**Exercício – Adicionar um armazenamento de dados**

Concluído100 XP

* 4 minutos

Antes de começar a implementar uma API Web para pizza, é preciso ter um armazenamento de dados em que você possa executar operações.

Você precisa de uma classe model para representar uma pizza no estoque. O modelo contém propriedades que representam as características de uma pizza. O modelo é usado para passar dados na API Web e persistir opções de pizza no armazenamento de dados.

Nesta unidade, o armazenamento de dados é um serviço de cache na memória local simples. Em um aplicativo do mundo real, você consideraria usar um banco de dados, como SQL Server, com o Entity Framework Core.

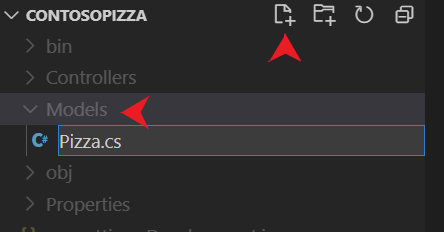
**Criar um modelo de pizza**

1. Execute o seguinte comando para criar uma pasta *Modelos*:

BashCopiar

mkdir Models

Selecione a pasta *Modelos* no Visual Studio Code e adicione um novo arquivo chamado *Pizza.cs*.



Agora, a raiz do projeto contém um diretório *Models* com um arquivo *Pizza.cs* vazio. O nome do diretório *Models* é uma convenção. O nome do diretório vem da arquitetura *model*-view-controller que a API Web usa.

1. Adicione o código a seguir a *Models/Pizza.cs* para definir uma pizza. Salve suas alterações.

C#Copiar

namespace ContosoPizza.Models;

public class Pizza

{

public int Id { get; set; }

public string? Name { get; set; }

public bool IsGlutenFree { get; set; }

}

**Dica**

Os atalhos de teclado a seguir são úteis no editor do Azure Cloud Shell.

| **Atalho de teclado** | **Comando** |
| --- | --- |
| CTRL + V (Windows) ⌘ + V (macOS) | Colar |
| CTRL + S (Windows) ⌘ + S (macOS) | Salvar |

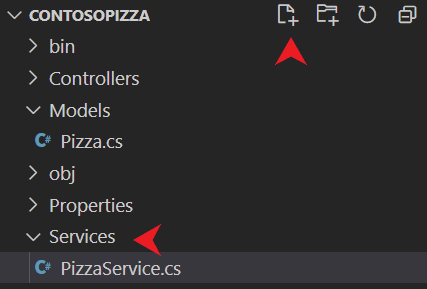
**Adicionar um serviço de dados**

1. Execute o seguinte comando para criar uma pasta *Serviços*:

BashCopiar

mkdir Services

Selecione a pasta no Visual Studio Code e adicione um novo arquivo chamado *PizzaService.cs*.



1. Adicione o código a seguir a *Services/PizzaService.cs* para criar um serviço de dados de pizza na memória. Salve suas alterações.

C#Copiar

using ContosoPizza.Models;

namespace ContosoPizza.Services;

public static class PizzaService

{

static List<Pizza> Pizzas { get; }

static int nextId = 3;

static PizzaService()

{

Pizzas = new List<Pizza>

{

new Pizza { Id = 1, Name = "Classic Italian", IsGlutenFree = false },

new Pizza { Id = 2, Name = "Veggie", IsGlutenFree = true }

};

}

public static List<Pizza> GetAll() => Pizzas;

public static Pizza? Get(int id) => Pizzas.FirstOrDefault(p => p.Id == id);

public static void Add(Pizza pizza)

{

pizza.Id = nextId++;

Pizzas.Add(pizza);

}

public static void Delete(int id)

{

var pizza = Get(id);

if(pizza is null)

return;

Pizzas.Remove(pizza);

}

public static void Update(Pizza pizza)

{

var index = Pizzas.FindIndex(p => p.Id == pizza.Id);

if(index == -1)

return;

Pizzas[index] = pizza;

}

}

Este serviço fornece um serviço simples de cache de dados na memória com duas pizzas por padrão. Nossa API Web usará esse serviço para fins de demonstração. Quando você interrompe e inicia a API Web, o cache de dados na memória é redefinido para as duas pizzas padrão do construtor do PizzaService.

**Compilar o projeto de API Web**

Execute o comando a seguir para criar o aplicativo:

CLI do .NETCopiar

dotnet build

O build tem êxito sem nenhum aviso. Se o build falhar, verifique a saída para obter informações de solução de problemas.

Na próxima unidade, você criará um controlador que usará o modelo Pizza e a classe PizzaService.