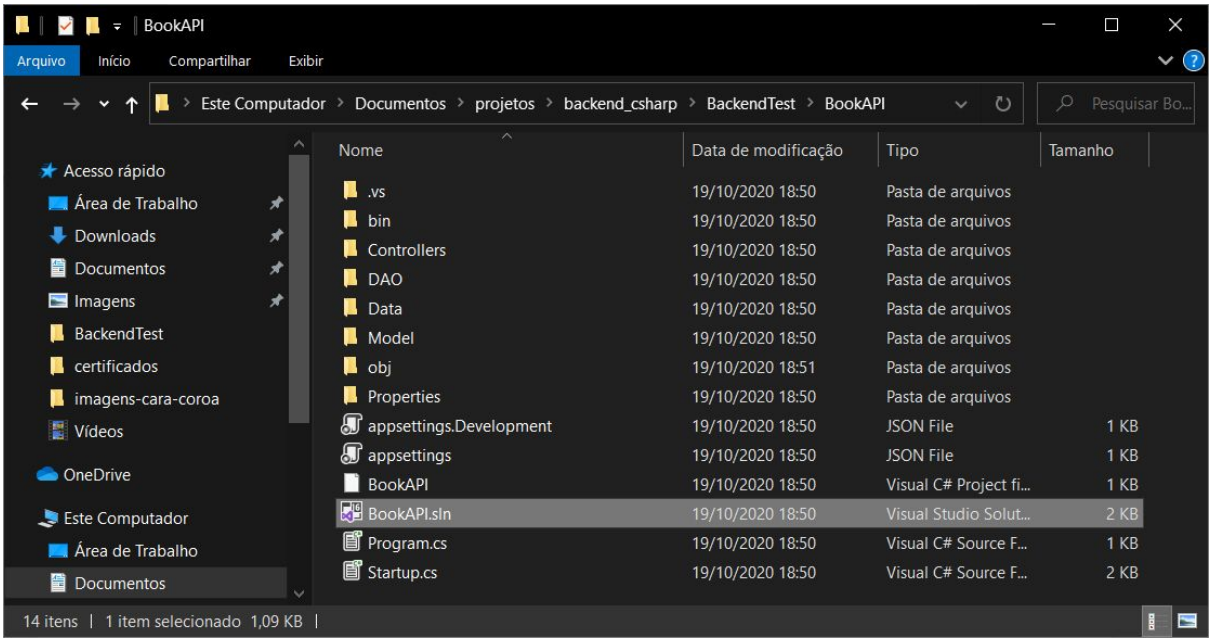
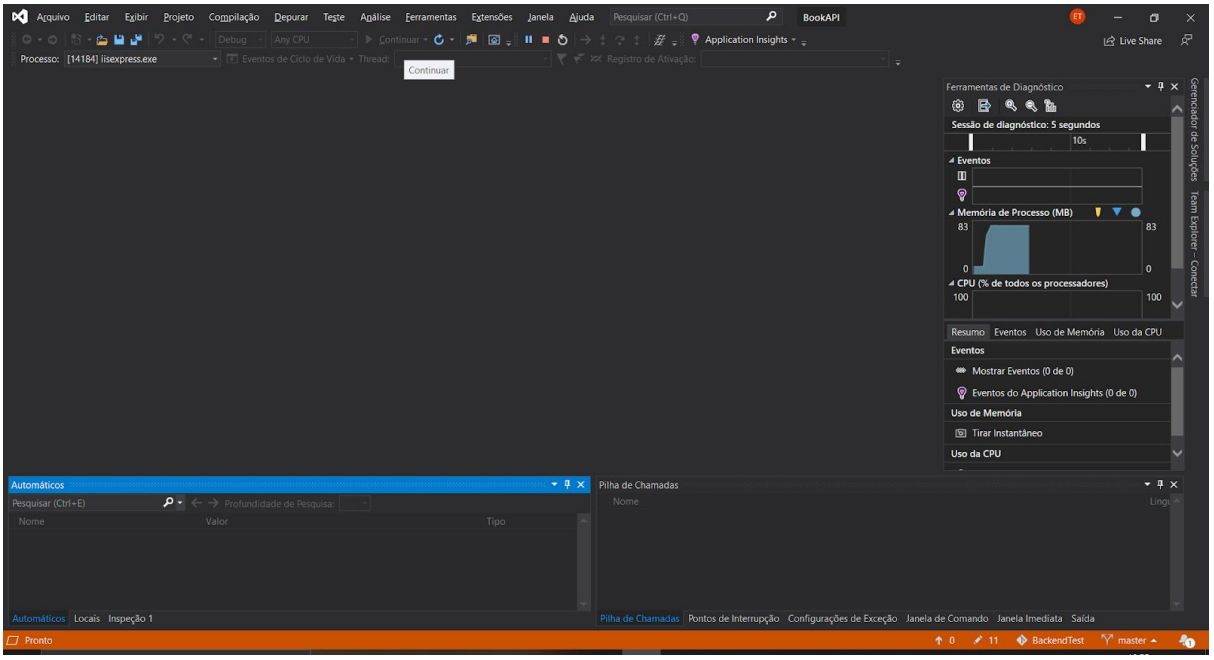
O sistema é uma api que gerencia algumas operações relacionadas a busca de livros.
Nesse sistema foi utilizado C# com API ASP.NET, para a execução do sistema foi utilizado a IDE Visual Studio e a ferramenta Postman.

A seguir será apresentada as funcionalidades do sistema.

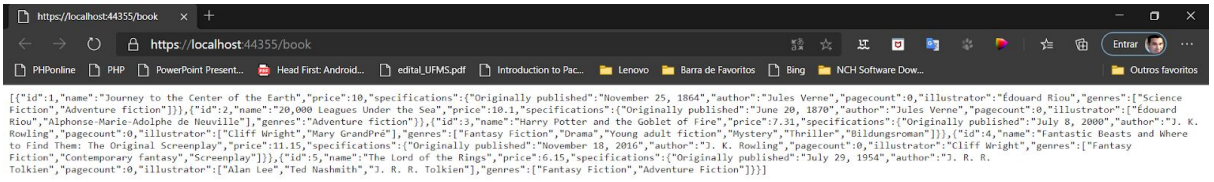
1 - baixe o repositório e abra o projeto no Visual Studio através do arquivo BookApi.sln



2 - Com o projeto aberto execute ele no Visual Studio



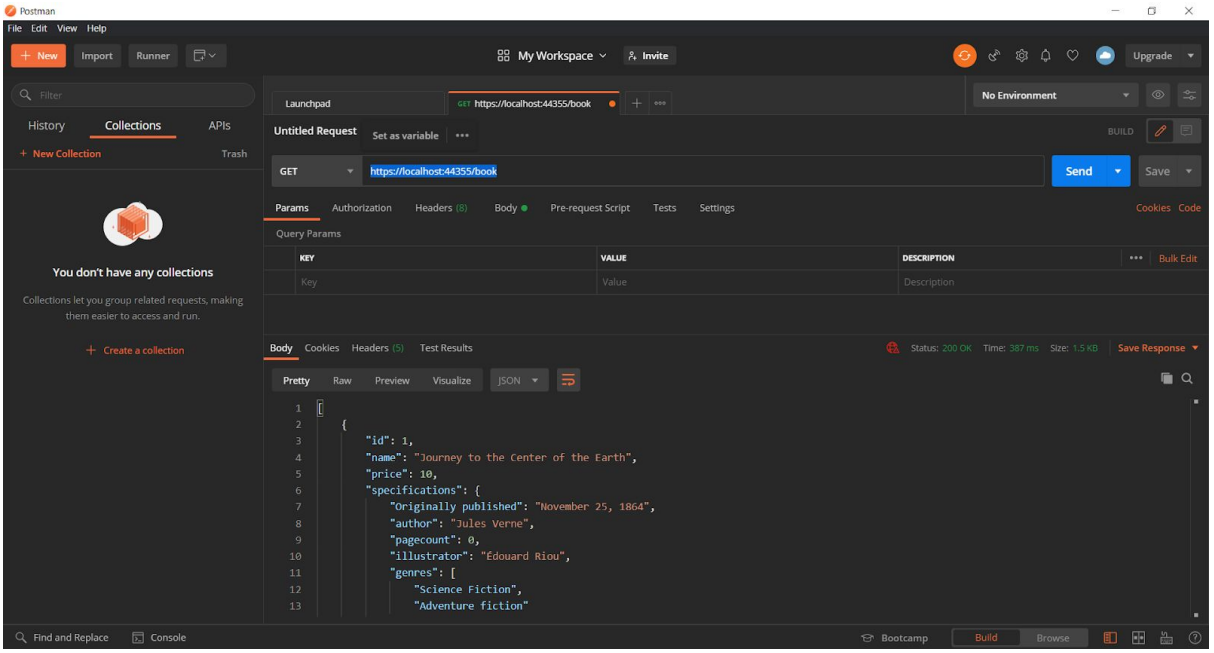
3 - Abrirá uma página do navegador e ela irá solicitar o método Get() que irá retornar todos os livros.



Obs.: O arquivo book.json não foi alterado porém foi movido para dentro do projeto.

Para fins de testes da api as demais funcionalidades, inclusive essa será apresentada no Postman, porém também poderão ser executados no browser, basta apenas digitar a URL que será apresentada

url: https://localhost:44355/book



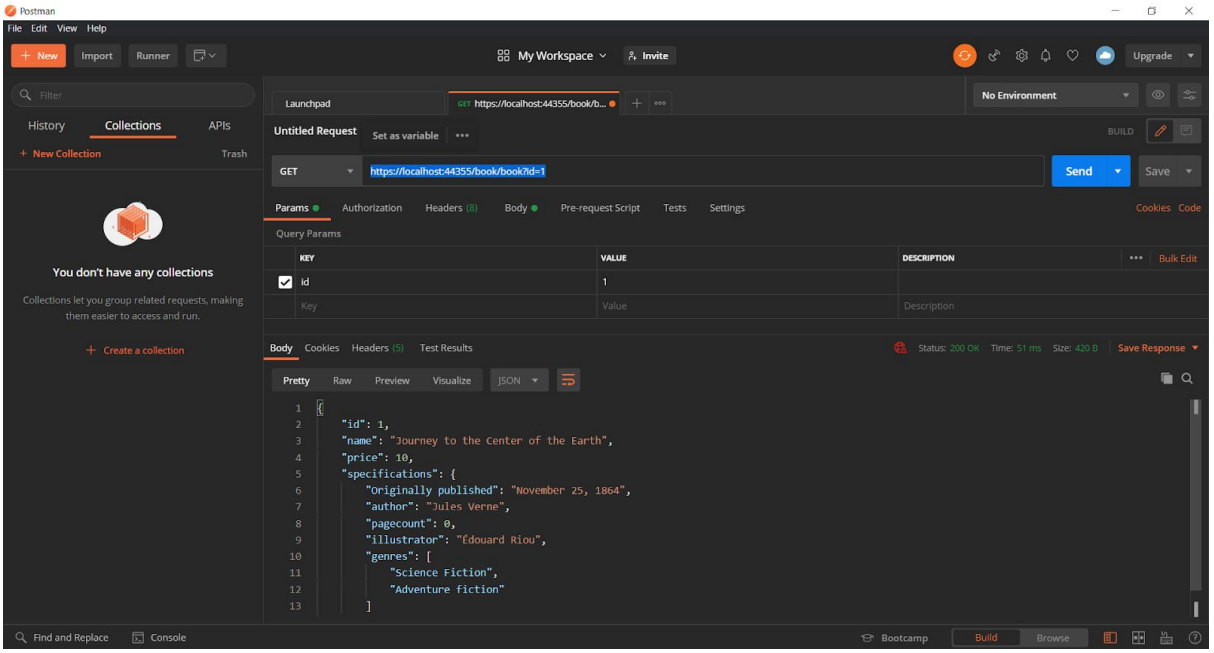
.

implementação dessa funcionalidade

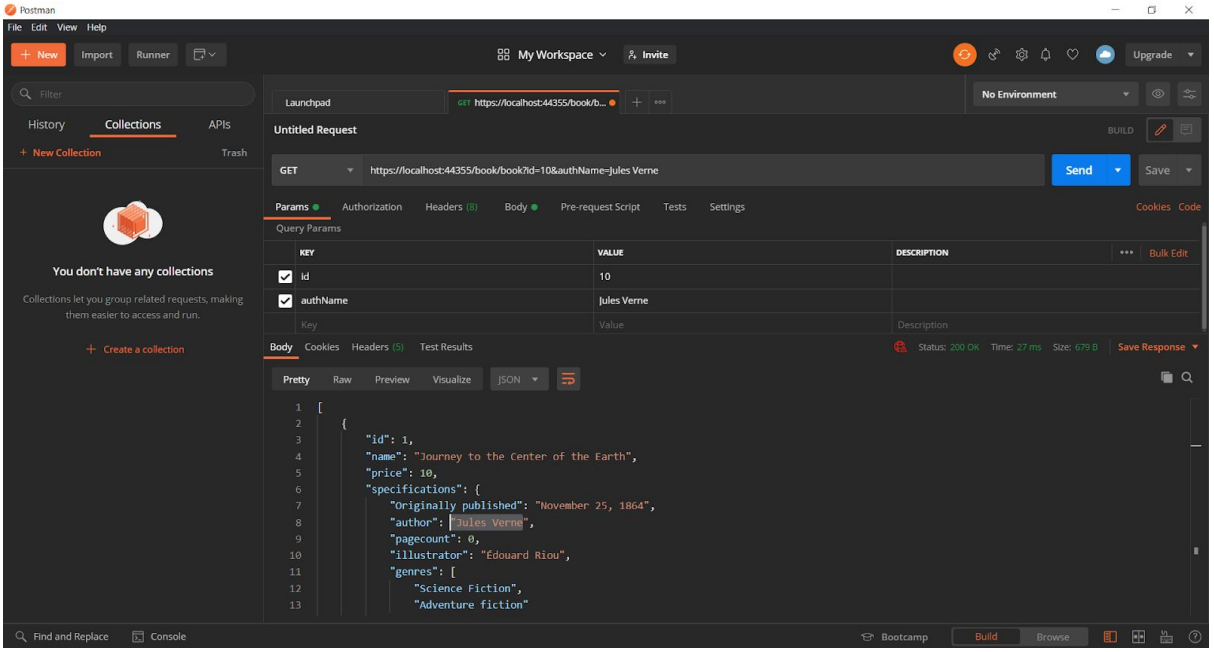
```
        /**
         * Busca todos os livros
         *
            * sort - um parâmetro que indica se a lista de livros deverá
ser ordenada ou não
         */
        [HttpGet]
        public IEnumerable<Book> Get(bool sort)
        {
            List<Book> list = new List<Book>();

            try {
                list = this.ReadBooks();
            } catch (Exception e) {
                return null;
            }

            return sort ? sortList(list) : list;
```

```
}
```

4 - Buscar livro por id
url: https://localhost:44355/book/book?id=1

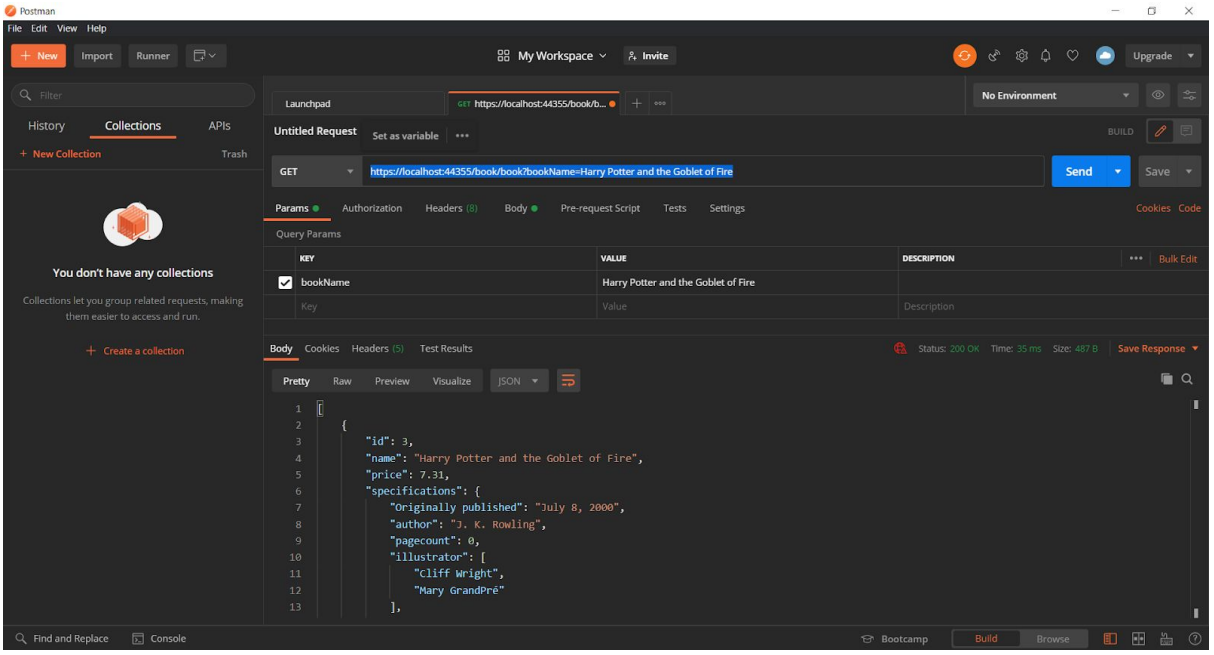


5 - Buscando o livro pelo nome do autor
url : https://localhost:44355/book/book?id=10&authName=Jules Verne

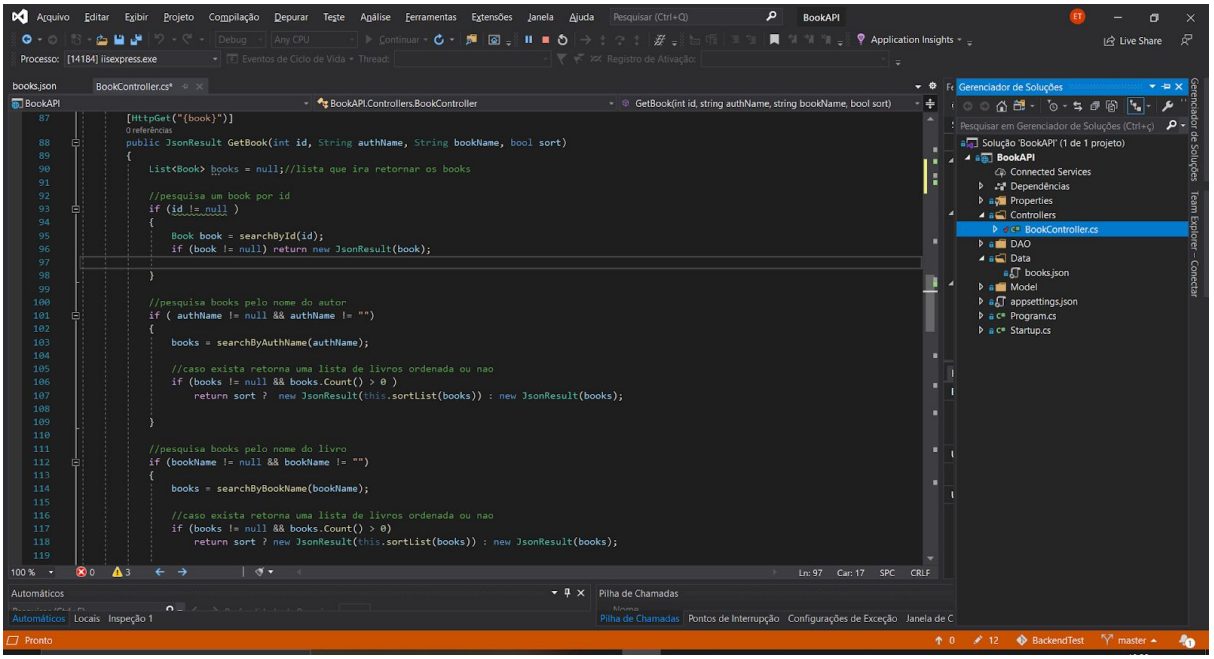


6 - Buscando livro pelo nome do livro

url: https://localhost:44355/book/book?bookName=Harry Potter and the Goblet of Fire

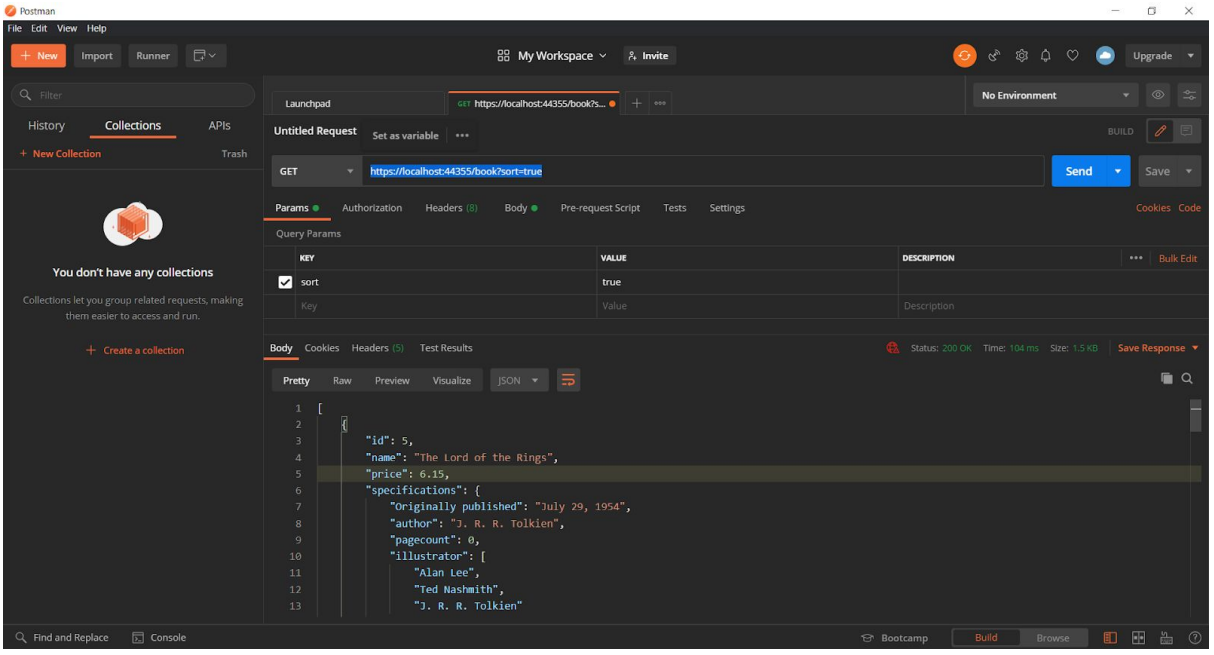


A seguir será exibido trecho do código que implementa essas funcionalidades.



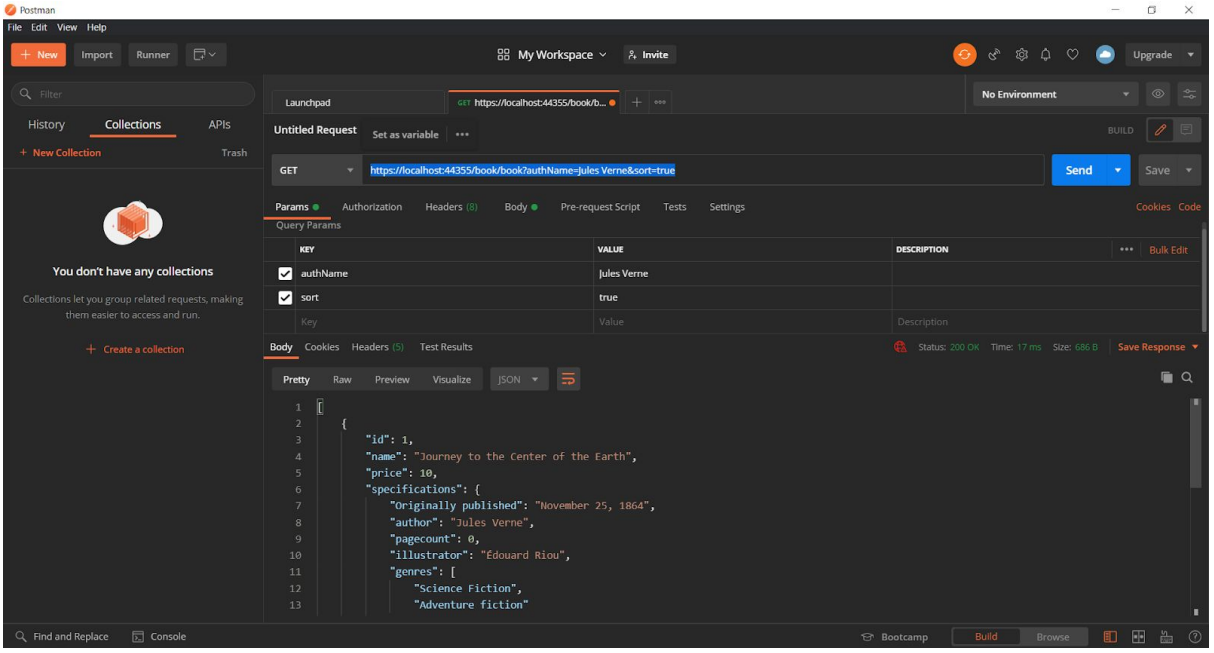
7 - Exibindo todos livros ordenados pelo preço

url: https://localhost:44355/book?sort=true



8 - Exibindo todos livros de um autor específico ordenados pelo preço

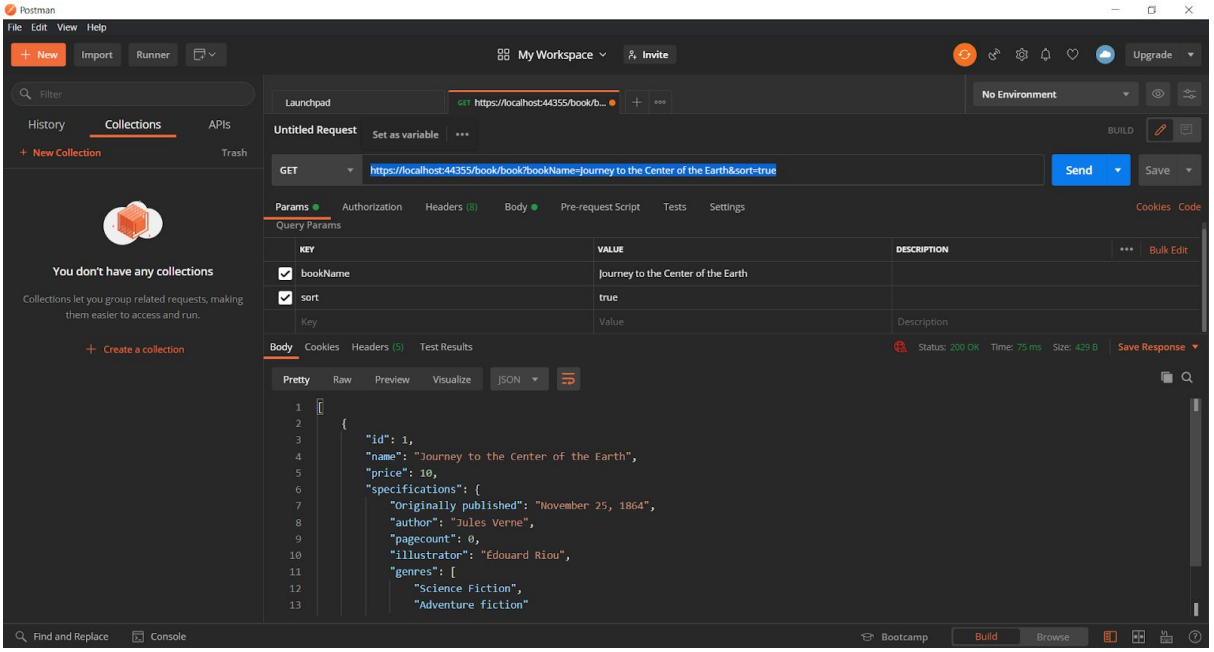
url: https://localhost:44355/book/book?authName=Jules Verne&sort=true



9 - Exibindo todos livros com o mesmo nome ordenados pelo preço

obs. no arquivo book.json não existia dados para este teste nem no documento de requisitos estava explícito se era ou não necessário a implementação disso, porém mesmo assim implementei visando a escalabilidade do sistema.

url: https://localhost:44355/book/book?bookName=Journey to the Center of the Earth&sort=true

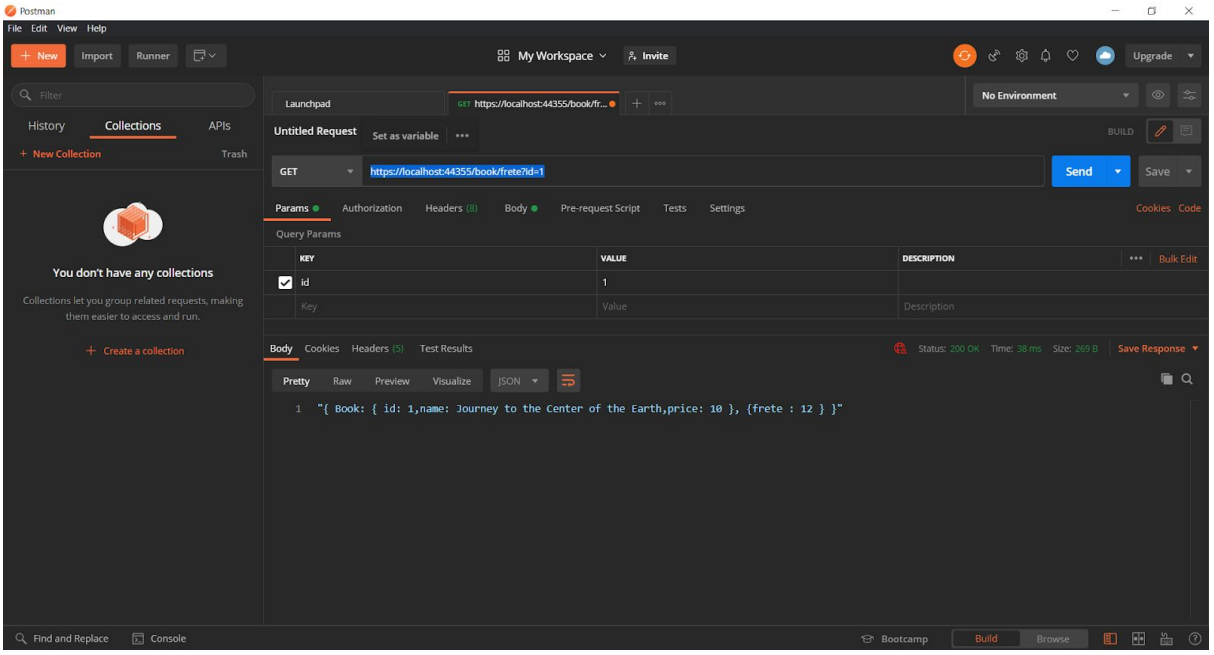


trecho de código com essa implementação

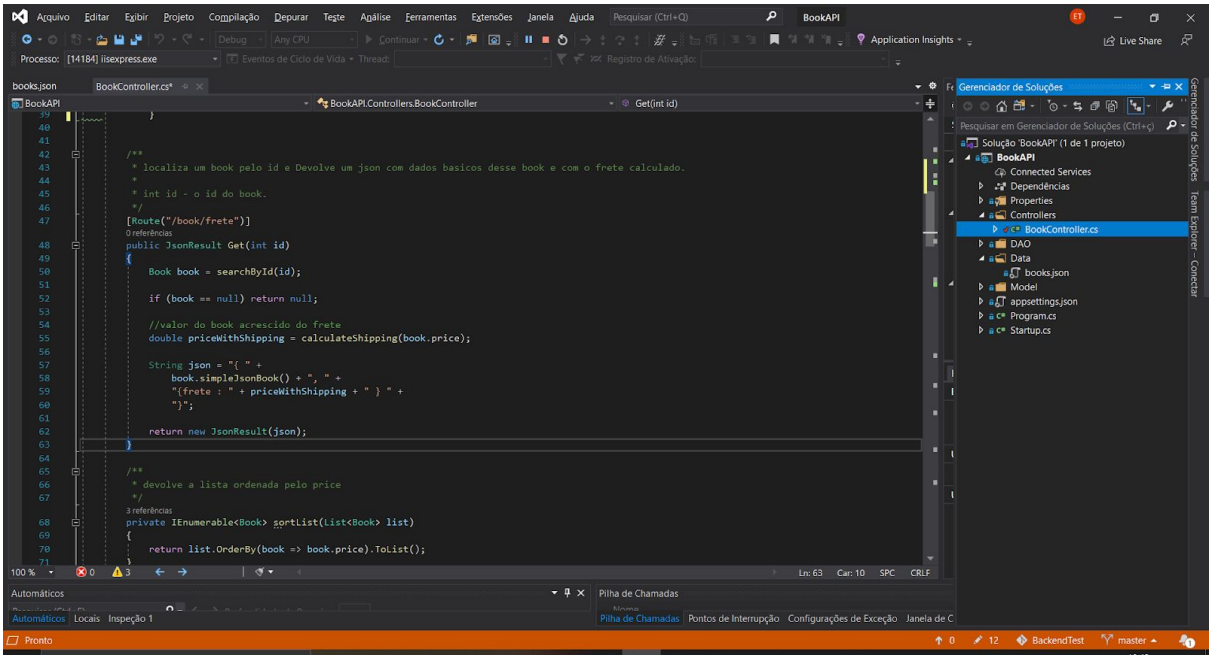
```
/**
```

```
         * devolve a lista ordenada pelo price
         */
        private IEnumerable<Book> sortList(List<Book> list)
        {
            return list.OrderBy(book => book.price).ToList();
        }
```

10 - método que calcula o valor do frete.
url: https://localhost:44355/book/frete?id=1



trecho do código


Arquivo Editar Exibir Projeto Compilação Depurar Teste Análise Ferramentas Extensões Janela Ajuda
BookAPI
Application Insights
Live Share
books.json
BookController.cs*
BookAPI.Controllers.BookController
Get(int id)
/**
* localiza um book pelo id e Devolve um json com dados basicos desse book e com o frete calculado.
*
* int id - o id do book.
*/
[Route("/book/frete")]
0 referências
public JsonResult Get(int id)
{
Book book = searchById(id);
if (book == null) return null;
//valor do book acrescido do frete
double priceWithShipping = calculateShipping(book.price);
String json = "{ " +
book.simpleJsonBook() + ", " +
"{frete : " + priceWithShipping + " } " +
"}";
return new JsonResult(json);
}
/**
* devolve a lista ordenada pelo price
*/
3 referências
private IEnumerable<Book> sortList(List<Book> list)
{
return list.OrderBy(book => book.price).ToList();
}
Gerenciador de Soluções
Solução 'BookAPI' (1 de 1 projeto)
BookAPI
Connected Services
Dependências
Properties
Controllers
BookController.cs
DAO
Data
books.json
Model
appsettings.json
Program.cs
Startup.cs
Automáticos
Pilha de Chamadas
Pronto
BackendTest
master