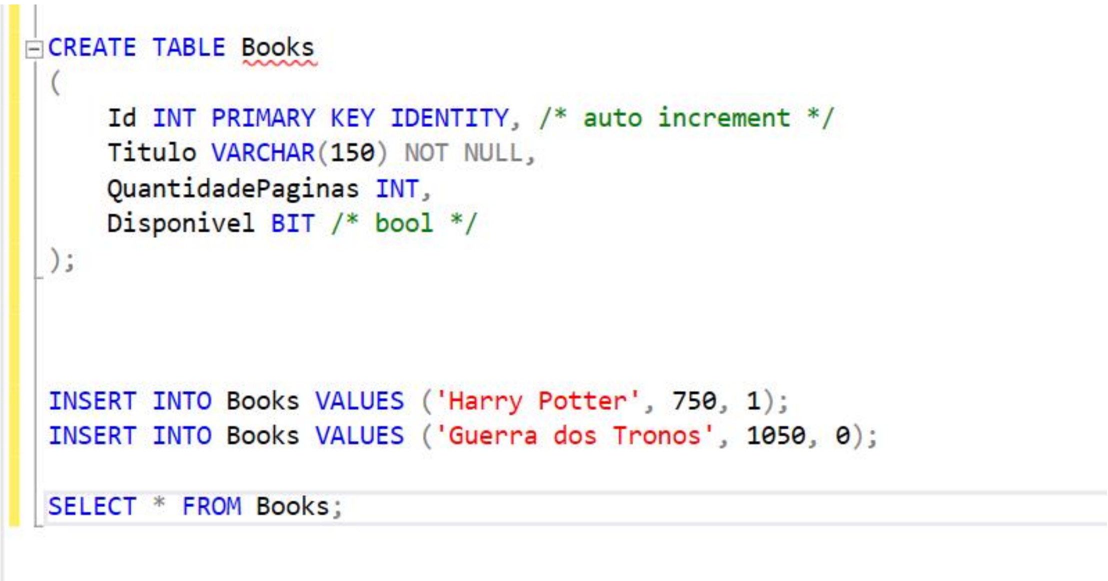
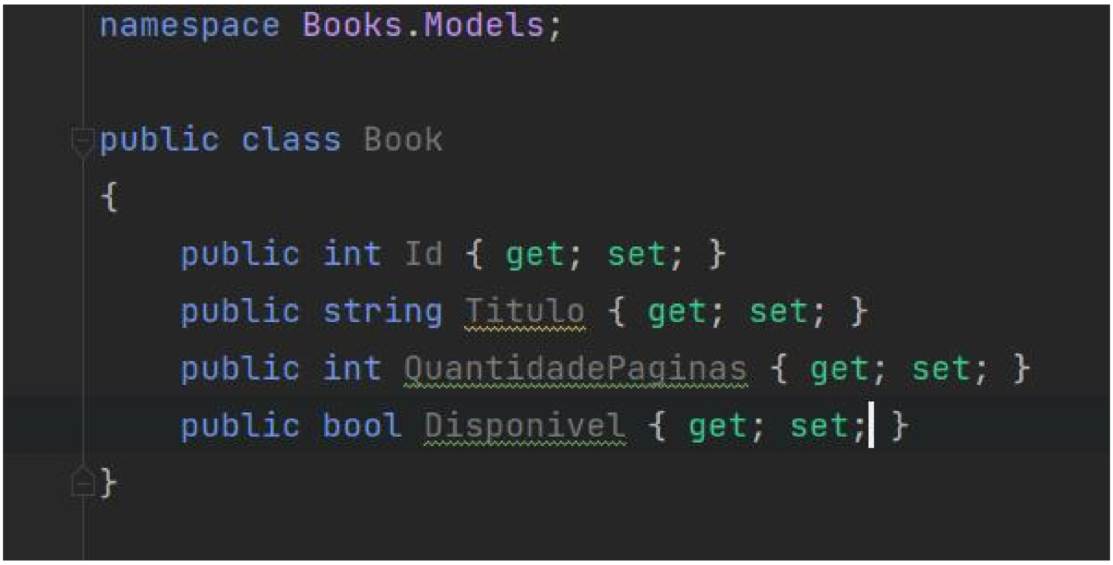
Database



```sql
CREATE TABLE Books
(
    Id INT PRIMARY KEY IDENTITY, /* auto increment */
    Titulo VARCHAR(150) NOT NULL,
    QuantidadePaginas INT,
    Disponivel BIT /* bool */
);


INSERT INTO Books VALUES ('Harry Potter', 750, 1);
INSERT INTO Books VALUES ('Guerra dos Tronos', 1050, 0);

SELECT * FROM Books;
```

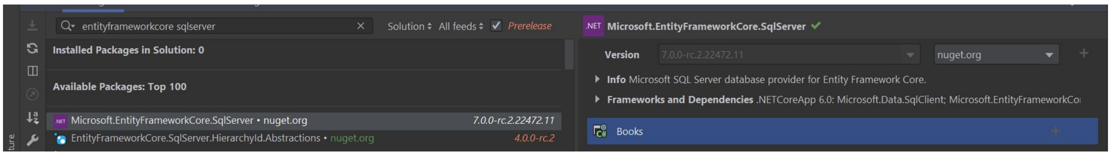
Book model



```csharp
namespace Books.Models;

public class Book
{
    public int Id { get; set; }
    public string Titulo { get; set; }
    public int QuantidadePaginas { get; set; }
    public bool Disponivel { get; set; }
}
```

Instalação do entity framework

Ligação entre o código e o banco de dados. Já fornece vários métodos que podem ser usados para lidar com os dados do BD



entityframeworkcore sqlserver

Solution All feeds ☑ Prerelease

Installed Packages in Solution: 0

Available Packages: Top 100

Microsoft.EntityFrameworkCore.SqlServer • nuget.org 7.0.0-rc.2.22472.11

EntityFrameworkCore.SqlServer.HierarchyId.Abstractions • nuget.org 4.0.0-rc.2

.NET Microsoft.EntityFrameworkCore.SqlServer ✓

Version 7.0.0-rc.2.22472.11 nuget.org

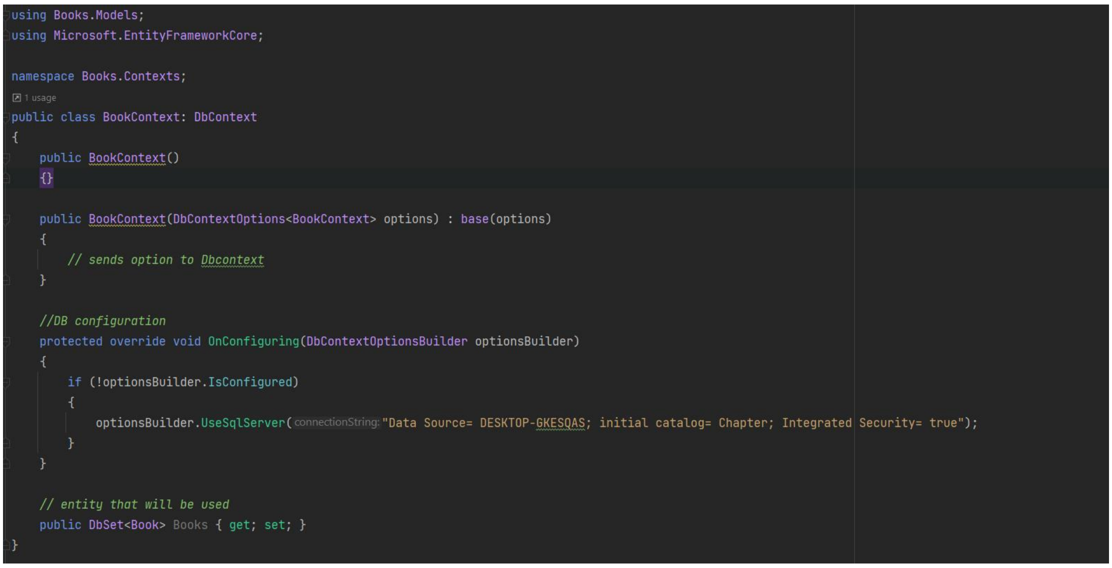
Info Microsoft SQL Server database provider for Entity Framework Core.

Frameworks and Dependencies .NETCoreApp 6.0: Microsoft.Data.SqlClient; Microsoft.EntityFrameworkCo

Books

CONTEXT

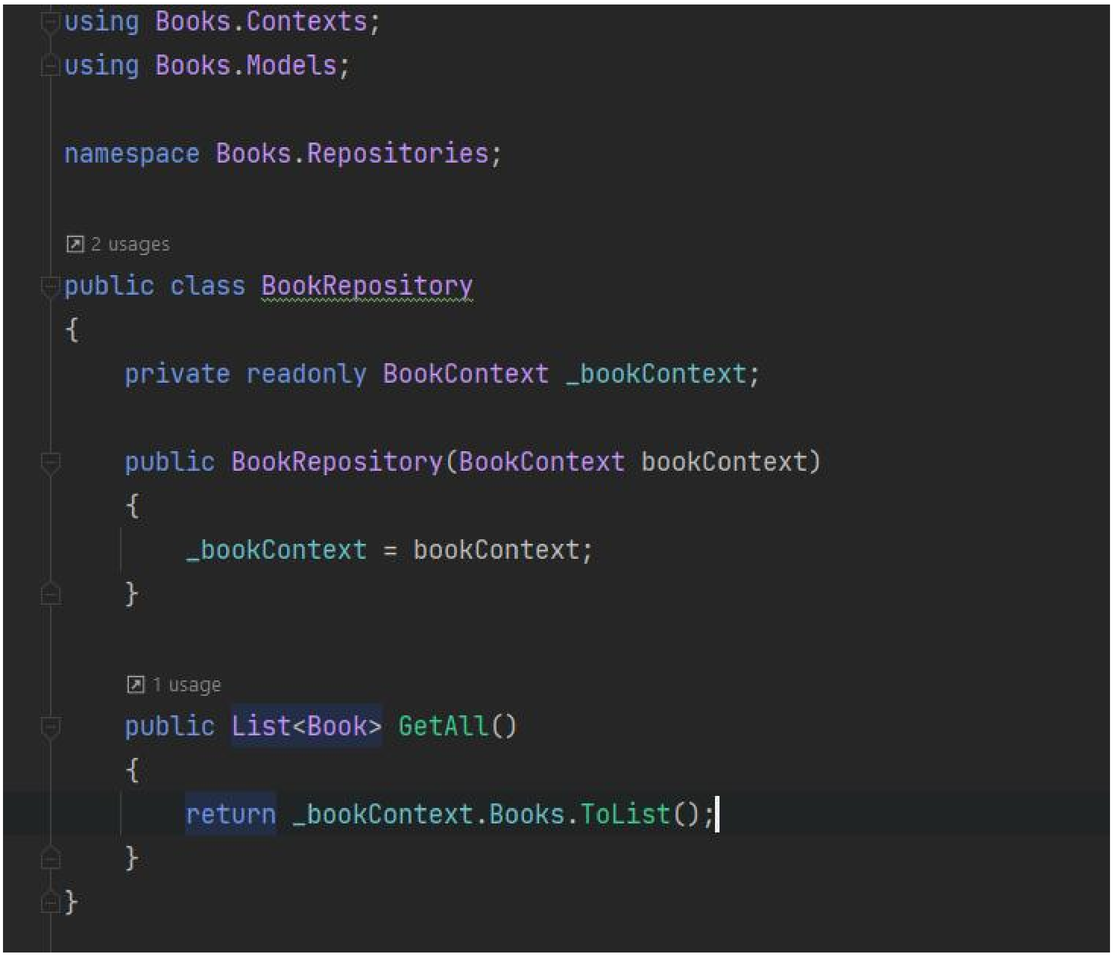
Configuração do banco de dados



```
using Books.Models;
using Microsoft.EntityFrameworkCore;

namespace Books.Contexts;
1 usage
public class BookContext: DbContext
{
    public BookContext()
    {}

    public BookContext(DbContextOptions<BookContext> options) : base(options)
    {
        // sends option to Dbcontext
    }

    //DB configuration
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        if (!optionsBuilder.IsConfigured)
        {
            optionsBuilder.UseSqlServer( connectionString: "Data Source= DESKTOP-GKESQAS; initial catalog= Chapter; Integrated Security= true");
        }
    }

    // entity that will be used
    public DbSet<Book> Books { get; set; }
}
```

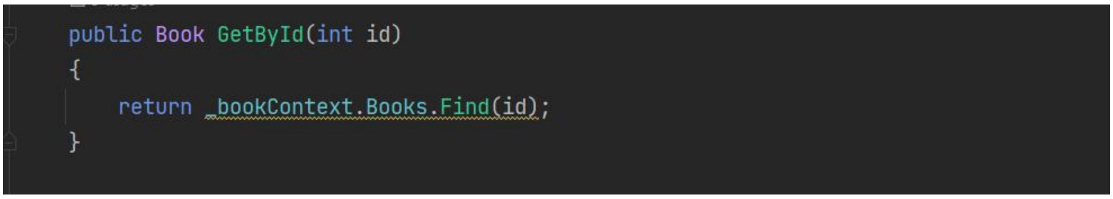
Repositories

GetAll



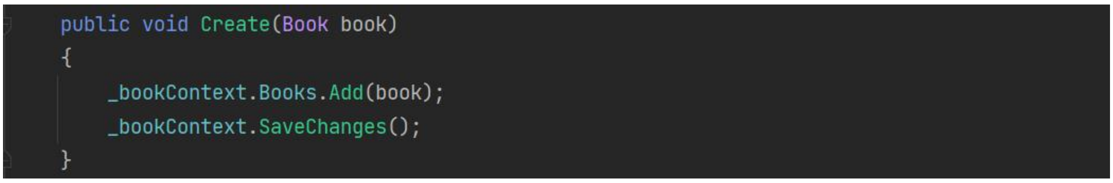
```
using Books.Contexts;
using Books.Models;

namespace Books.Repositories;

2 usages
public class BookRepository
{
    private readonly BookContext _bookContext;

    public BookRepository(BookContext bookContext)
    {
        _bookContext = bookContext;
    }

    1 usage
    public List<Book> GetAll()
    {
        return _bookContext.Books.ToList();
    }
}
```

GetById



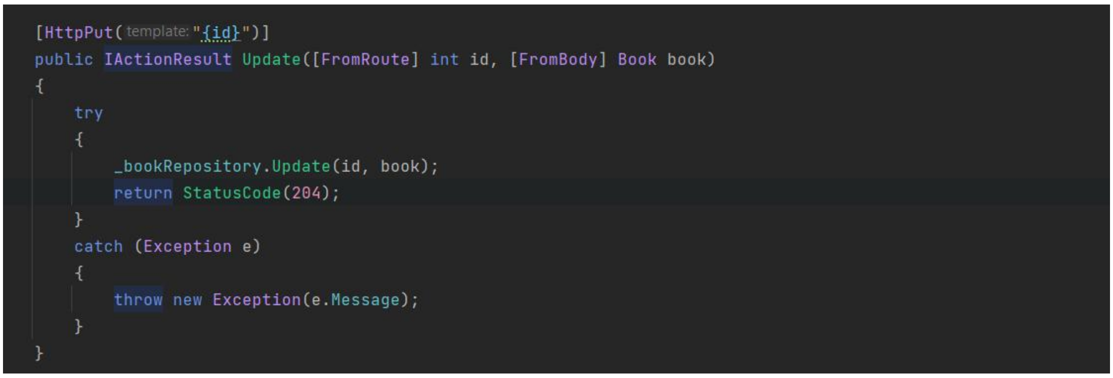
```
public Book GetById(int id)
{
    return _bookContext.Books.Find(id);
}
```

Create



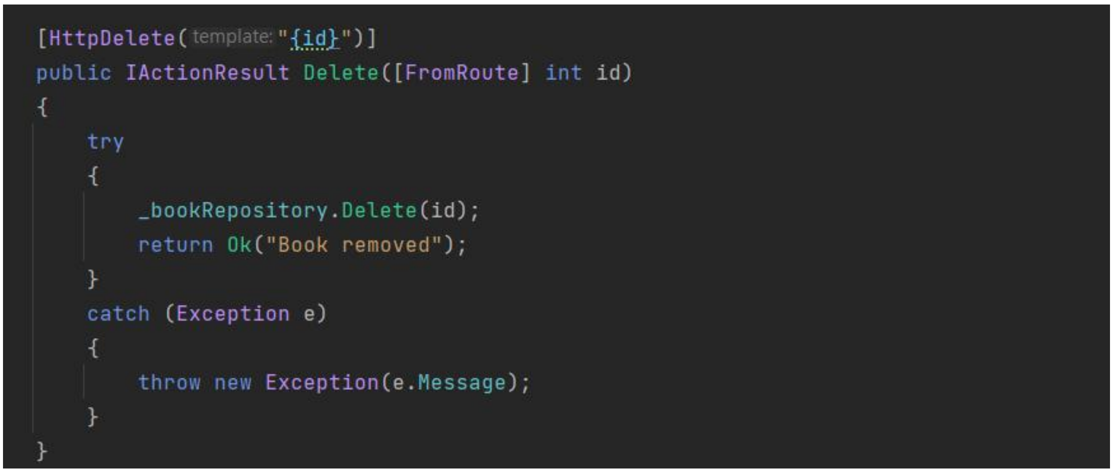
```
public void Create(Book book)
{
    _bookContext.Books.Add(book);
    _bookContext.SaveChanges();
}
```

Update



```
[HttpPut( template: "{id}")]
public IActionResult Update([FromRoute] int id, [FromBody] Book book)
{
    try
    {
        _bookRepository.Update(id, book);
        return StatusCode(204);
    }
    catch (Exception e)
    {
        throw new Exception(e.Message);
    }
}
```

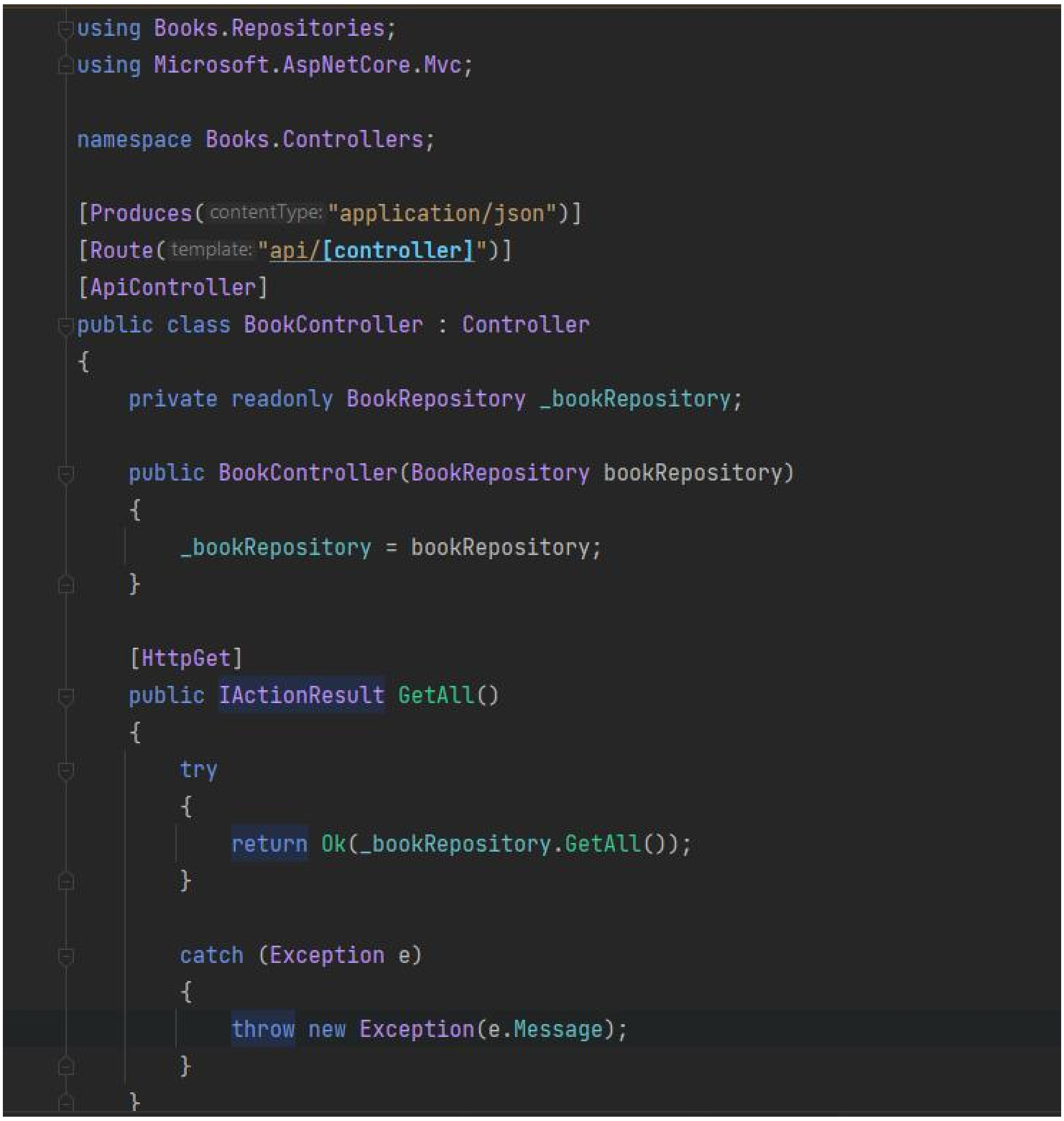
Delete



```
[HttpDelete( template: "{id}")]
public IActionResult Delete([FromRoute] int id)
{
    try
    {
        _bookRepository.Delete(id);
        return Ok("Book removed");
    }
    catch (Exception e)
    {
        throw new Exception(e.Message);
    }
}
```

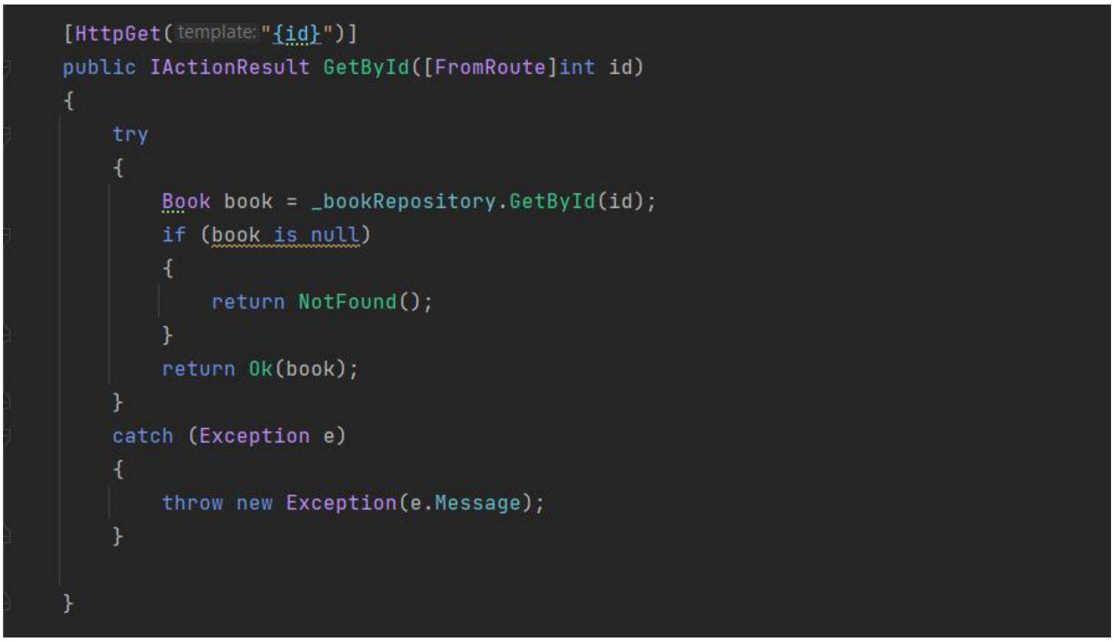
Controller

GetAll



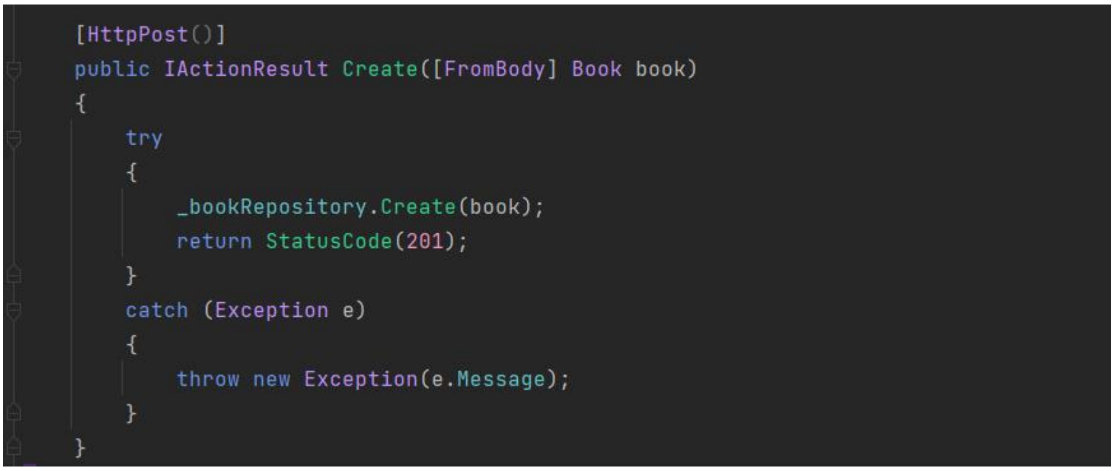
```
using Books.Repositories;
using Microsoft.AspNetCore.Mvc;

namespace Books.Controllers;

[Produces( contentType: "application/json")]
[Route( template: "api/[controller]")]
[ApiController]
public class BookController : Controller
{
    private readonly BookRepository _bookRepository;

    public BookController(BookRepository bookRepository)
    {
        _bookRepository = bookRepository;
    }

    [HttpGet]
    public IActionResult GetAll()
    {
        try
        {
            return Ok(_bookRepository.GetAll());
        }

        catch (Exception e)
        {
            throw new Exception(e.Message);
        }
    }
```

GetById



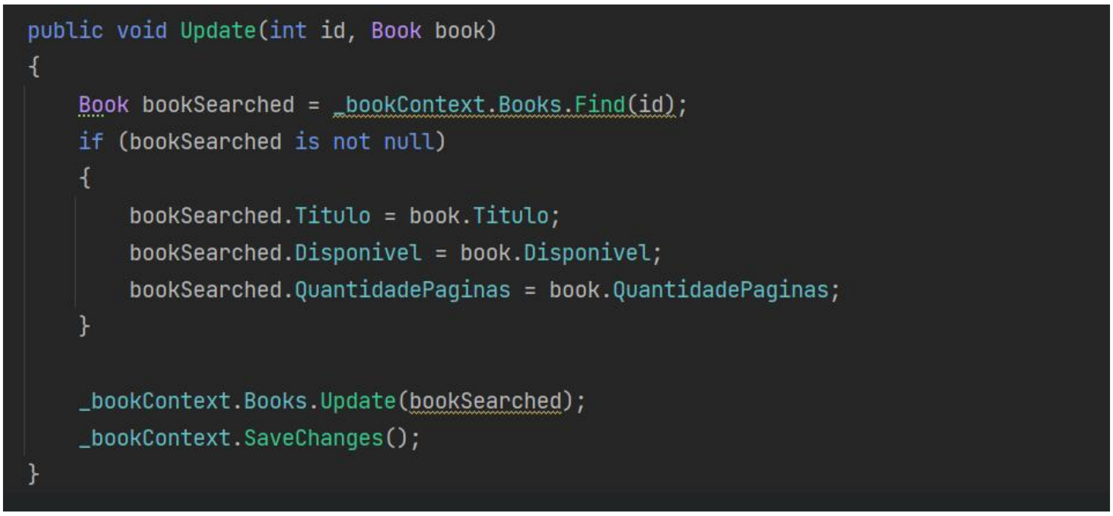
```
[HttpGet( template: "{id}")]
public IActionResult GetById([FromRoute]int id)
{
    try
    {
        Book book = _bookRepository.GetById(id);
        if (book is null)
        {
            return NotFound();
        }
        return Ok(book);
    }
    catch (Exception e)
    {
        throw new Exception(e.Message);
    }

}
```

Create



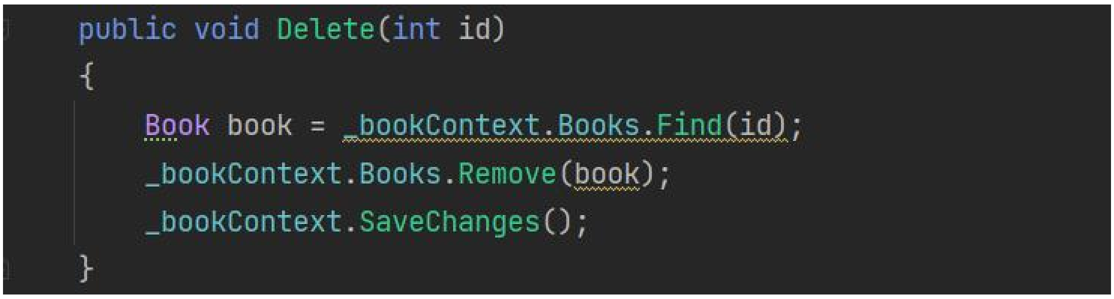
```
[HttpPost()]
public IActionResult Create([FromBody] Book book)
{
    try
    {
        _bookRepository.Create(book);
        return StatusCode(201);
    }
    catch (Exception e)
    {
        throw new Exception(e.Message);
    }
}
```

Update



```
public void Update(int id, Book book)
{
    Book bookSearched = _bookContext.Books.Find(id);
    if (bookSearched is not null)
    {
        bookSearched.Titulo = book.Titulo;
        bookSearched.Disponivel = book.Disponivel;
        bookSearched.QuantidadePaginas = book.QuantidadePaginas;
    }

    _bookContext.Books.Update(bookSearched);
    _bookContext.SaveChanges();
}
```

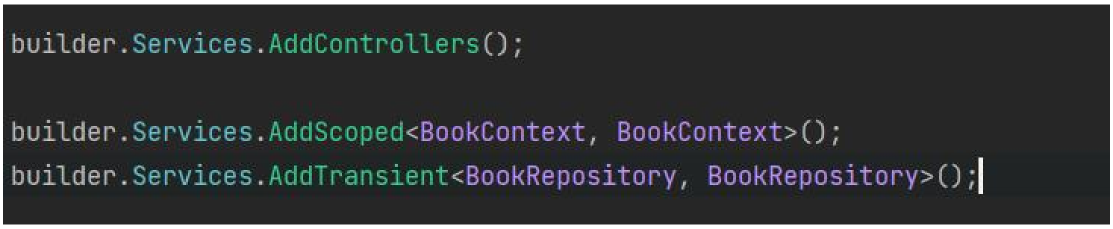
Delete



```
    public void Delete(int id)
    {
        Book book = _bookContext.Books.Find(id);
        _bookContext.Books.Remove(book);
        _bookContext.SaveChanges();
    }
```

Program.cs

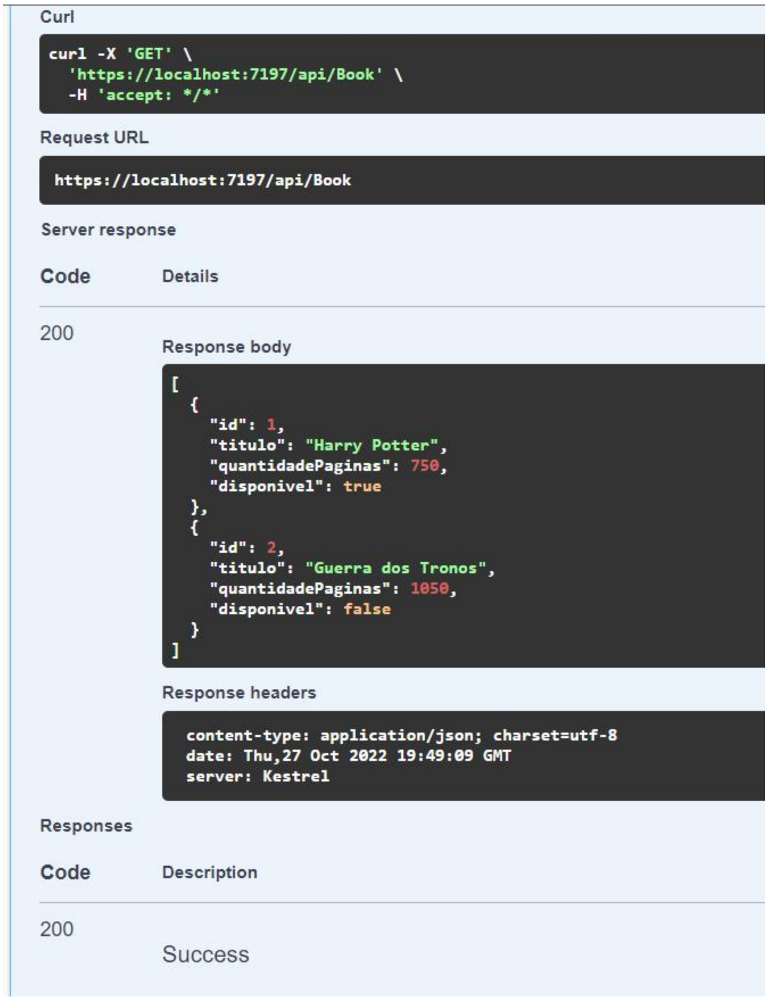
Adicionas as ultimas 2 linhas



```
builder.Services.AddControllers();

builder.Services.AddScoped<BookContext, BookContext>();
builder.Services.AddTransient<BookRepository, BookRepository>();
```

Swagger

GetAll



Curl

```
curl -X 'GET' \
  'https://localhost:7197/api/Book' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7197/api/Book
```

Server response

Code	Details
200	Response body

```
[
  {
    "id": 1,
    "titulo": "Harry Potter",
    "quantidadePaginas": 750,
    "disponivel": true
  },
  {
    "id": 2,
    "titulo": "Guerra dos Tronos",
    "quantidadePaginas": 1050,
    "disponivel": false
  }
]
```

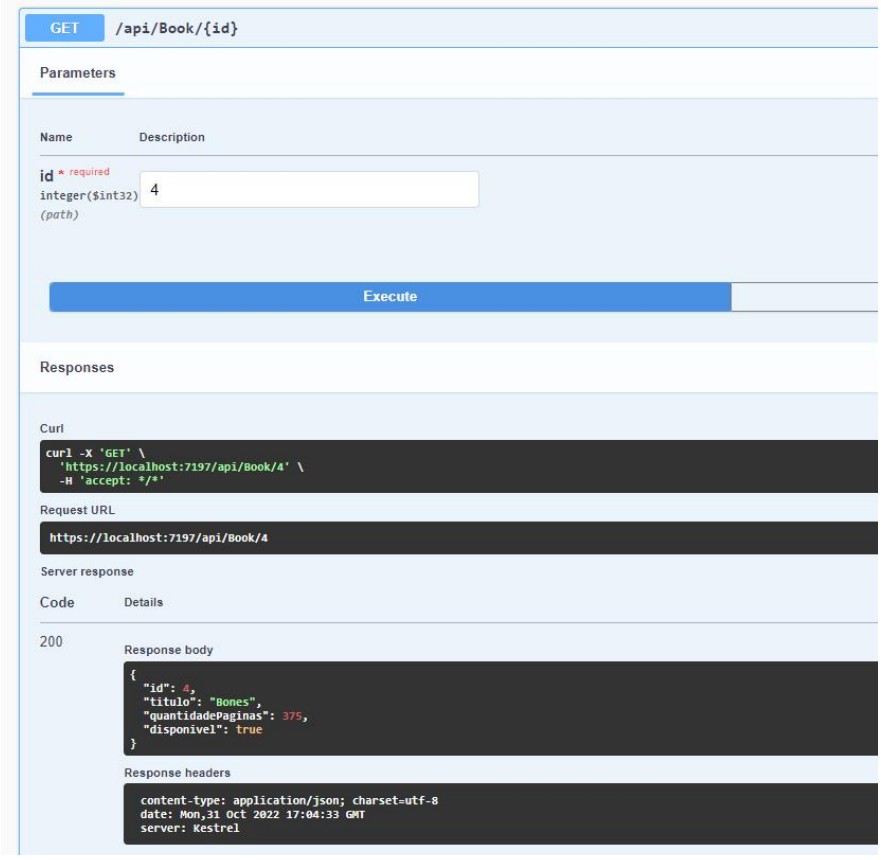
Response headers

```
content-type: application/json; charset=utf-8
date: Thu,27 Oct 2022 19:49:09 GMT
server: Kestrel
```

Responses

Code	Description
200	Success

GetById



/api/Book/{id}

id * required
integer($int32)
(path)
4

Curl

```
curl -X 'GET' \
  'https://localhost:7197/api/Book/4' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7197/api/Book/4
```

Server response

200

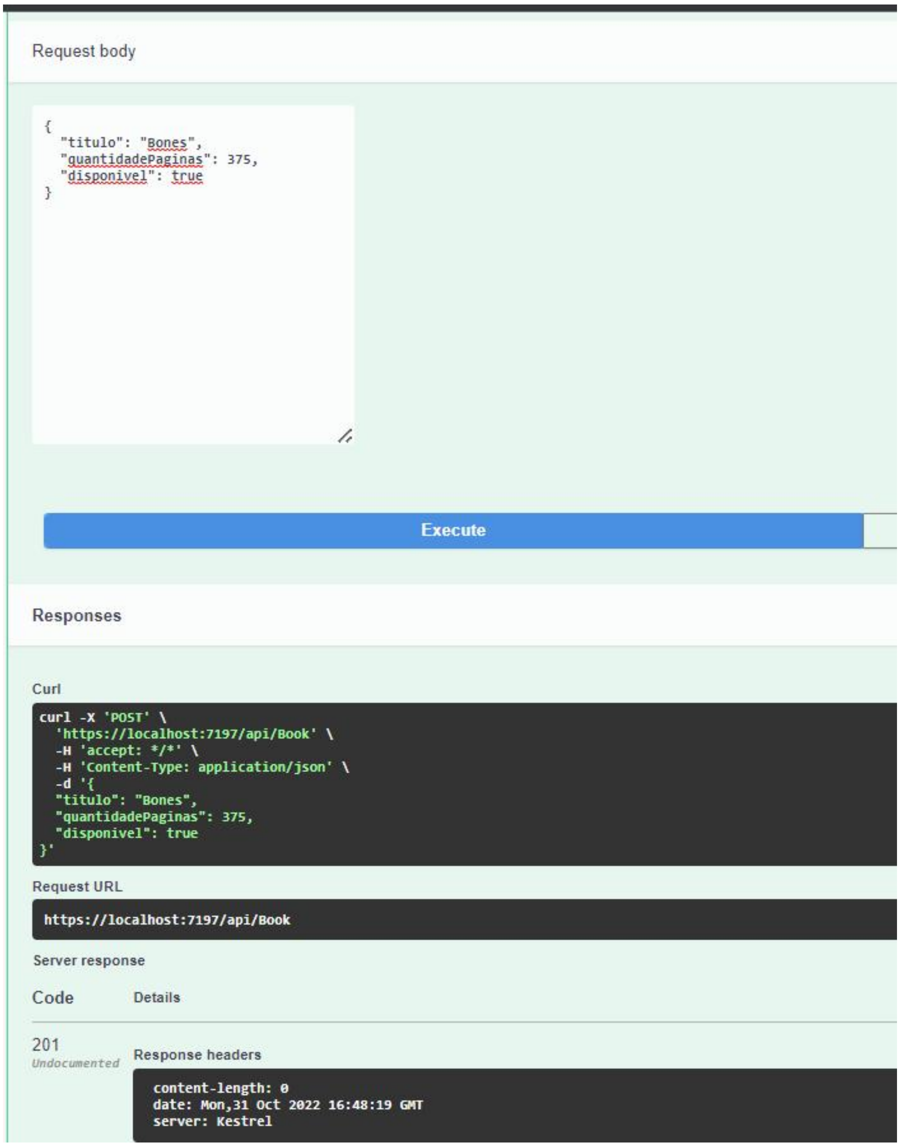
Response body

```
{
  "id": 4,
  "titulo": "Bones",
  "quantidadePaginas": 375,
  "disponivel": true
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Mon,31 Oct 2022 17:04:33 GMT
server: Kestrel
```

Create



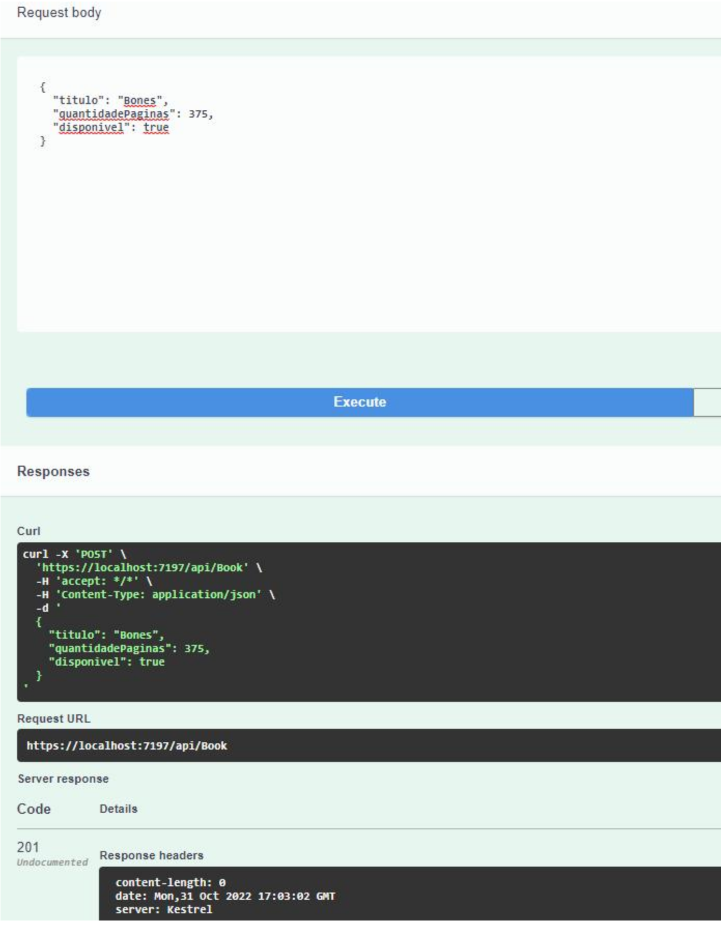
```
{
  "titulo": "Bones",
  "quantidadePaginas": 375,
  "disponivel": true
}
```

```
curl -X 'POST' \
  'https://localhost:7197/api/Book' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "titulo": "Bones",
  "quantidadePaginas": 375,
  "disponivel": true
}'
```

```
https://localhost:7197/api/Book
```

201
Undocumented

```
content-length: 0
date: Mon,31 Oct 2022 16:48:19 GMT
server: Kestrel
```



Request body

```
{
  "titulo": "Bones",
  "quantidadePaginas": 375,
  "disponivel": true
}
```

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7197/api/Book' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '
  {
    "titulo": "Bones",
    "quantidadePaginas": 375,
    "disponivel": true
  }
'
```

Request URL

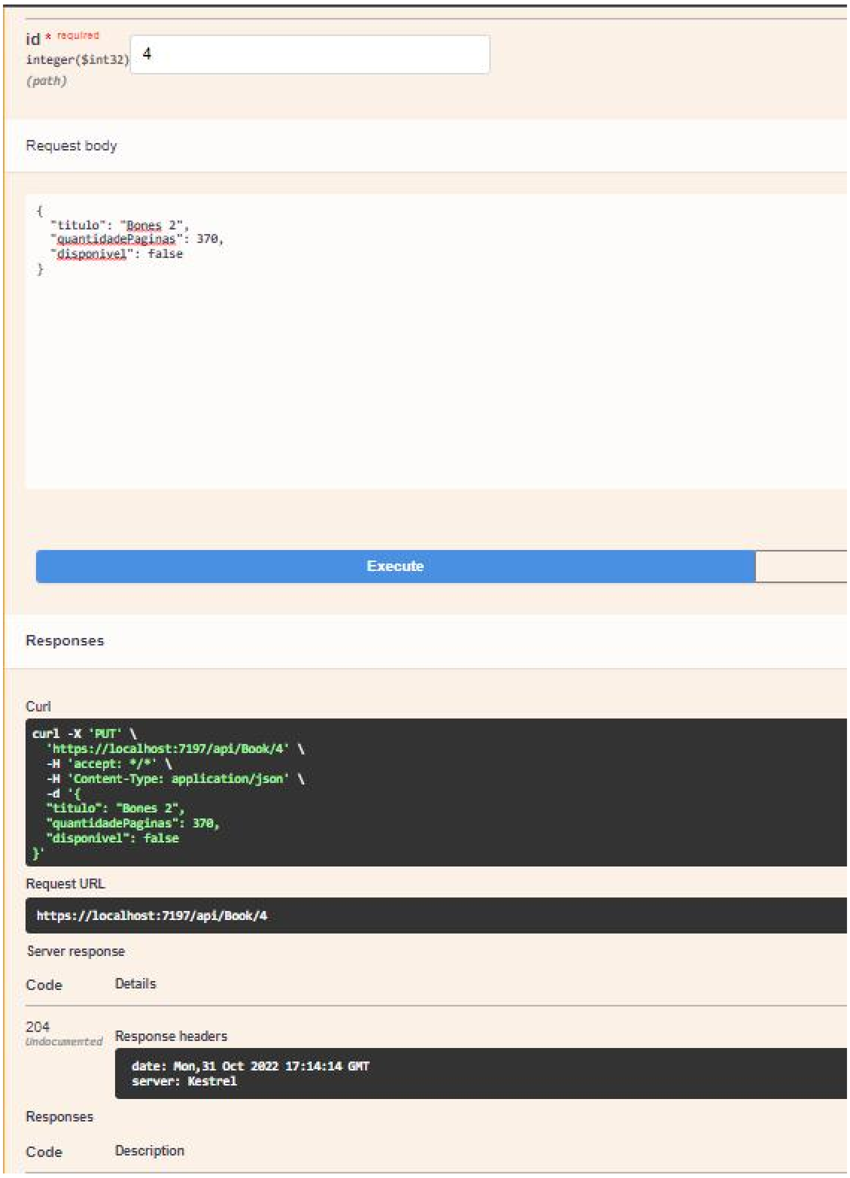
```
https://localhost:7197/api/Book
```

Server response

Code	Details
201 *Undocumented*	Response headers

```
content-length: 0
date: Mon,31 Oct 2022 17:03:02 GMT
server: Kestrel
```

Update



id * required
integer($int32)
(path)

4

Request body

```
{
  "titulo": "Bones 2",
  "quantidadePaginas": 370,
  "disponivel": false
}
```

Responses

Curl

```
curl -X 'PUT' \
  'https://localhost:7197/api/Book/4' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "titulo": "Bones 2",
  "quantidadePaginas": 370,
  "disponivel": false
}'
```

Request URL

```
https://localhost:7197/api/Book/4
```

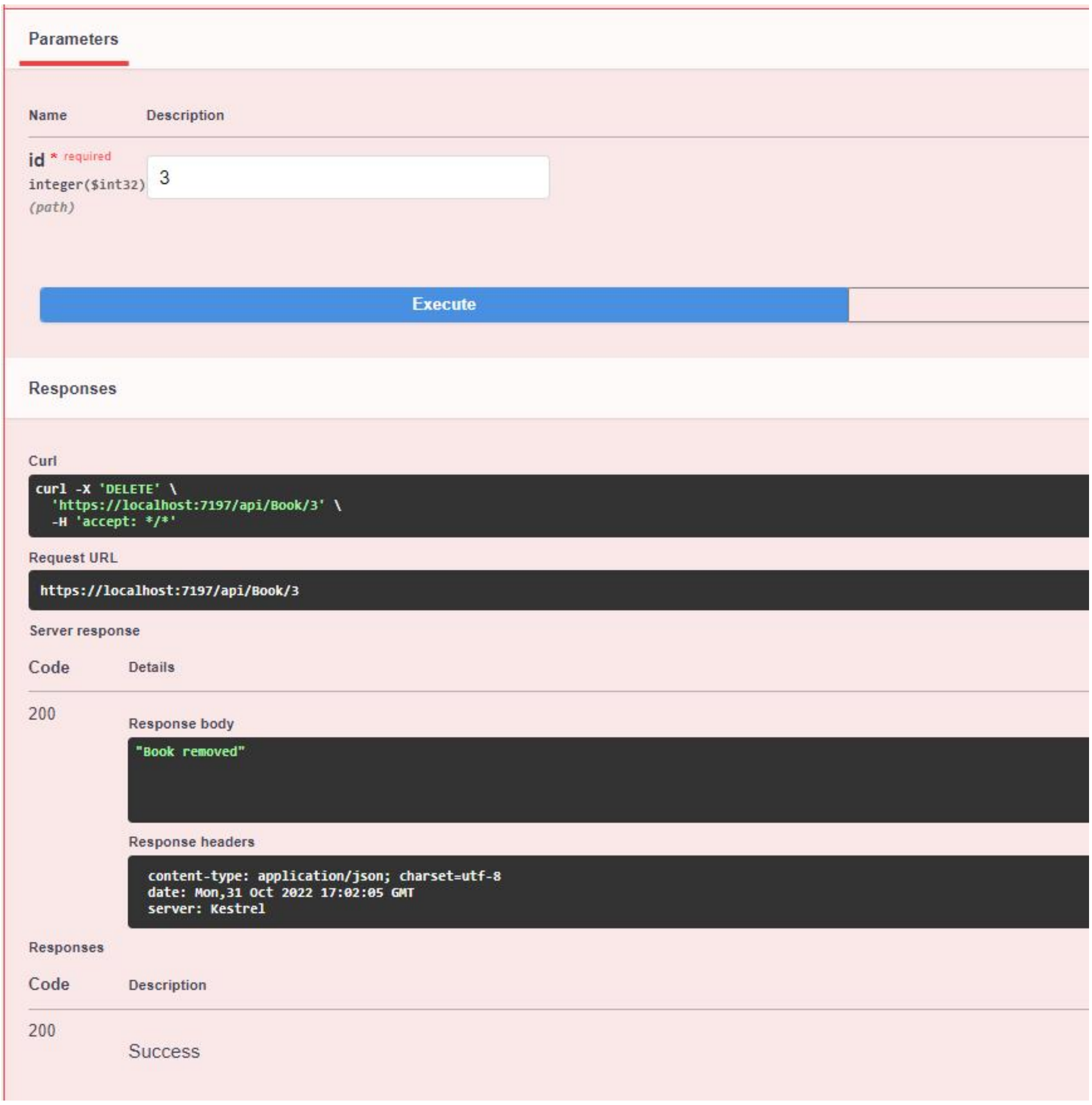
Server response

Code	Details
204 Undocumented	Response headers date: Mon,31 Oct 2022 17:14:14 GMT server: Kestrel

Responses

Code	Description

Delete



Name	Description
id * required integer($int32) *(path)*	3

Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7197/api/Book/3' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7197/api/Book/3
```

Server response

Code	Details
200	Response body "Book removed" Response headers content-type: application/json; charset=utf-8 date: Mon,31 Oct 2022 17:02:05 GMT server: Kestrel

Responses

Code	Description
200	Success