

Programação Visual

Trabalho de Laboratório nº 7

Objetivo	MVC – Introdução. Aplicações básicas com WebAPI .
Programa	Pretende-se continuar a implementação da aplicação ESTeSoccer com o desenvolvimento de uma API que disponibilize as funcionalidades da aplicação (visualizar ligas, jogadores, equipas...) através de serviços REST , para poderem ser utilizados por outras aplicações.
Regras	Implementar o código necessário e testar no fim de cada nível. Use as convenções de codificação adotadas para a linguagem C# e para o modelo MVC.
Descrição	
Nível 1	<ul style="list-style-type: none">• Adicione um controlador designado LeaguesApiController ao projeto ESTeSoccer fornecido com este enunciado. Para isso, faça Add > Controller e, dentro da janela, selecione o modelo API Controller with actions, using Entity Framework.• Antes de testar será necessário adicionar as migrações e atualizar a base de dados com os comandos Nuget para esse fim.• Compile agora a aplicação e execute-a (CTRL+F5)<ul style="list-style-type: none">• Verifique que consegue obter a lista de ligas em formato json, usando os exemplos abaixo. Note que vai necessitar de modificar o valor do porto nestes exemplos: http://localhost:58279/api/LeaguesApi http://localhost:58279/api/LeaguesApi/2 <p>Nota: verifique o valor do porto no ficheiro launchSettings.json na pasta das propriedades ou no browser quando arranca com a aplicação.</p> <ul style="list-style-type: none">• Para verificar os restantes métodos de CRUD vai-se utilizar o Swagger. Sendo assim comece por instalar o package necessário através do seguinte comando: install-package Swashbuckle.AspNetCore• Agora no ficheiro Program.cs instale o serviço do Swagger: <pre>builder.Services.AddEndpointsApiExplorer(); builder.Services.AddSwaggerGen();</pre>• No mesmo ficheiro depois de <code>var app = builder.Build();</code> coloque o seguinte código: <pre>App.UseSwaggerUI(); App.UseSwagger();</pre>• Usando o swagger em http://localhost:58279/swagger/index.html, testes os restantes métodos da API.

Programação Visual

Trabalho de Laboratório nº 7

Nível 2

- Vamos agora desenvolver uma aplicação cliente para a **WebApi** criada no nível anterior. Aplicações clientes típicas podem ser aplicações Web, Apps de Smartphones Android/iPhone, Windows Desktop, etc. Para efeitos ilustrativos, é suficiente utilizar um menu de opções feito numa janela de consola, que sirva para testar a interação com a aplicação **ESTeSoccer**.
- Sendo assim, **adicione à solução** um novo projeto do tipo **Console App (.Net Core)**, com o nome **ESTeSoccerClient**. Para facilitar o desenvolvimento deste novo projeto, defina-o como principal (**Set as Startup Project**).
- Na janela do projeto, adicione a referência ao projeto **ESTeSoccer** (add project reference).
- **Selecione o projeto ESTeSoccerClient** na consola do **NuGet** e instale o seguinte pacote:

Install-Package Microsoft.AspNet.WebApi.Client

- Adicione ao projeto **ESTeSoccerClient**, a classe **ESTeSoccerClientUtils** fornecida com este enunciado.
- Abra agora o ficheiro fornecido com o nome **Cliente_WebApi_ilustrativo.txt** e copie o seu conteúdo para dentro da classe **Program** do projeto **ESTeSoccerClient**.
- No novo conteúdo da classe **Program**, Corrija o URL usado no seguinte código:

```
client.BaseAddress = new Uri("http://localhost: 58279/");
```

Nota: O url correto está dentro do ficheiro **launchSettings.json** (entrada **applicationUrl**) do projeto **ESTeSoccer**.

- Compile e verifique a ausência de erros.

Nível 3

- Neste passo, vai necessitar de ter a funcionar simultaneamente os projetos **ESTeSoccer** e **ESTeSoccerClient**. No **Solution Explorer**, selecione o projeto **ESTeSoccer** e execute a opção "**Debug > Start New Instance**".
- De forma similar, execute o projeto **ESTeSoccerClient**.
- Experimente a opção 1 do menu fornecida já implementada. Constate o seu funcionamento correto.
- Implemente agora o método **GetLeaguesAsync** de forma a que possa visualizar uma liga fornecendo o seu Id.

Nível 4

- Implemente agora os métodos **CreateLeagueAsync**, **UpdateLeagueAsync** e **DeleteLeagueAsync**.
- Chame cada um dos métodos implementados a partir da opção do menu existente no método **Main**. Constate o bom funcionamento dos métodos referidos.

Nível 5

- Acrescente agora uma opção de menu "[6] Consultar clubes de uma Liga" que a partir do Id da liga liste os clubes dessa liga. A Api deve retornar apenas o nome e a sigla do clube. Deve mostrar o nome da liga e listar os seus clubes mostrando o nome do clube seguida da sigla entre parênteses.

Programação Visual

Trabalho de Laboratório nº 7

Desafio

- Transforme o projeto inicial Asp.Net Core MVC para que passe a usar a API fornecida em vez de fazer o acesso direto à base de dados. Neste caso, apenas para as operações com as ligas.

Notas

Para os identificadores siga as convenções adotadas pelo C#, nomeadamente:

- A notação camelCase para o nome das variáveis locais e identificadores privados.
- A notação PascalCase para os nomes públicos dos métodos e classes
- Não utilize o símbolo de qualquer forma '_' nos identificadores
- Não use abreviaturas