**Exercício – implementar operações CRUD**

Concluído100 XP

* 10 minutos

Vamos continuar estendendo nosso controlador de API Web para adicionar a capacidade de criar (POST), atualizar (PUT) e excluir (DELETE) pizzas de nosso estoque.

**Adicionar uma pizza**

Habilite uma pizza a ser adicionada por meio da API Web usando um método POST.

Substitua o comentário // POST action em *Controllers/PizzaController.cs* pelo seguinte código:

C#Copiar

[HttpPost]

public IActionResult Create(Pizza pizza)

{

PizzaService.Add(pizza);

return CreatedAtAction(nameof(Create), new { id = pizza.Id }, pizza);

}

A ação anterior:

* Responde apenas ao verbo HTTP POST, conforme indicado pelo atributo [HttpPost].
* Insere o objeto Pizza do corpo da solicitação no cache na memória.

**Observação**

Como o controlador está anotado com o atributo [ApiController], está implícito que o parâmetro Pizza será encontrado no corpo da solicitação.

O primeiro parâmetro na chamada de método CreatedAtAction representa um nome de ação. A palavra-chave nameof é usada para evitar hard-coding do nome da ação. CreatedAtAction usa o nome da ação para gerar um cabeçalho de resposta HTTP location com uma URL para a pizza recém-criada, conforme explicado na unidade anterior.

**Modificar uma pizza**

Habilite uma pizza a ser atualizada por meio da API Web usando um método PUT.

Substitua o comentário // PUT action em *Controllers/PizzaController.cs* pelo seguinte código:

C#Copiar

[HttpPut("{id}")]

public IActionResult Update(int id, Pizza pizza)

{

if (id != pizza.Id)

return BadRequest();

var existingPizza = PizzaService.Get(id);

if(existingPizza is null)

return NotFound();

PizzaService.Update(pizza);

return NoContent();

}

A ação anterior:

* Responde apenas ao verbo HTTP PUT, conforme indicado pelo atributo [HttpPut].
* Requer que o valor do parâmetro id seja incluído no segmento da URL após pizza/.
* Retorna IActionResult porque o tipo de retorno ActionResult não é conhecido até o runtime. Os métodos BadRequest, NotFound e NoContent retornam os tipos BadRequestResult, NotFoundResult e NoContentResult respectivamente.

**Observação**

Como o controlador está anotado com o atributo [ApiController], está implícito que o parâmetro Pizza será encontrado no corpo da solicitação.

**Remover uma pizza**

Habilite uma pizza a ser removida por meio da API Web usando um método DELETE.

Substitua o comentário // DELETE action em *Controllers/PizzaController.cs* pelo seguinte código:

C#Copiar

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

var pizza = PizzaService.Get(id);

if (pizza is null)

return NotFound();

PizzaService.Delete(id);

return NoContent();

}

A ação anterior:

* Responde apenas ao verbo HTTP DELETE, conforme indicado pelo atributo [HttpDelete].
* Requer que o valor do parâmetro id seja incluído no segmento da URL após pizza/.
* Retorna IActionResult porque o tipo de retorno ActionResult não é conhecido até o runtime. Os métodos NotFound e NoContent retornam os tipos NotFoundResult e NoContentResult, respectivamente.
* Consulta o cache na memória para uma pizza que corresponde ao parâmetro id fornecido.

**Compilar e executar a API Web concluída**

1. Compile e inicie a API Web executando o seguinte comando:

CLI do .NETCopiar

dotnet run

1. Abra novamente o terminal httprepl existente ou um novo terminal integrado do Visual Studio Code selecionando **Terminal**>**Novo Terminal** no menu principal.
2. Se você abriu um novo terminal, 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. Vá até o ponto de extremidade Pizza executando o seguinte comando:

CLI do .NETCopiar

cd Pizza

1. Execute o seguinte comando para ver as novas ações na API de Pizza:

CLI do .NETCopiar

ls

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

CLI do .NETCopiar

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

. [GET|POST]

.. []

{id} [GET|PUT|DELETE]

1. Faça uma solicitação POST para adicionar uma nova pizza a HttpRepl usando o seguinte comando:

CLI do .NETCopiar

post -c "{"name":"Hawaii", "isGlutenFree":false}"

O comando anterior retorna uma lista de todas as pizzas:

CLI do .NETCopiar

HTTP/1.1 201 Created

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

Date: Fri, 02 Apr 2021 23:23:09 GMT

Location: https://localhost:{PORT}/Pizza?id=3

Server: Kestrel

Transfer-Encoding: chunked

{

"id": 3,

"name": "Hawaii",

"isGlutenFree": false

}

1. Atualize a nova pizza Hawaii para uma pizza Hawaiian com uma solicitação PUT usando o seguinte comando:

CLI do .NETCopiar

put 3 -c "{"id": 3, "name":"Hawaiian", "isGlutenFree":false}"

O comando anterior retorna a seguinte saída que indica êxito:

CLI do .NETCopiar

HTTP/1.1 204 No Content

Date: Fri, 02 Apr 2021 23:23:55 GMT

Server: Kestrel

Para verificar se a pizza foi atualizada, execute novamente a ação GET usando o seguinte comando:

CLI do .NETCopiar

get 3

O comando anterior retorna a pizza recém-atualizada:

CLI do .NETCopiar

HTTP/1.1 200 OK

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

Date: Fri, 02 Apr 2021 23:27:37 GMT

Server: Kestrel

Transfer-Encoding: chunked

{

"id": 3,

"name": "Hawaiian",

"isGlutenFree": false

}

1. Nossa API também pode excluir a pizza recém-criada por meio da ação DELETE se você executar o seguinte comando:

CLI do .NETCopiar

delete 3

O comando anterior retorna um resultado 204 No Content para êxito:

CLI do .NETCopiar

HTTP/1.1 204 No Content

Date: Fri, 02 Apr 2021 23:30:04 GMT

Server: Kestrel

Para verificar se a pizza foi removida, execute novamente a ação GET usando o seguinte comando:

CLI do .NETCopiar

get

O comando anterior retorna as pizzas originais como resultados:

CLI do .NETCopiar

HTTP/1.1 200 OK

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

Date: Fri, 02 Apr 2021 23:31:15 GMT

Server: Kestrel

Transfer-Encoding: chunked

[

{

"id": 1,

"name": "Classic Italian",

"isGlutenFree": false

},

{

"id": 2,

"name": "Veggie",

"isGlutenFree": true

}

]

Você terminou de implementar e testar uma API Web recém-criada com ASP.NET Core.