CRIANDO API WEB .NET 6.0

**Exercício – Criar um projeto de API Web**

Concluído100 XP

* 8 minutos

Este módulo usa o SDK do .NET 6.0. Verifique se tem o .NET 6.0 instalado, executando o seguinte comando em seu terminal preferido:

CLI do .NETCopiar

dotnet --list-sdks

Saída semelhante à seguinte exibida:

ConsoleCopiar

3.1.100 [C:\program files\dotnet\sdk]

5.0.100 [C:\program files\dotnet\sdk]

6.0.100 [C:\program files\dotnet\sdk]

Verifique se uma versão que começa com 6 está listada. Se nenhum estiver listado ou o comando não for encontrado, [instale o SDK do .NET 6.0 mais recente](https://dotnet.microsoft.com/download).

**Criar e explorar um projeto de API Web**

Para configurar o projeto do .NET a fim de trabalhar com a API Web, usaremos o Visual Studio Code. O Visual Studio Code inclui um terminal integrado, o que facilita a criação de projetos. Se você não quiser usar um editor de código, poderá executar os comandos deste módulo em um terminal.

1. No Visual Studio Code, escolha **Arquivo**>**Abrir Pasta**.
2. Crie uma pasta chamada **ContosoPizza** no local de sua escolha e clique em **Selecionar Pasta**.
3. No Visual Studio Code, abra o terminal integrado selecionando **Exibir**>**Terminal** no menu principal.
4. Na janela do terminal, copie e cole o seguinte comando:

CLI do .NETCopiar

dotnet new webapi -f net6.0

Esse comando cria os arquivos de um projeto básico de API Web que usa controladores, juntamente com um arquivo de projeto em C# chamado *ContosoPizza.csproj* que retornará uma lista de previsões meteorológicas. Caso obtenha um erro, verifique se tem instalado o [SDK do .NET 6](https://dotnet.microsoft.com/download).

**Importante**

Os projetos de API Web são protegidos com https por padrão. Se você tiver problemas, [**configure o certificado de desenvolvimento HTTPS do ASP.NET Core**](https://docs.microsoft.com/pt-BR/aspnet/core/security/enforcing-ssl#trust-the-aspnet-core-https-development-certificate-on-windows-and-macos).

Você poderá receber um prompt do Visual Studio Code para adicionar ativos para depurar o projeto. Selecione **Sim** na caixa de diálogo.

O comando usa um modelo de projeto ASP.NET Core, com o alias *webapi*, para gerar o scaffolding de um projeto de API Web baseado em C#. Um diretório *ContosoPizza* é criado. Esse diretório contém um projeto ASP.NET Core em execução no .NET. O nome do projeto corresponde ao nome do diretório *ContosoPizza*.

Agora, você deve ter acesso aos estes arquivos:

BashCopiar

-| Controllers

-| obj

-| Properties

-| appsettings.Development.json

-| appsettings.json

-| ContosoPizza.csproj

-| Program.cs

-| WeatherForecast.cs

1. Examine os seguintes arquivos e diretórios:

| **Nome** | **Descrição** |
| --- | --- |
| *Controllers/* | Contém classes com métodos públicos expostos como pontos de extremidade HTTP |
| *Program.cs* | Configura serviços e o pipeline de solicitação HTTP do aplicativo e contém o ponto de entrada gerenciado do aplicativo |
| *ContosoPizza.csproj* | Contém metadados de configuração para o projeto |

**Compilar e testar a API Web**

1. Execute o seguinte comando da CLI do .NET Core no shell de comando:

CLI do .NETCopiar

dotnet run

O comando anterior:

* + Localiza o arquivo de projeto no diretório atual.
  + Recupera e instala as dependências de projeto necessárias para este projeto.
  + Compila o código do projeto.
  + Hospeda a API Web com o servidor Web Kestrel do ASP.NET Core em um ponto de extremidade HTTP e HTTPS.

Quando o projeto for criado, uma porta de 5000 a 5300 será selecionada para HTTP e de 7000 a 7300 para HTTPS. As portas que você usou durante o desenvolvimento podem ser facilmente alteradas com a edição do arquivo *launchSettings.json* do projeto. Esse módulo usa a URL segura localhost que começa com https .

Uma variação da seguinte saída aparece para indicar que seu aplicativo está em execução:

ConsoleCopiar

Building...

info: Microsoft.Hosting.Lifetime[14]

Now listening on: https://localhost:7294

info: Microsoft.Hosting.Lifetime[14]

Now listening on: http://localhost:5118

info: Microsoft.Hosting.Lifetime[0]

Application started. Press Ctrl+C to shut down.

info: Microsoft.Hosting.Lifetime[0]

Hosting environment: Development

Se estiver executando esse aplicativo em seu próprio computador, você poderá direcionar um navegador para o link HTTPS exibido na saída (no caso anterior, https://localhost:7294) para exibir a página resultante. Lembre-se dessa porta, pois você a usará em todo o módulo em que {PORT} for usado.

**Importante**

Verifique a saída do terminal se você encontrar algum comportamento inesperado. Se o build falhar ou se ocorrerem outros erros, as informações do arquivo de log ajudarão a solucionar os problemas. Ao fazer alterações no código, você precisará interromper a API Web selecionando Ctrl+C no teclado e executando o comando dotnet run novamente.

1. Abra um navegador da Web e acesse:

BashCopiar

https://localhost:{PORT}/weatherforecast

A seguinte saída representa um trecho do JSON retornado:

JSONCopiar

[

{

"date": "2021-11-09T20:36:01.4678814+00:00",

"temperatureC": 33,

"temperatureF": 91,

"summary": "Scorching"

},

{

"date": "2021-11-09T20:36:01.4682337+00:00",

"temperatureC": -8,

"temperatureF": 18,

"summary": "Cool"

},

// ...

]

1. Abra um novo terminal integrado do Visual Studio Code selecionando **Terminal**>**Novo Terminal** no menu principal. Em seguida, execute o seguinte comando:

CLI do .NETCopiar

dotnet tool install -g Microsoft.dotnet-httprepl

O comando anterior instala a ferramenta de linha de comando REPL HTTP do .NET que será usada para fazer solicitações HTTP para a API Web.

1. Conecte-se à API Web executando o seguinte comando:

CLI do .NETCopiar

httprepl https://localhost:{PORT}

Outra opção é executar o seguinte comando a qualquer momento enquanto HttpRepl estiver em execução:

CLI do .NETCopiar

(Disconnected)> connect https://localhost:{PORT}

1. Explore os pontos de extremidade disponíveis executando o seguinte comando:

CLI do .NETCopiar

ls

O comando anterior detecta todas as APIs disponíveis no ponto de extremidade conectado. Ele deve exibir o seguinte:

CLI do .NETCopiar

https://localhost:{PORT}/> ls

. []

WeatherForecast [GET]

1. Vá até o ponto de extremidade WeatherForecast executando o seguinte comando:

CLI do .NETCopiar

cd WeatherForecast

O comando anterior mostra uma saída das APIs disponíveis para o ponto de extremidade WeatherForecast:

CLI do .NETCopiar

https://localhost:{PORT}/> cd WeatherForecast

/WeatherForecast [GET]

1. Faça uma solicitação GET no HttpRepl usando o seguinte comando:

CLI do .NETCopiar

get

O comando anterior faz uma solicitação GET semelhante a ir para o ponto de extremidade no navegador:

CLI do .NETCopiar

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Date: Fri, 02 Apr 2021 17:31:43 GMT

Server: Kestrel

Transfer-Encoding: chunked

[

{

"date": 4/3/2021 10:31:44 AM,

"temperatureC": 13,

"temperatureF": 55,

"summary": "Sweltering"

},

{

"date": 4/4/2021 10:31:44 AM,

"temperatureC": -13,

"temperatureF": 9,

"summary": "Warm"

},

// ..

]

1. Saia da sessão HttpRepl atual usando o seguinte comando:

CLI do .NETCopiar

exit

1. Retorne ao terminal dotnet na lista suspensa no Visual Studio Code. Desligue a API Web selecionando Ctrl+C no teclado.

Agora que você criou a API Web, você a modificará para atender às necessidades da API Web de pizza.