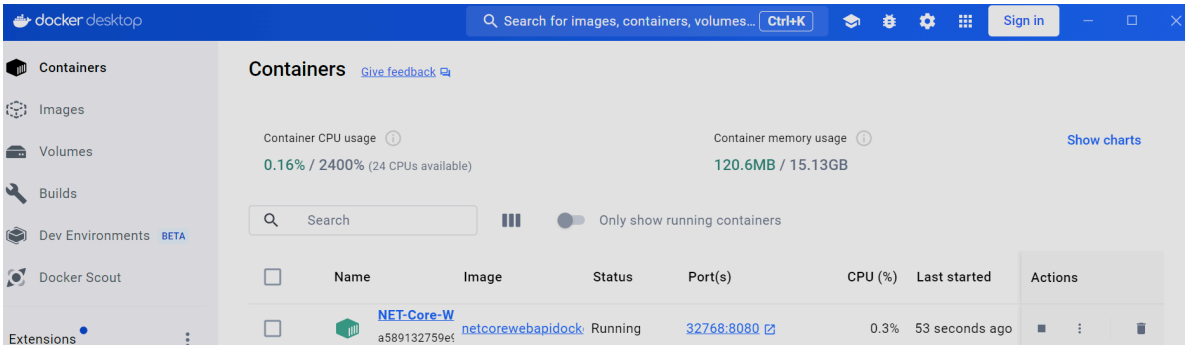
# **Exercise 1**: Create a dockerized ASP.NET API application.

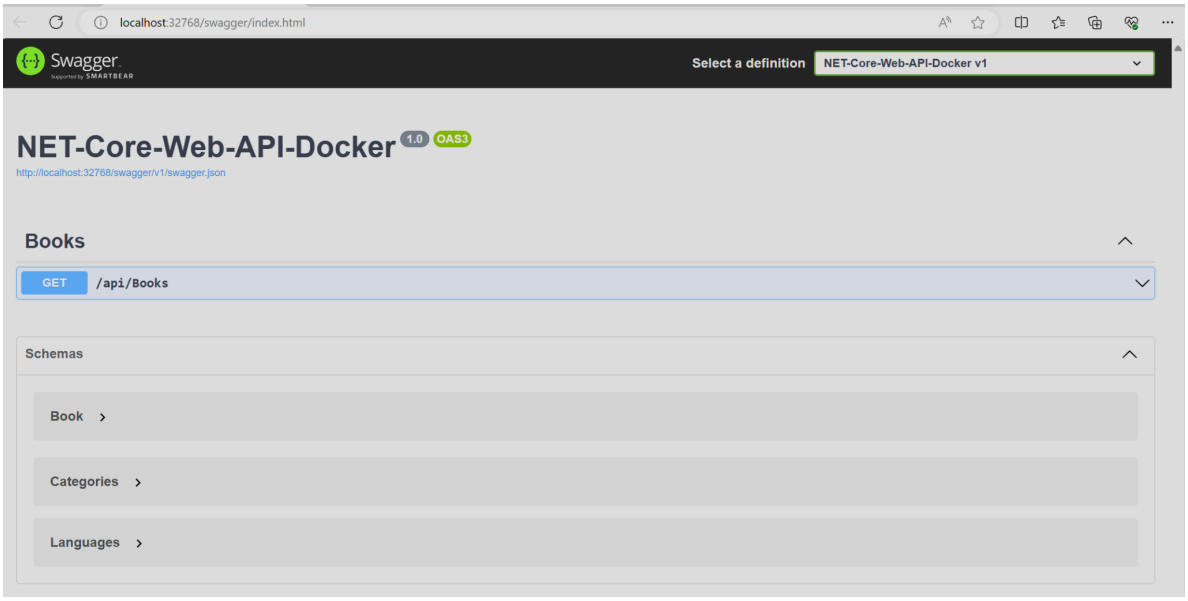
## **Statement**:

Create a dockerized ASP.NET web API application with a single endpoint to return a list of books. The books can be a static in-memory collection, and have names and publish dates.

## Hints for Implementation:

Adding docker: You can use the “add docker support” visual studio feature to create the required docker files.



Curl

```
curl -X 'GET' \
  'http://localhost:32768/api/Books' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:32768/api/Books
```

Server response

| Code | Details |
|---|---|
| 200 | Response body |


```
[
  {
    "id": 1,
    "category": 0,
    "title": "How to Create Action Fantasy and Adventure Comics",
    "publisher": "North Ligth Books",
    "edition": 1,
    "editionDate": "1996-04-02T00:00:00",
    "language": 0,
    "paperbackPages": 143,
    "isbN_10": "0891346616",
    "isbN_13": "9780891346616",
    "weight": 0.7,
    "dimensions": "25 x 15 x 1.25"
  },
  {
    "id": 2,
    "category": 0,
    "title": "How to Create Action Fantasy and Adventure Comics",
    "publisher": "Atheuneau",
    "edition": 2,
    "editionDate": "1988-11-12T00:00:00",
    "language": 5,
    "paperbackPages": 363,
    "isbN_10": "0000000000",
    "isbN_13": "0000000000000",
    "weight": 0.87,
```


Download

Response headers

```
content-type: application/json; charset=utf-8
date: Tue,09 Apr 2024 00:02:05 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

| Code | Description | Links |
|---|---|---|
| 200 | Success | *No links* |

# **Exercise 2**: Create a dockerized ASP.NET MVC application.

### Statement:

Create a dockerized ASP.NET MVC application that retrieves the books from the API built on the first exercise, and displays a page with the list of books.

# **Exercise 3**: Implement docker compose.

## Statement:

Create a docker compose file that enables both applications to run and communicate with each other.

## Hints for Implementation:

- **Adding docker compose:** You can use the “add container orchestrator support” visual studio feature to add each application to the compose file.
- **Making the API internal:** Services inside the same compose file use the same network by default, you can use the service name as the hostname to make the HTTP requests from the web application to the API