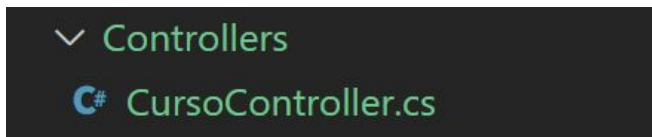


Objetivo: Usando o mesmo projeto do Exercício 3, vamos criar e executar a api de Curso.

- 1) Criar um arquivo CursoController.cs na pasta Controller.



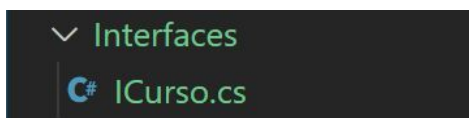
E adicione o código abaixo:

```
namespace shesharpapi.Controllers
{
    [Route("[controller]")]
    [ApiController]
    public class CursoController : Controller
    {
        public ICurso Cursos { get; set; }

        public CursoController(ICurso cur)
        {
            Cursos = cur;
        }

        [HttpGet]
        public List<CursoModel> GetTodosCursos()
        {
            return Cursos.GetTodosCursos();
        }
    }
}
```

- 2) Criar uma pasta Interfaces e depois um arquivo ICurso.cs



E adicione o código abaixo:

```
namespace shesharpapi.Interfaces
{
```

```

public interface ICurso
{
    List<CursoModel> GetTodosCursos();
}
}

```

- 3) Criar uma pasta Repository e depois um arquivo CursoRepository.cs



E adicione o código abaixo:

```

using System.Collections.Generic;
using System.Linq;
using shesharpapi.Interfaces;
using shesharpapi.Models;

namespace shesharpapi.Repository
{
    public class CursoRepository : ICurso
    {
        private readonly BDContext _context;
        public CursoRepository(BDContext context)
        {
            _context = context;
        }
        public List<CursoModel> GetTodosCursos()
        {
            return _context.cursos.ToList();
        }
    }
}

```

- 4) No arquivo Startup.cs adicione a linha:

```

services.AddScoped<ICurso, CursoRepository>();

```

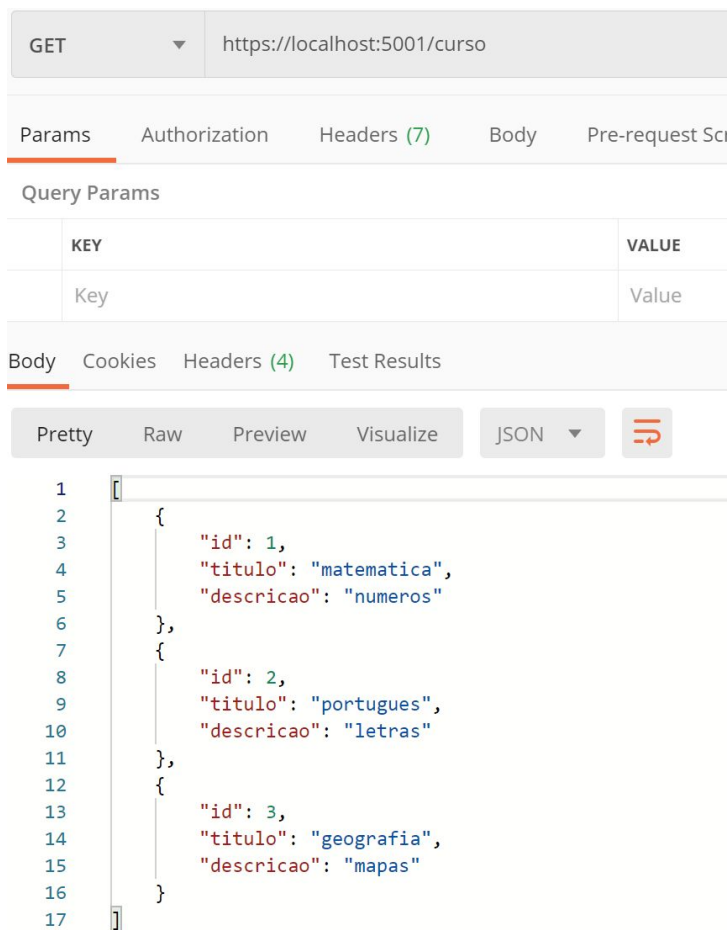
```

public void ConfigureServices(IServiceCollection services)
{
    var connection = @"Data Source=C:\SheSharp\bd\bdmentoria.db";
    services.AddDbContext<BDContext>(options => options.UseSqlite(connection));

    services.AddControllers();
    services.AddScoped<ICurso, CursoRepository>();
    services.AddCors();
}

```

- 5) No terminal e execute: dotnet run
- 6) Va no Postman e teste a API: <https://localhost:5001/curso>



The screenshot shows a Postman interface for a GET request to `https://localhost:5001/curso`. The 'Query Params' tab is active, showing a table with 'KEY' and 'VALUE' columns. Below the table, the 'Body' tab is selected, displaying a JSON array of three course objects. The JSON is formatted in 'Pretty' view.

KEY	VALUE
Key	Value

```

1  [
2    {
3      "id": 1,
4      "titulo": "matematica",
5      "descricao": "numeros"
6    },
7    {
8      "id": 2,
9      "titulo": "portugues",
10     "descricao": "letras"
11   },
12   {
13     "id": 3,
14     "titulo": "geografia",
15     "descricao": "mapas"
16   }
17 ]

```

- 7) Se funcionar, acrescente o arquivo .gitignore e acrescente o código abaixo:

```

*.swp
*.*~

```

```
project.lock.json
.DS_Store
*.pyc
nupkg/

# Visual Studio Code
.vscode

# Rider
.idea

# User-specific files
*.suo
*.user
*.userosscache
*.sln.docstates

# Build results
[Dd]ebug/
[Dd]ebugPublic/
[Rr]elease/
[Rr]eleases/
x64/
x86/
build/
bld/
[Bb]in/
[Oo]bj/
[Oo]ut/
msbuild.log
msbuild.err
msbuild.wrn

# Visual Studio 2015
.vs/
```

- 8) Por fim, não esqueça de fazer o commit e o push para o Github