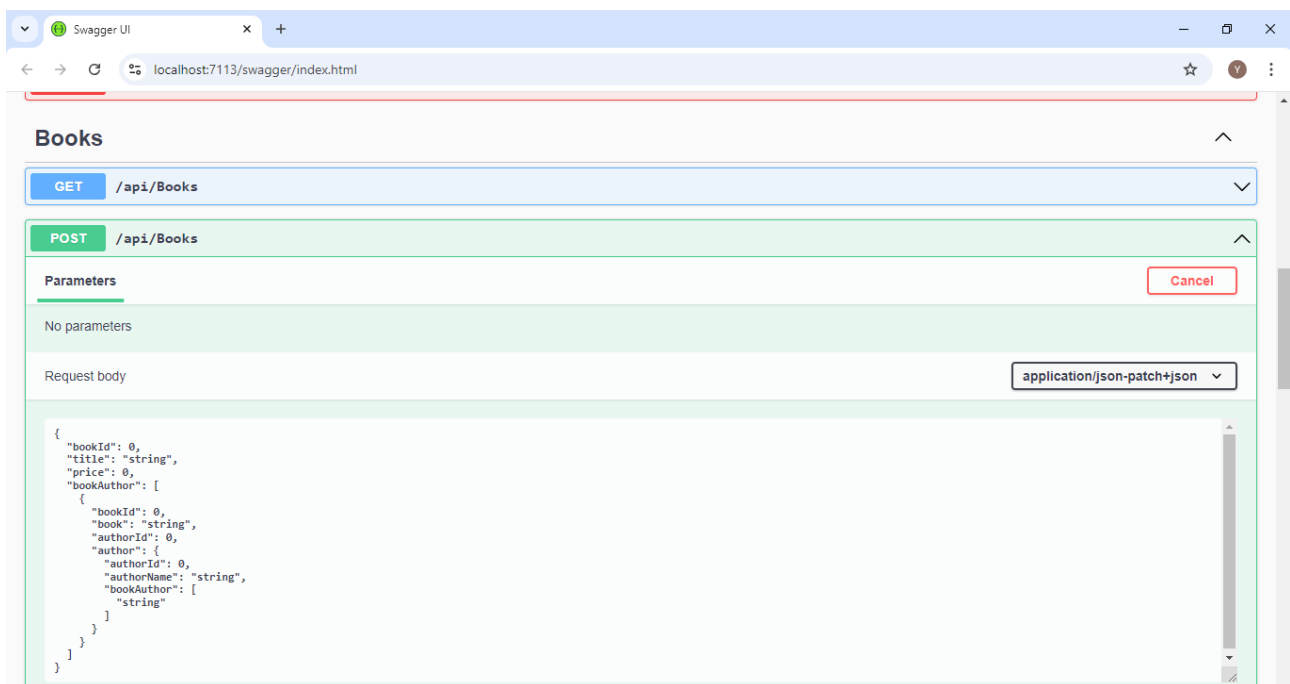
Code	Description	Links
200	Success	No links

Book:

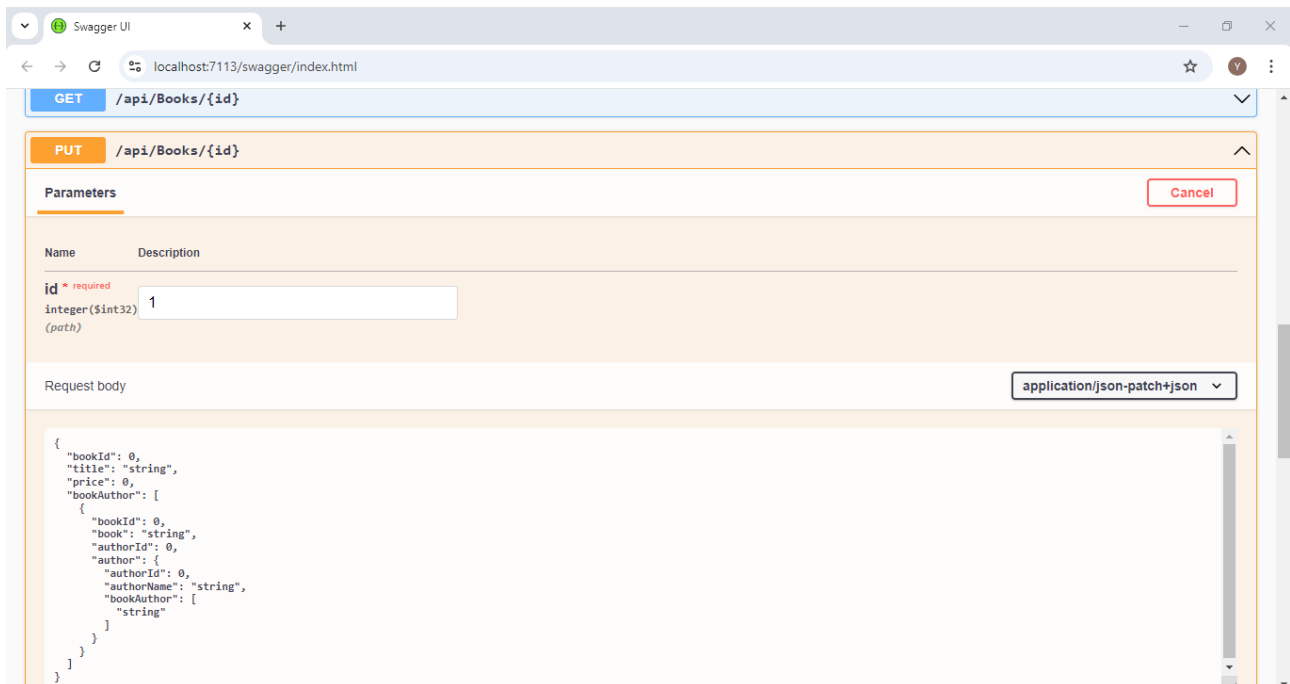
Get:



Post:



Put:



The image shows the Swagger UI interface for a PUT endpoint at `/api/Books/{id}`. The endpoint is highlighted in orange. The parameters section shows a required integer path parameter `id` with a value of `1`. The request body is set to `application/json-patch+json` and contains a JSON patch document. A red 'Cancel' button is visible in the top right of the parameters section.

Parameters

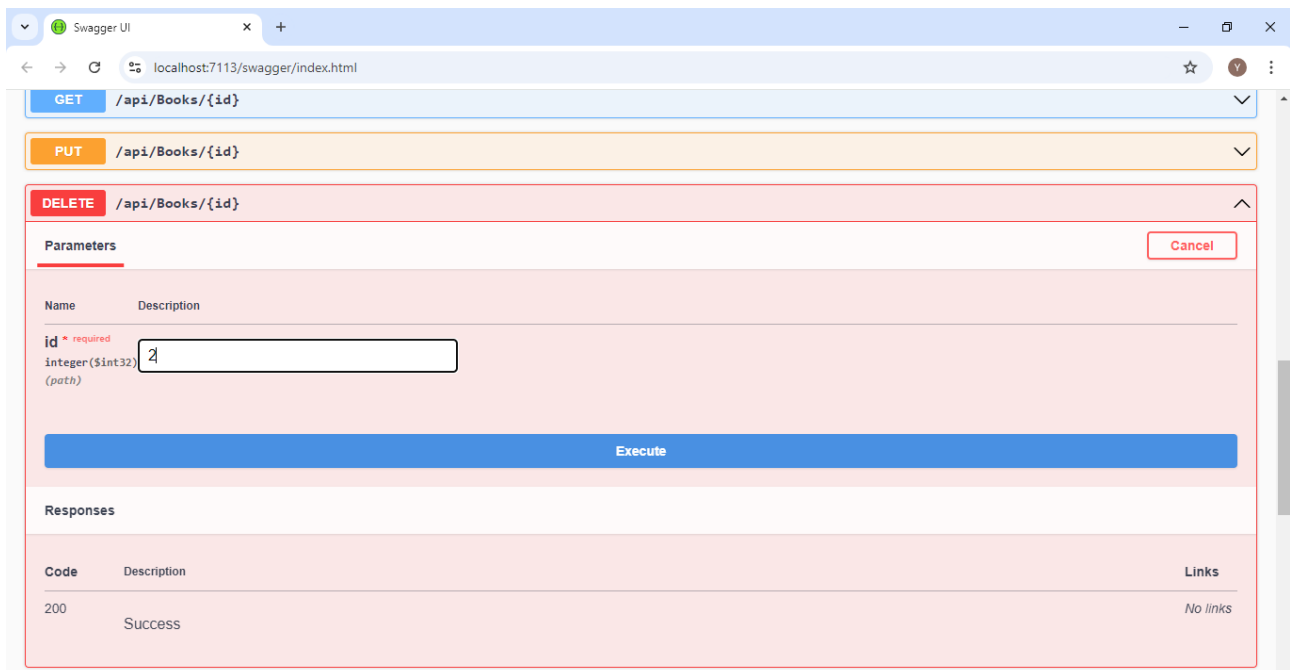
Name	Description
<code>id</code> * required	
integer(\$int32)	
(path)	

Request body

application/json-patch+json

```
{
  "bookId": 0,
  "title": "string",
  "price": 0,
  "bookAuthor": [
    {
      "bookId": 0,
      "book": "string",
      "authorId": 0,
      "author": {
        "authorId": 0,
        "authorName": "string",
        "bookAuthor": [
          "string"
        ]
      }
    }
  ]
}
```

Delete:



The image shows the Swagger UI interface for a DELETE endpoint at `/api/Books/{id}`. The endpoint is highlighted in red. The parameters section shows a required integer path parameter `id` with a value of `2`. A blue 'Execute' button is visible below the parameters section. The responses section shows a 200 status code with a 'Success' description and 'No links'.

Parameters

Name	Description
<code>id</code> * required	
integer(\$int32)	
(path)	

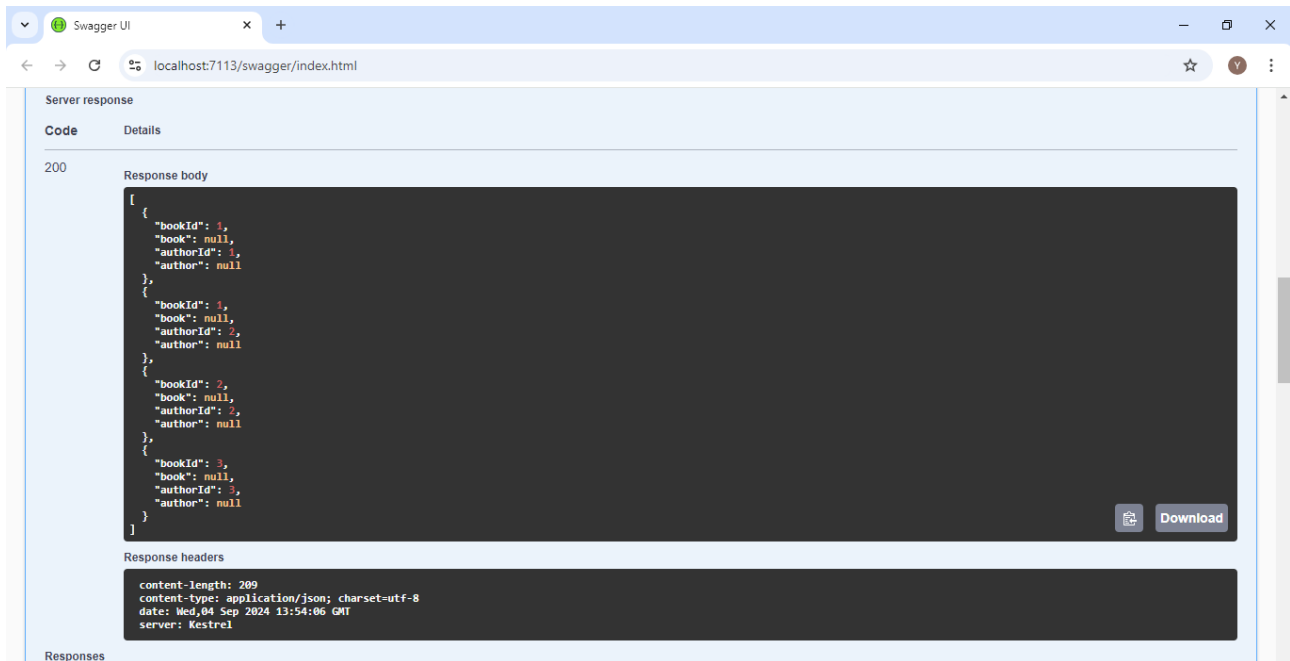
Execute

Responses

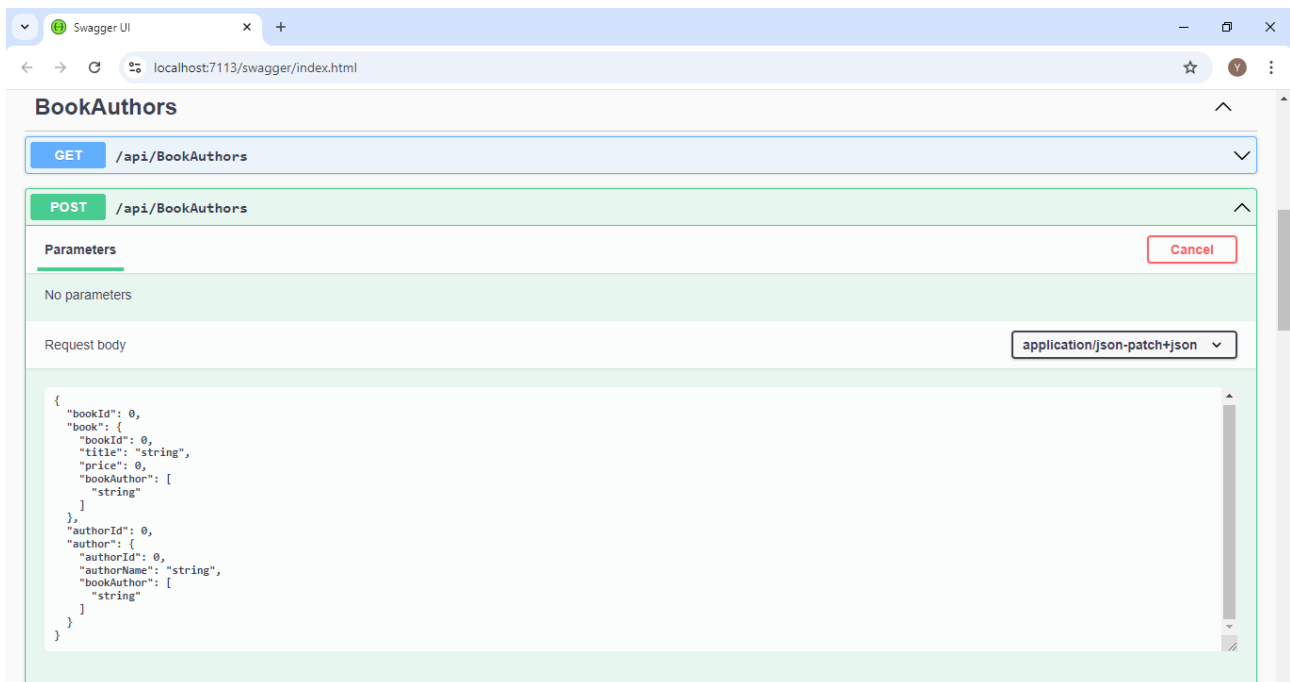
Code	Description	Links
200	Success	No links

BookAuthor:

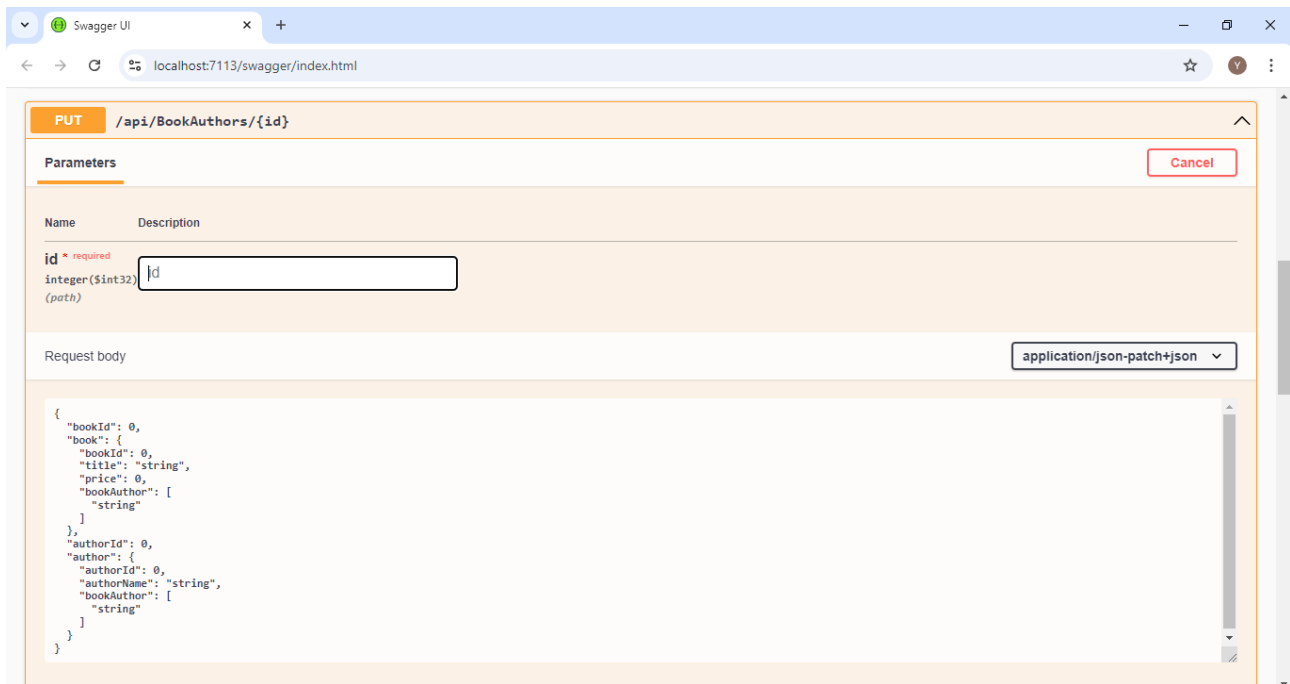
Get:



Post:



Put:



The image shows the Swagger UI interface for a PUT endpoint. The browser address bar shows 'localhost:7113/swagger/index.html'. The endpoint is '/api/BookAuthors/{id}' with a 'PUT' method. The 'Parameters' section has a table with 'Name' and 'Description' columns. A parameter 'id' is listed as 'required', 'integer(\$int32)', and '(path)'. A text input field next to it contains the value '1'. The 'Request body' section has a dropdown menu set to 'application/json-patch+json'. Below it, a JSON patch document is displayed in a text area.

**PUT** /api/BookAuthors/{id}

**Parameters**

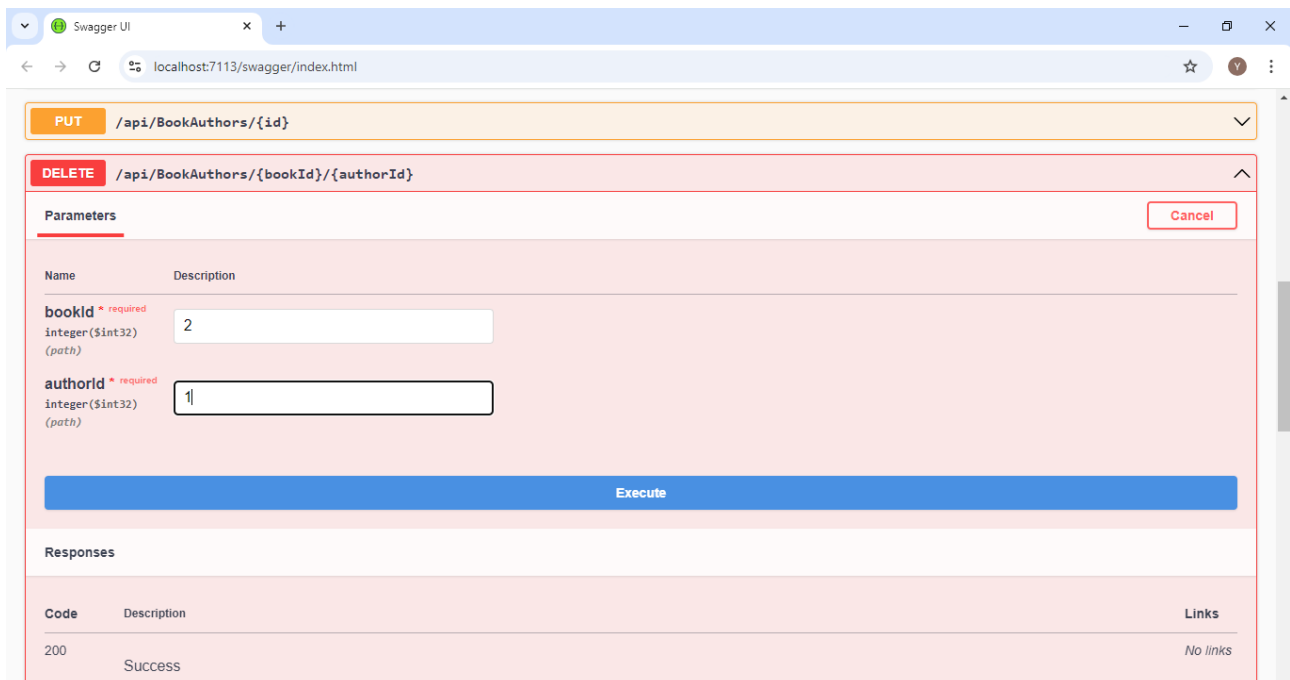
Name	Description
id * required integer(\$int32) (path)	<input type="text" value="1"/>

Request body

application/json-patch+json

```
{
  "bookId": 0,
  "book": {
    "bookId": 0,
    "title": "string",
    "price": 0,
    "bookAuthor": [
      "string"
    ]
  },
  "authorId": 0,
  "author": {
    "authorId": 0,
    "authorName": "string",
    "bookAuthor": [
      "string"
    ]
  }
}
```

Delete:



The image shows the Swagger UI interface for a DELETE endpoint. The browser address bar shows 'localhost:7113/swagger/index.html'. The endpoint is '/api/BookAuthors/{bookId}/{authorId}' with a 'DELETE' method. The 'Parameters' section has a table with 'Name' and 'Description' columns. Two parameters are listed: 'bookId' (required, integer(\$int32), path) with a value of '2', and 'authorId' (required, integer(\$int32), path) with a value of '1'. A blue 'Execute' button is visible. The 'Responses' section shows a table with 'Code', 'Description', and 'Links' columns. A response with code '200' and description 'Success' is listed, with 'No links' in the 'Links' column.

**DELETE** /api/BookAuthors/{bookId}/{authorId}

**Parameters**

Name	Description
bookId * required integer(\$int32) (path)	<input type="text" value="2"/>
authorId * required integer(\$int32) (path)	<input type="text" value="1"/>

**Execute**

**Responses**

Code	Description	Links
200	Success	No links