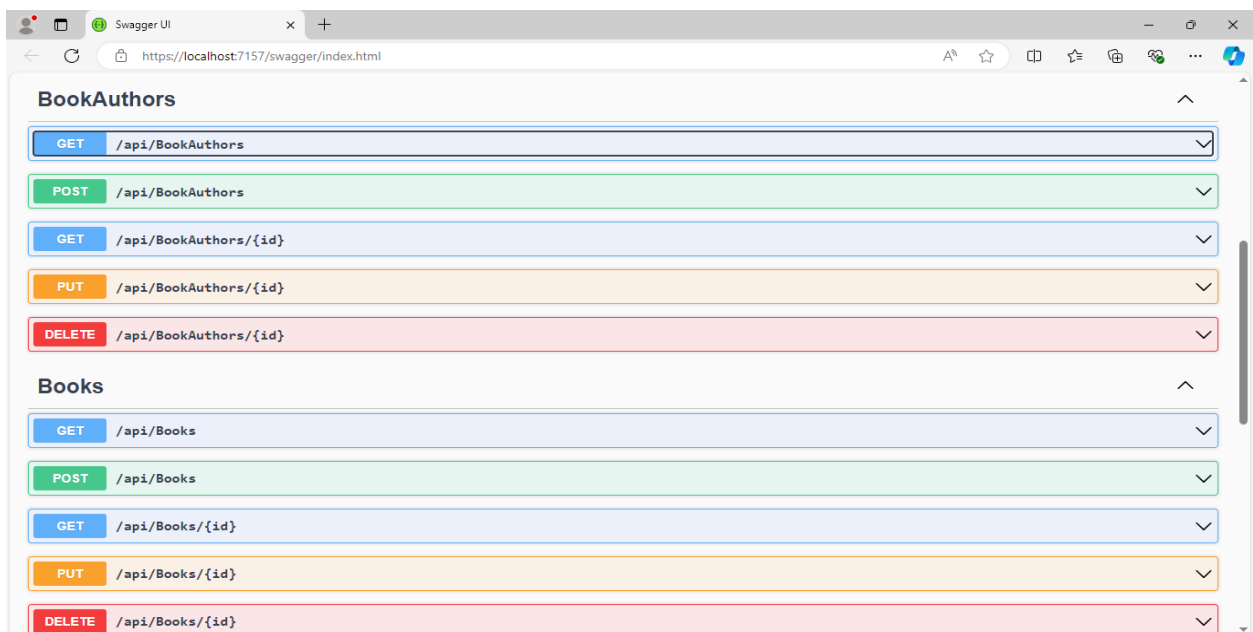
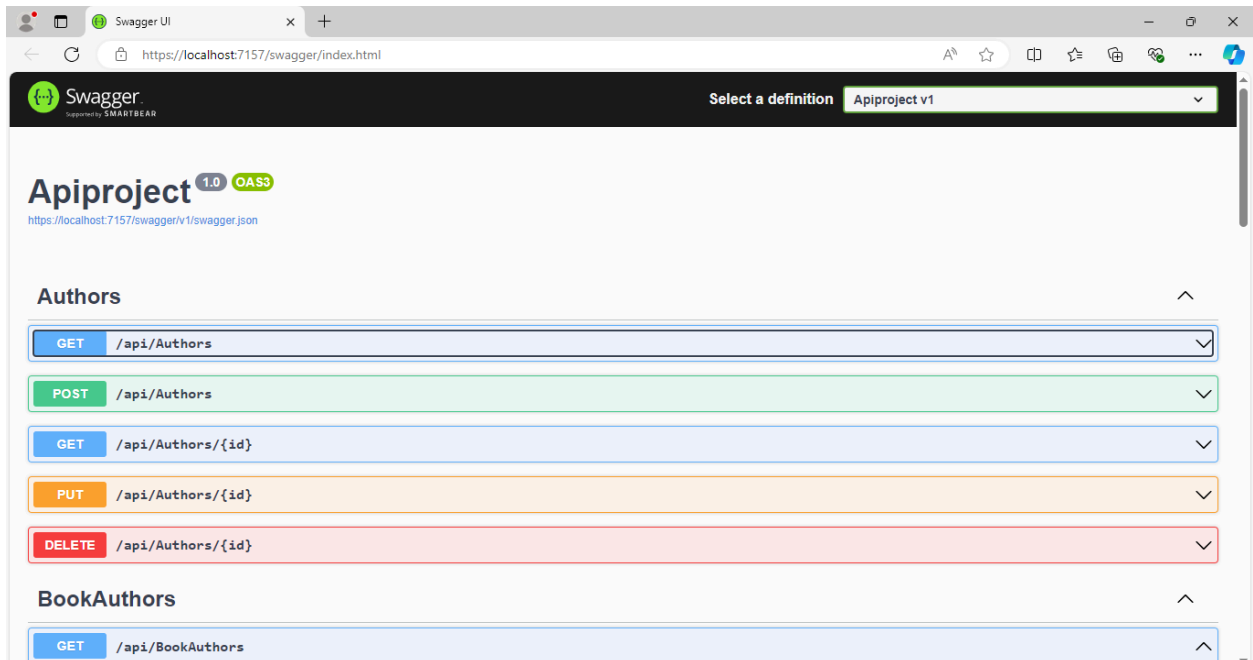


Date:04.09.20242

## API -Many to Many

### OUTPUT



# Author

## GET

The screenshot shows the Swagger UI interface for the endpoint `GET /api/Authors`. The request URL is `https://localhost:7157/api/Authors`. The server response is a 200 status code with a JSON body containing an array of three author objects. The response headers include `content-length: 160`, `content-type: application/json; charset=utf-8`, `date: Thu, 09 Sep 2024 04:01:51 GMT`, and `server: Kestrel`.

```
curl -H 'accept: text/plain' https://localhost:7157/api/Authors
```

Request URL

`https://localhost:7157/api/Authors`

Server response

Code	Details
200	<p>Response body</p> <pre>[{"authorId": 1, "authorName": "Sam", "bookAuthors": null}, {"authorId": 2, "authorName": "sam", "bookAuthors": null}, {"authorId": 3, "authorName": "Ram", "bookAuthors": null}]</pre> <p>Response headers</p> <pre>content-length: 160 content-type: application/json; charset=utf-8 date: Thu, 09 Sep 2024 04:01:51 GMT server: Kestrel</pre>

Responses

Code	Description	Links
200		No links

## Post

The screenshot shows the Swagger UI interface for the endpoint `POST /api/Authors`. The endpoint is currently selected, and the parameters section is empty. The request body is set to `application/json-patch+json` and contains a JSON object with `authorId` and `authorName` fields. The `Execute` button is visible at the bottom.

GET /api/Authors

POST /api/Authors

Parameters

No parameters

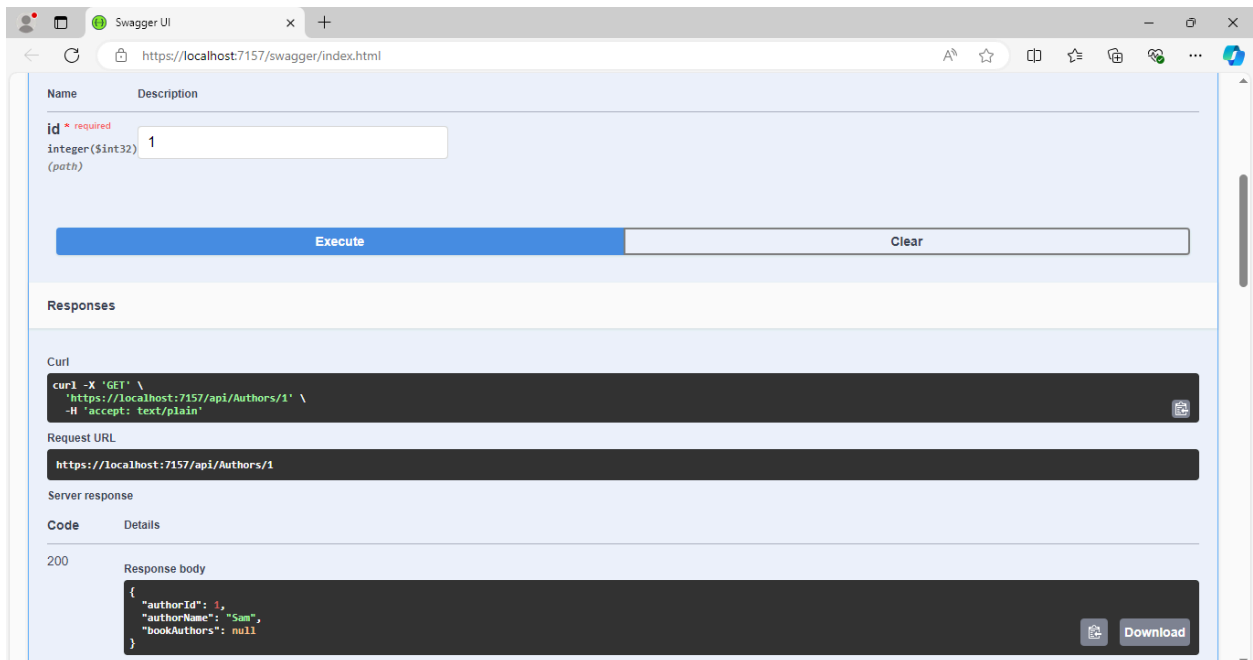
Request body

`application/json-patch+json`

```
{  "authorId": 1,  "authorName": "Sam",}
```

Execute

## GET



The image shows the Swagger UI interface for a GET endpoint. The endpoint is named `/api/Authors/{id}` and is located at `https://localhost:7157/swagger/index.html`. The parameter `id` is required and is of type `integer($int32)`. The user has entered the value `1` for the `id` parameter. The `Execute` button is highlighted in blue. Below the `Execute` button, the `Responses` section is visible. The `200` response is selected, showing the `Response body` as a JSON object: `{ "authorId": 1, "authorName": "Sam", "bookAuthors": null }`. The `Download` button is visible next to the response body.

Name Description

**id** \* required  
integer(\$int32)  
(path)

1

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7157/api/Authors/1' \
  -H 'accept: text/plain'
```

Request URL

`https://localhost:7157/api/Authors/1`

Server response

Code Details

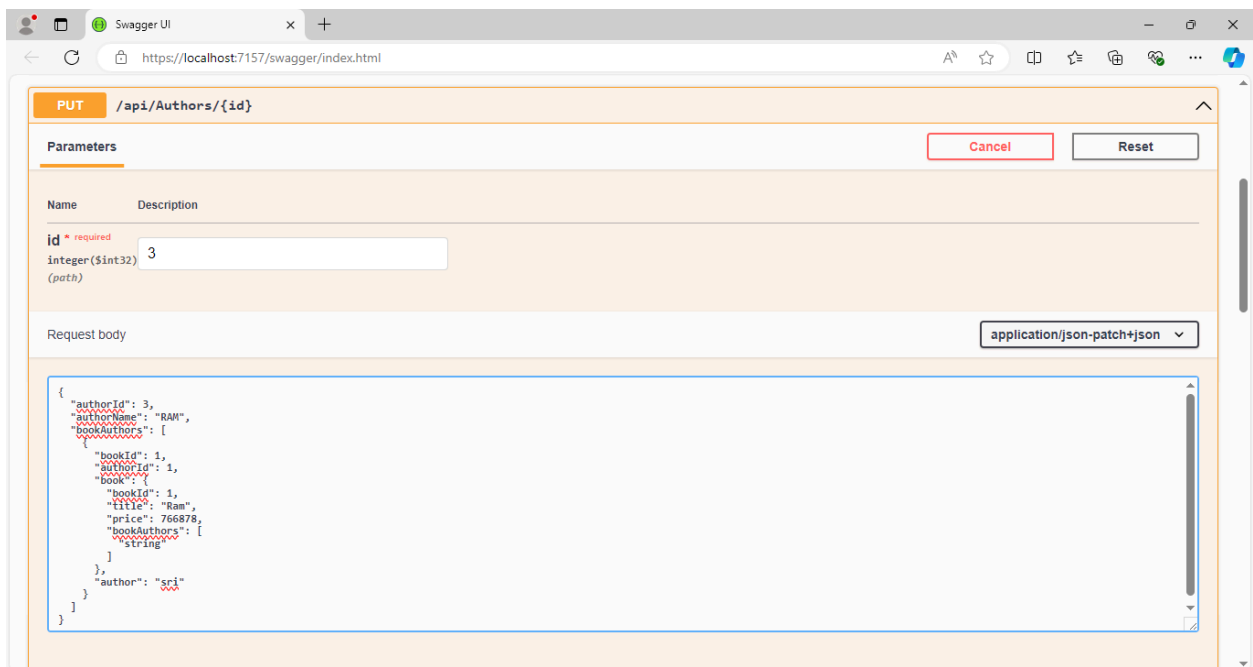
200

Response body

```
{
  "authorId": 1,
  "authorName": "Sam",
  "bookAuthors": null
}
```

Download

## PUT



The image shows the Swagger UI interface for a PUT endpoint. The endpoint is named `/api/Authors/{id}` and is located at `https://localhost:7157/swagger/index.html`. The parameter `id` is required and is of type `integer($int32)`. The user has entered the value `3` for the `id` parameter. The `Request body` is set to `application/json-patch+json`. The request body is a JSON object: `{ "authorId": 3, "authorName": "RAM", "bookAuthors": [ { "bookId": 1, "authorId": 1, "book": { "bookId": 1, "title": "Ram", "price": 766878, "bookAuthors": [ "string" ] } }, { "author": "sri" } ] }`. The `Cancel` and `Reset` buttons are visible.

**PUT** /api/Authors/{id}

Parameters

**id** \* required  
integer(\$int32)  
(path)

3

Request body

application/json-patch+json

```
{
  "authorId": 3,
  "authorName": "RAM",
  "bookAuthors": [
    {
      "bookId": 1,
      "authorId": 1,
      "book": {
        "bookId": 1,
        "title": "Ram",
        "price": 766878,
        "bookAuthors": [
          "string"
        ]
      }
    },
    {
      "author": "sri"
    }
  ]
}
```

Cancel Reset

## Delete

The image shows the Swagger UI for a DELETE endpoint. At the top, a status bar indicates a 200 Success response with no links. Below this, the endpoint is identified as **DELETE /api/Authors/{id}**. A 'Parameters' section contains a single path parameter named 'id', which is required and of type integer (\$int32). A text input field next to the parameter name contains the value '1'. An 'Execute' button is positioned below the parameters. At the bottom, a 'Responses' section shows a 200 Success response with no links.

200 Success No links

**DELETE** /api/Authors/{id}

Parameters Cancel

Name	Description
id <small>required</small>	

integer(\$int32)  
(path)

1

Execute

Responses

Code	Description	Links
200	Success	No links

## Book

## GET

The image shows the Swagger UI for a GET endpoint. It displays the curl command for the request, the request URL, and the server response. The response is a 200 status code with a JSON body containing book details. The response headers are also listed.

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7157/api/Books' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7157/api/Books

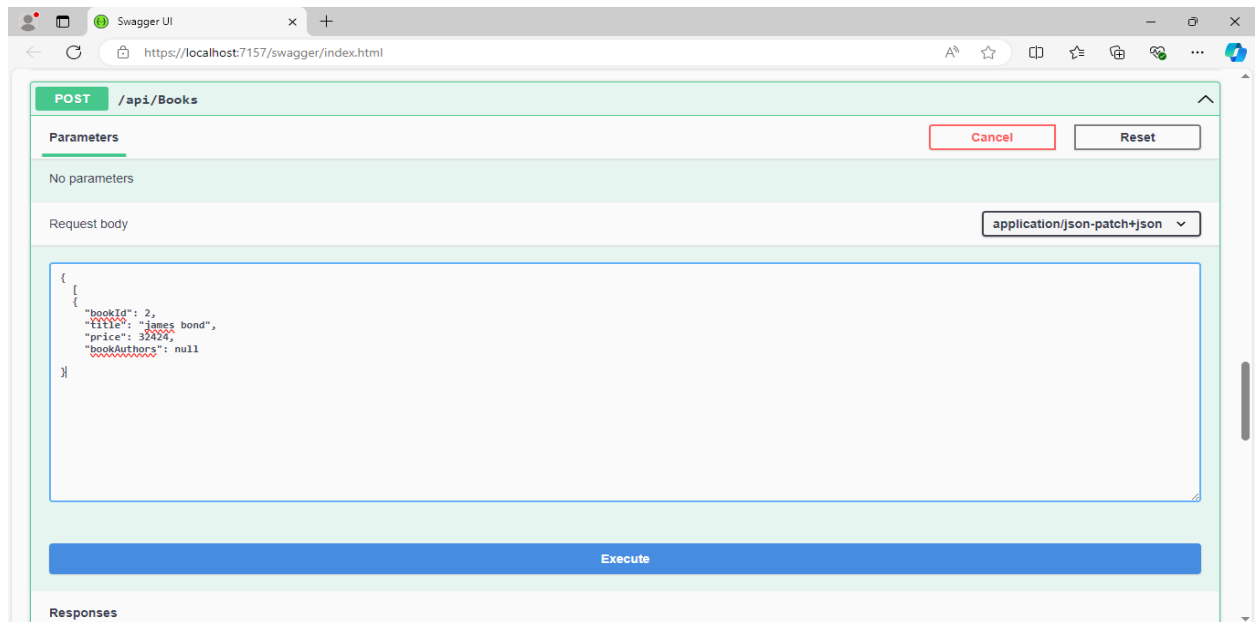
Server response

Code	Details
200	<p>Response body</p> <pre>{   "bookId": 1,   "title": "James",   "price": 0,   "bookAuthors": null }</pre> <p>Response headers</p> <pre>content-length: 59 content-type: application/json; charset=utf-8 date: Thu, 05 Sep 2024 05:15:09 GMT server: Kestrel</pre>

Responses

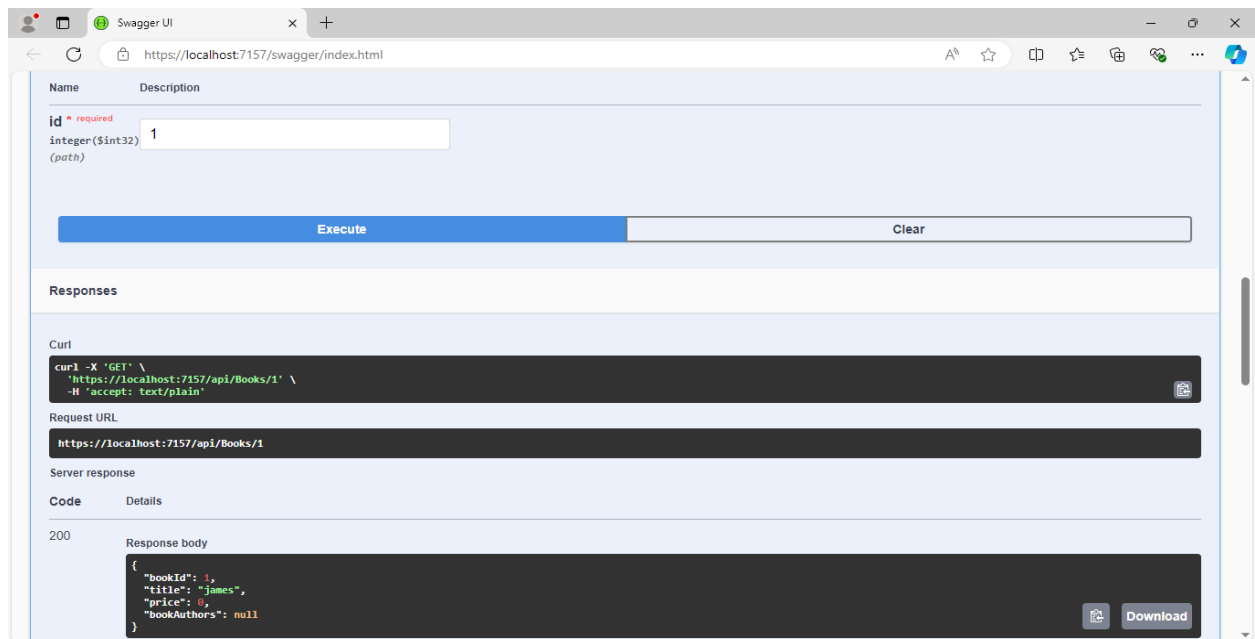
Code	Description	Links
------	-------------	-------

## POST



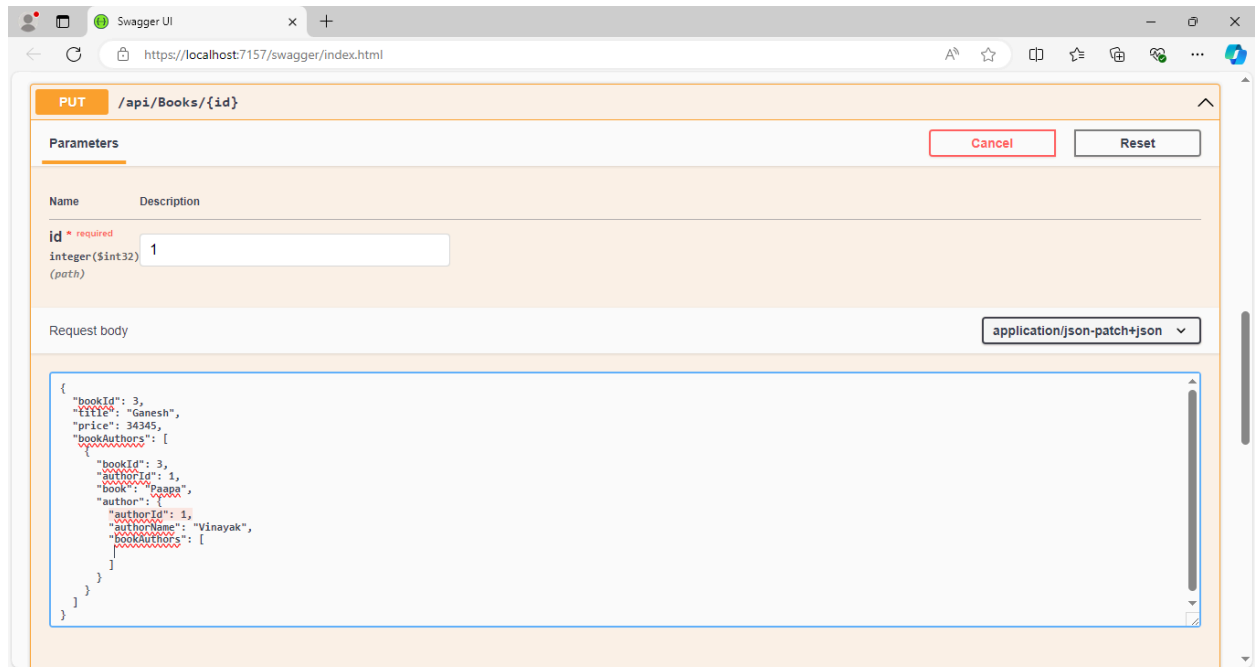
Swagger UI interface showing the configuration for a POST endpoint at `/api/Books`. The interface includes a "Parameters" section with "No parameters", a "Request body" section with a dropdown menu set to `application/json-patch+json`, and a large text area containing a JSON object: `{ "bookId": 2, "title": "james bond", "price": 32424, "bookAuthors": null }`. Below the text area is a blue "Execute" button. The "Responses" section is visible at the bottom.

## GET



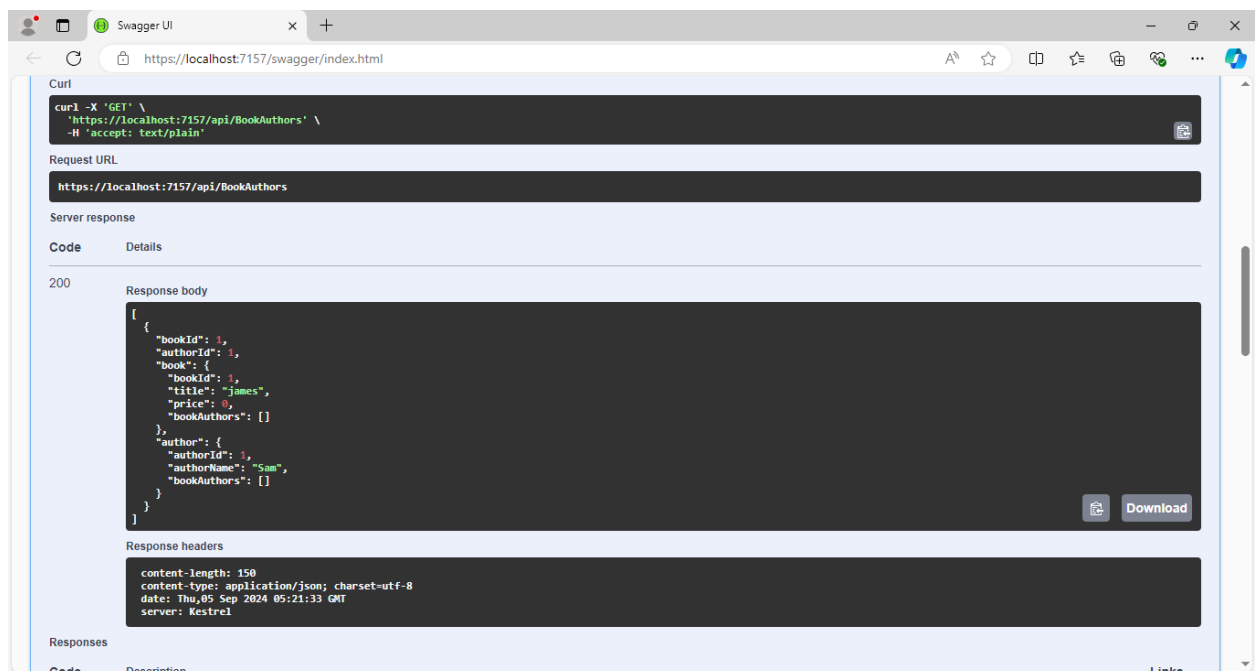
Swagger UI interface showing the configuration for a GET endpoint at `/api/Books/1`. The "Name" section shows a required parameter `id` of type `integer($int32)` with a value of `1`. Below this is a blue "Execute" button and a "Clear" button. The "Responses" section is expanded, showing a "Curl" command: `curl -X 'GET' \ 'https://localhost:7157/api/Books/1' \ -H 'accept: text/plain'`, a "Request URL" field with the value `https://localhost:7157/api/Books/1`, and a "Server response" section. The "Server response" section shows a "Code" of `200` and a "Response body" containing a JSON object: `{ "bookId": 1, "title": "james", "price": 0, "bookAuthors": null }`. A "Download" button is visible next to the response body.

## PUT

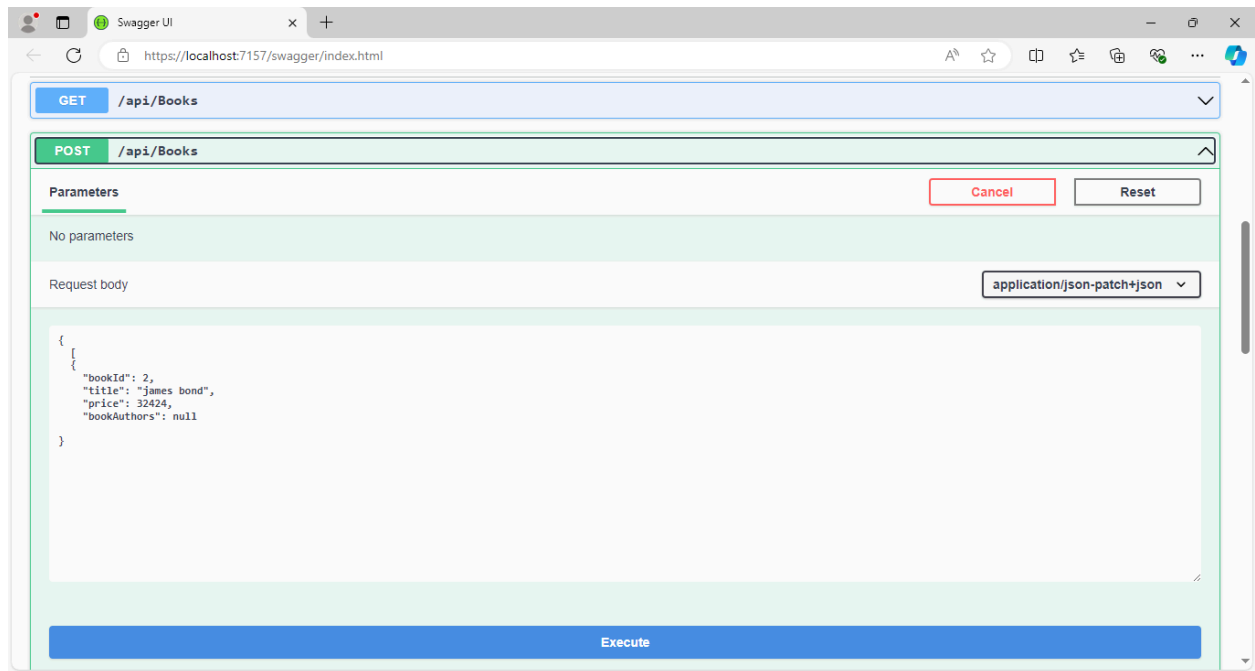


## BookAuthor

## GET



## POST

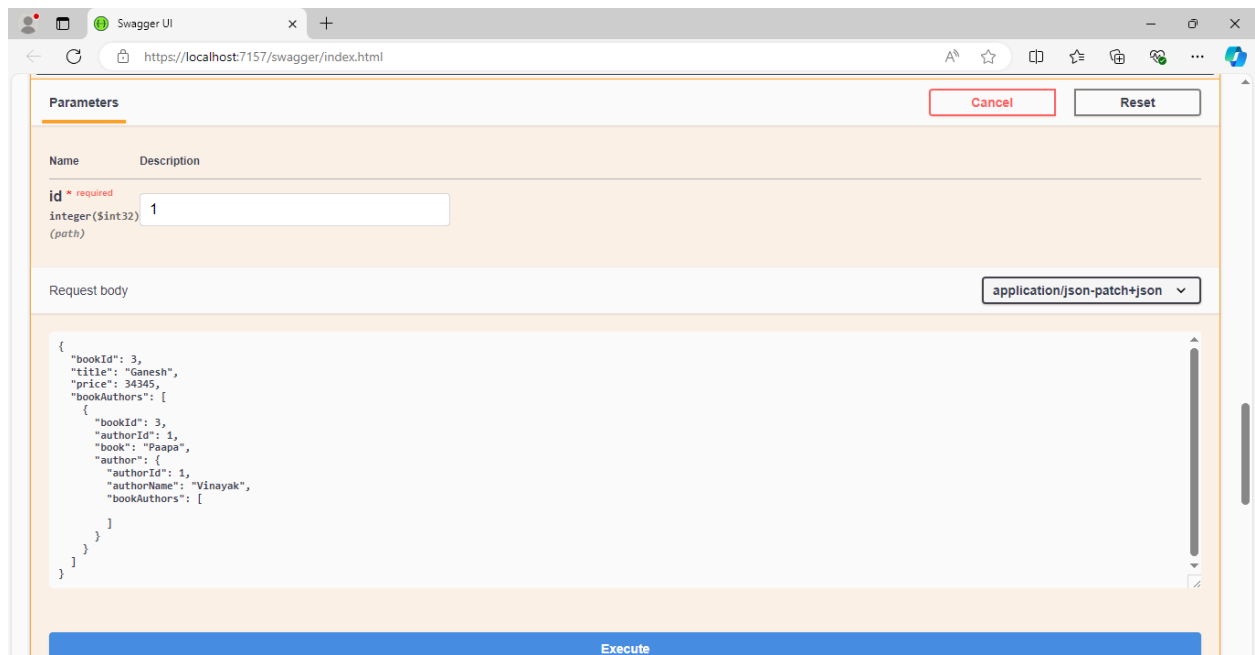


Swagger UI interface showing a POST request configuration for the endpoint `/api/Books`. The interface includes a "Parameters" section with "No parameters" and a "Request body" section with a dropdown menu set to `application/json-patch+json`. The request body contains a JSON object:

```
{
  {
    "bookId": 2,
    "title": "James bond",
    "price": 32424,
    "bookAuthors": null
  }
}
```

Buttons for "Cancel", "Reset", and "Execute" are visible.

## PUT



Swagger UI interface showing a PUT request configuration for the endpoint `/api/Books`. The interface includes a "Parameters" section with a table of parameters:

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

The `id` parameter is set to `1`. The "Request body" section has a dropdown menu set to `application/json-patch+json`. The request body contains a JSON object:

```
{
  "bookId": 3,
  "title": "Ganesh",
  "price": 34345,
  "bookAuthors": [
    {
      "bookId": 3,
      "authorId": 1,
      "book": "Paapa",
      "author": {
        "authorId": 1,
        "authorName": "Vinayak",
        "bookAuthors": [

```

Buttons for "Cancel", "Reset", and "Execute" are visible.

## DELETE

Swagger UI

https://localhost:7157/swagger/index.html

DELETE /api/Books/{id}

Parameters

Cancel

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

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'https://localhost:7157/api/Books/3' \
-H 'accept: */*'
```

Request URL

```
https://localhost:7157/api/Books/3
```

Server response

Code	Details
404	