



.NET 6: criando uma web API

94%


Buscar neste curso


Pesquisar 


AULA ATUAL


06. Documentando com Swagger 


ATIVIDADES


 01 Projeto da aula anterior


 02 Documentando a API 07min

 03 Configurando o Swagger


 04 Faça como eu fiz: documentando com Swagger


 05 Projeto final do curso


 06 O que aprendemos?

 07 Conclusão 03min

OUTROS LINKS

 Discord Alura

 Fórum do curso

 Voltar para Dashboard

MODO NOTURNO

Tiago Neves
28.8k xp

a

Chegou a hora de configurar o Swagge, uma aplicação open source que auxilia pessoas desenvolvedoras a criar, documentar e consumir APIs REST. Você colocou isso em prática? Vamos colocar a mão na massa e verifique se ficou com alguma dúvida. Se sim, você pode clicar na “Opinião do instrutor” e conferir passo a passo como isso foi feito.

Opinião do instrutor

De início, precisamos colocar nossas informações acerca do método `AdicionaFilme()` em nossa classe `FilmeController` como comentário seguindo o padrão XML:

```
/// <summary>
/// Adiciona um filme ao banco de dados
/// </summary>
/// <param name="filmeDto">Objeto com os campos necessários
/// <returns>ActionResult</returns>
/// <response code="201">Caso inserção seja feita com sucesso
[HttpPost]
[ProducesResponseType(StatusCodes.Status201Created)]
public IActionResult AdicionaFilme(
    [FromBody] CreateFilmeDto filmeDto)
{
    //código omitido
}
```

COPIAR CÓDIGO

Em seguida, precisamos alterar a nossa chamada do método `AddSwagger()` em nossa classe `Program.cs` :

```
builder.Services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new OpenApiInfo { Title = "FilmesAPI" });
    var xmlFile = $"{Assembly.GetExecutingAssembly().GetNameSpace()}.xml";
    var xmlPath = Path.Combine(AppContext.BaseDirectory, xmlFile);
    c.IncludeXmlComments(xmlPath);
});
```

COPIAR CÓDIGO

Por fim, altere seu `<PropertyGroup>` no arquivo `FilmesApi.csproj`:

```
<PropertyGroup>
  <TargetFramework>net6.0</TargetFramework>
  <Nullable>enable</Nullable>
  <ImplicitUsings>enable</ImplicitUsings>
  <GenerateDocumentationFile>true</GenerateDocumentationFile>
</PropertyGroup>
```

COPIAR CÓDIGO

Com isso, nosso método `AdicionaFilme()` estará devidamente documentado, informando os parâmetros recebidos, tipo de resposta e padrões implementados. Como desafio, faça sua própria documentação para os métodos de leitura, atualização e escrita!