Testing MVC Applications

How to Run Unit Tests

1. Abrir UnitTestingExample.sln
2. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, in Solution Explorer, right-click \*\*Solution 'UnitTestingExample'\*\*, point to \*\*Add\*\*, and then click \*\*New Project\*\*.

3. In the \*\*Add New Project\*\* dialog box, in the navigation pane, expand \*\*Installed\*\*, expand \*\*Visual C#\*\*, and then click \*\*.NET Core\*\*.

4. In the \*\*Add New Project\*\* dialog box, in the result pane, click \*\*MSTest Test Project (.NET Core)\*\*.

5. In the \*\*Add New Project\*\* dialog box, in the \*\*Name\*\* box, type \*\*ProductsWebsite.Tests\*\*, and then click \*\*OK\*\*.

6. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, in Solution Explorer, under \*\*ProductsWebsite.Tests\*\*, right-click \*\*Dependencies\*\*, and then click \*\*Add Reference\*\*.

7. In the \*\*Reference Manager - ProductsWebsite.Tests\*\* dialog box, in the navigation pane, expand \*\*Projects\*\*, and then click \*\*Solution\*\*.

8. In the \*\*Reference Manager - ProductsWebsite.Tests\*\* dialog box, in the result pane, select the \*\*ProductsWebsite\*\* check box, and then click \*\*OK\*\*.

9. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, in Solution Explorer, under \*\*ProductsWebsite.Tests\*\*, right-click \*\*UnitTest1.cs\*\*, and then click \*\*Rename\*\*.

10. In the \*\*UnitTest1.cs\*\* box, type \*\*ProductControllerTest.cs\*\*, and then press Enter.

11. In the \*\*Microsoft Visual Studio\*\* dialog box, click \*\*Yes\*\*.

12. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, on the \*\*TOOLS\*\* menu, point to \*\*NuGet Package Manager\*\*

```

Install - Package Microsoft.AspNetCore.Mvc

14. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, in Solution Explorer, right-click \*\*ProductsWebsite.Tests\*\*, point to \*\*Add\*\*, and then click \*\*New Folder\*\*.

15. In the \*\*NewFolder\*\* box, type \*\*FakeRepositories\*\*, and then press Enter.

16. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, in Solution Explorer, right-click \*\*FakeRepositories\*\*, point to \*\*Add\*\*, and then click \*\*Class\*\*.

17. In the \*\*Add New Item - ProductsWebsite.Tests\*\* dialog box, in the \*\*Name\*\* box, type \*\*FakeProductRepository\*\*, and then click \*\*Add\*\*.

18. In the \*\*FakeProductRepository.cs\*\* code window, locate the following code:

```cs

using System.Text;

```

19. Ensure that the cursor is at the end of the located code, press Enter, and then type the following code:

```cs

```

20. In the \*\*FakeProductRepository.cs\*\* code window, select the following code:

```cs

class FakeProductRepository

```

21. Replace the selected code with the following code:

```cs

internal class FakeProductRepository : IProductRepository

```

22. In the \*\*FakeProductRepository.cs\*\* code window, locate the following code:

```cs

internal class FakeProductRepository : IProductRepository

{

```

23. Ensure that the cursor is at the end of the located code, press Enter, and then type the following code:

```cs

public IEnumerable<Product> GetProducts()

{

return new List<Product>()

{

new Product{ Id = 1, Name = "Product1's name", BasePrice = 1.1F, Description = "A description for product 1.", ImageName = "image-name-1" },

new Product{ Id = 2, Name = "Product2's name", BasePrice = 2.2F, Description = "A description for product 2.", ImageName = "image-name-2" },

new Product{ Id = 3, Name = "Product3's name", BasePrice = 3.3F, Description = "A description for product 3.", ImageName = "image-name-3" }

};

}

```

24. In the \*\*ProductControllerTest.cs\*\* code window, locate the following code:

```cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

```

25. Ensure that the cursor is at the end of the located code, press Enter, and then type the following code:

```cs

using System.Collections.Generic;

using Microsoft.AspNetCore.Mvc;

using ProductsWebsite.Controllers;

using ProductsWebsite.Models;

using ProductsWebsite.Repositories;

using ProductsWebsite.Tests.FakeRepositories;

```

26. In the \*\*ProductControllerTest.cs\*\* code window, select the following code:

```cs

public void TestMethod1()

{

}

```

27. Replace the selected code with the following code:

```cs

public void IndexModelShouldContainAllProducts()

{

// Arrange

IProductRepository fakeProductRepository = new FakeProductRepository();

ProductController productController = new ProductController(fakeProductRepository);

// Act

ViewResult viewResult = productController.Index() as ViewResult;

List<Product> products = viewResult.Model as List<Product>;

// Assert

Assert.AreEqual(products.Count, 3);

}

```

28. In the \*\*ProductControllerTest.cs\*\* code window, locate the following code:

```cs

Assert.AreEqual(products.Count, 3);

}

```

29. Ensure that the cursor is at the end of the located code, press Enter twice, and then type the following code:

```cs

[TestMethod]

public void GetProductModelShouldContainTheRightProduct()

{

// Arrange

var fakeProductRepository = new FakeProductRepository();

var productController = new ProductController(fakeProductRepository);

// Act

var viewResult = productController.GetProduct(2) as ViewResult;

Product product = viewResult.Model as Product;

// Assert

Assert.AreEqual(product.Id, 2);

}

```

30. In the \*\*UnitTestingExample – Microsoft Visual Studio\*\* window, on the \*\*FILE\*\* menu, click \*\*Save All\*\*.

31. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, on the \*\*TEST\*\* menu, point to \*\*Run\*\*, and then click \*\*All Tests\*\*.

>\*\*Note\*\*: The \*\*Test Explorer\*\* displays one failed test: \*\*GetProductModelShouldContainTheRightProduct\*\*, and one passed test: \*\*IndexModelShouldContainAllProducts\*\*.

32. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, in Solution Explorer, under \*\*ProductsWebsite\*\*, expand \*\*Controllers\*\*, and then click \*\*ProductController.cs\*\*.

33. In the \*\*ProductController.cs\*\* code window, select the following code:

```cs

var product = products.Where(p => p.Id != id).FirstOrDefault();

```

34. Replace the selected code with the following code:

```cs

var product = products.Where(p => p.Id == id).FirstOrDefault();

```

35. In the \*\*UnitTestingExample – Microsoft Visual Studio\*\* window, on the \*\*FILE\*\* menu, click \*\*Save All\*\*.

36. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, on the \*\*TEST\*\* menu, point to \*\*Run\*\*, and then click \*\*All Tests\*\*.

>\*\*Note\*\*: The \*\*Test Explorer\*\* displays two passed tests: \*\*GetProductModelShouldContainTheRightProduct\*\* and \*\*IndexModelShouldContainAllProducts\*\*.

37. In the \*\*UnitTestingExample - Microsoft Visual Studio\*\* window, on the \*\*FILE\*\* menu, click \*\*Exit\*\*.