

Övning Skapa MVC-applikation med Unit-test

Arbetsuppgift 1: Starta Visual Studio och skapa ett nytt projekt

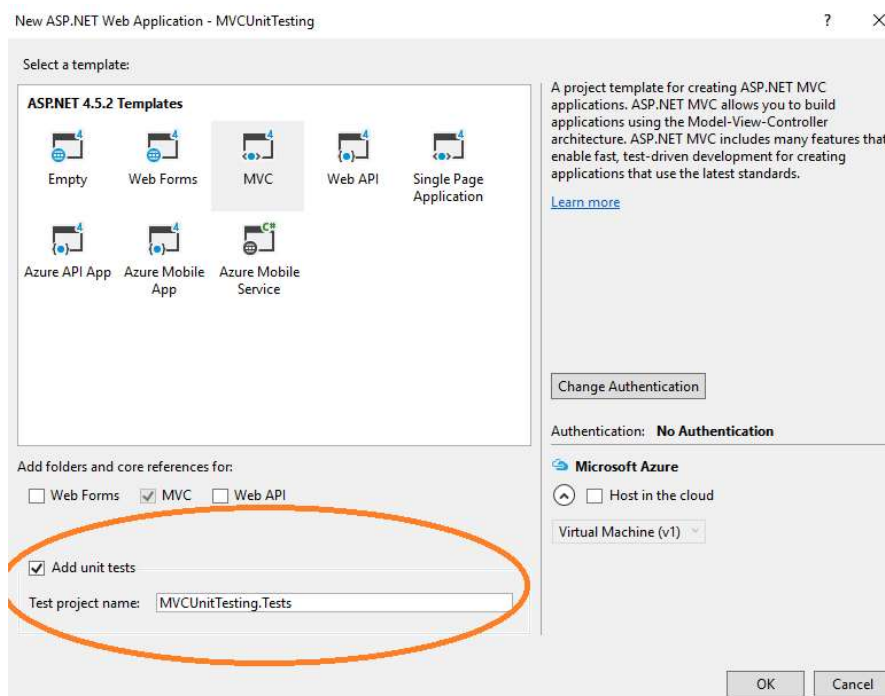
Steg 1: Starta Visual Studio.

Steg 2: Klicka på File – New – Project.

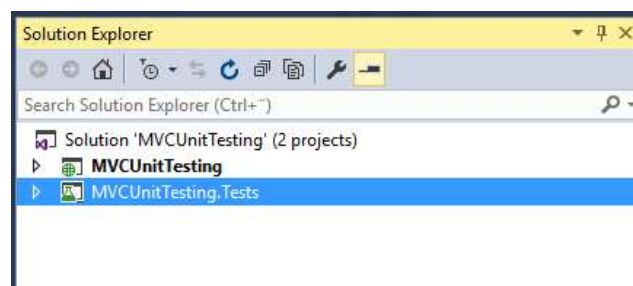
Steg 3: Expandera Templates – Visual C# - Web.

Steg 4: Välj ASP.NET Web Application.

Steg 5: I fältet för Name, skriv in MVCUnitTesting, klicka på OK.

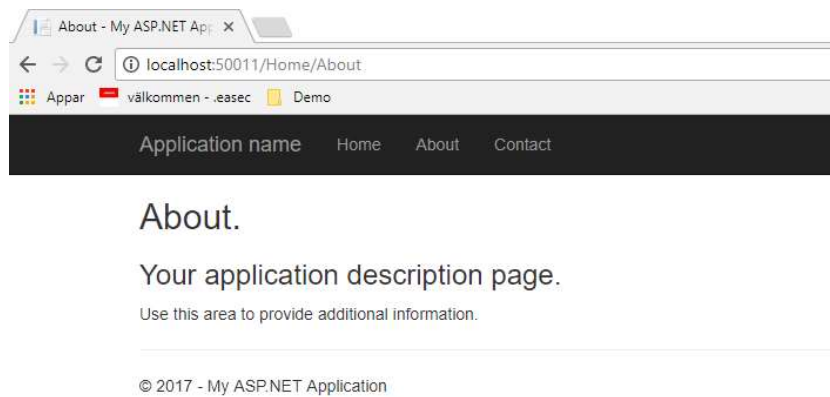


Steg 6: Välj MVC som mall och glöm inte att kryssa i rutan för Add unit tests, låt standardnamnet vara kvar (går att ändra om du vill). Klicka på Change Authentication, välj No Authentication, klicka på OK på två gånger.



Steg 7: I Solution Explorer visas två stycken projekt.

Steg 8: Klicka på Debug – Start Debugging.



Steg 9: I navigationsdelen finns tre stycken knappar, Home, About och Contact. Klicka på About.

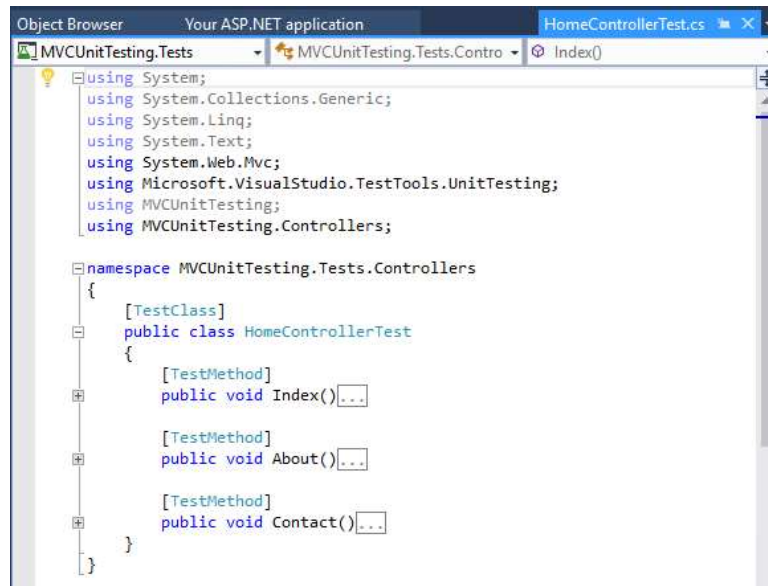
Steg 10: Avbryt Debugging.

Steg 11: Expandera projektet MVCUnitTesting, dubbelklicka på HomeController.cs.



Steg 12: Observera att HomeController.cs innehåller tre stycken sk action methods.

Steg 13: I Solution Explorer, expandera ditt testprojekt, dubbelklicka på HomeControllerTest.cs.



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Web.Mvc;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using MVCUnitTesting;
using MVCUnitTesting.Controllers;

namespace MVCUnitTesting.Tests.Controllers
{
    [TestClass]
    public class HomeControllerTest
    {
        [TestMethod]
        public void Index() {...}

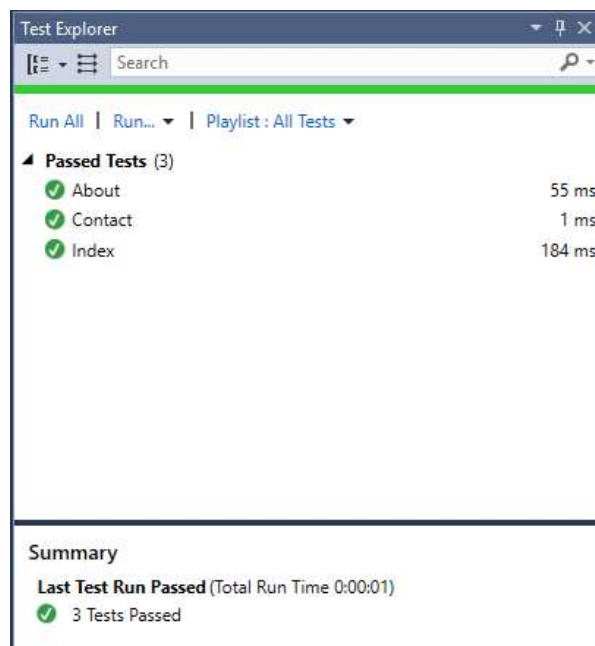
        [TestMethod]
        public void About() {...}

        [TestMethod]
        public void Contact() {...}
    }
}

```

Tre stycken testmetoder visas, en för var och en av action methods som vi såg tidigare. Dessa används för att kontrollera så att dessa fungerar.

Steg 14: Gå till Test menu, klicka på Run - All Tests.



Test Name	Duration
About	55 ms
Contact	1 ms
Index	184 ms

Summary
Last Test Run Passed (Total Run Time 0:00:01)
 3 Tests Passed

Arbetsuppgift 2: Lägg till ny action method Employee

Denna action method kommer att visa alla anställda.

Steg 1: Lägg till klass för Employee i katalogen Models.

Ange följande programkod:

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MVCUnitTesting.Models{
    public class Employee{
        public int ID { get; set; }
        public string Name { get; set; }
        public DateTime JoiningDate { get; set; }
        public int Age { get; set; }
    }
}
```

Steg 2: Högerklicka på katalogen Controller, klicka på Add – Controller.

Steg 3: Välj MVC 5 Controller – Empty, klicka på Add.

Steg 4: I rutan för namn, skriv in EmployeeController, klicka på Add.

Steg 5: Uppdatera programkoden med:

```
using MVCUnitTesting.Models;
using System;
using System.Collections.Generic;
using System.Linq;

using System.Web;
using System.Web.Mvc;

namespace MVCUnitTesting.Controllers {
```

```
public class EmployeeController : Controller{
    [NonAction]

    public List<Employee> GetEmployeeList(){
        return new List<Employee>{
            new Employee{
                ID = 1,
                Name = "Allan",
                JoiningDate =
DateTime.Parse(DateTime.Today.ToString()),
                Age = 23
            },

            new Employee{
                ID = 2,
                Name = "Carson",
                JoiningDate =
DateTime.Parse(DateTime.Today.ToString()),
                Age = 45
            },

            new Employee{
                ID = 3,
                Name = "Carson",
                JoiningDate =
DateTime.Parse(DateTime.Today.ToString()),
```

```
        Age = 37
    },

    new Employee{
        ID = 4,
        Name = "Laura",
        JoiningDate =
DateTime.Parse(DateTime.Today.ToString()),
        Age = 26
    },
};
}

// GET: Employee
public ActionResult Index(){
    return View();
}

public ActionResult Employees(){
    var employees = from e in GetEmployeeList()
        orderby e.ID
        select e;
    return View(employees);
}
}
```

```
}
```

Steg 6: Lägg till View, genom att högerklicka på händelse Employees, välj Add View.

Steg 7: Behåll standardnamnet, för Template välj List och Employee från Model class. Klicka bort markering för Reference script libraries, klicka på Add.

Steg 8: Lägg till länk för lista över anställda, klicka på _layout.cshtml, under Views/Shared, lägg till länk till anställda under länk för Contacts.

```
<li>@Html.ActionLink("Employees List", "Employees", "Employee")</li>
```

Arbetsuppgift 3: Lägga till unit-test

Steg 1: Expandera ditt testprojekt, dubbelklicka på HomeControllerTest.cs. Lägg till följande klass i ditt testprojekt:

```
[TestClass]

public class EmployeeControllerTest{

    [TestMethod]

    public void Employees(){

        // Arrange

        EmployeeController controller = new
EmployeeController();
```

```

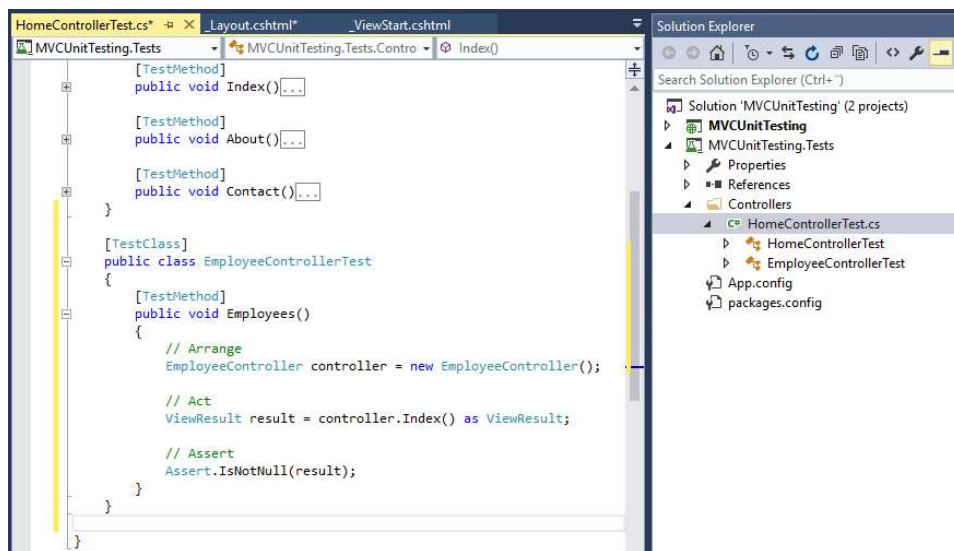
// Act

ActionResult result = controller.Index() as
ActionResult;

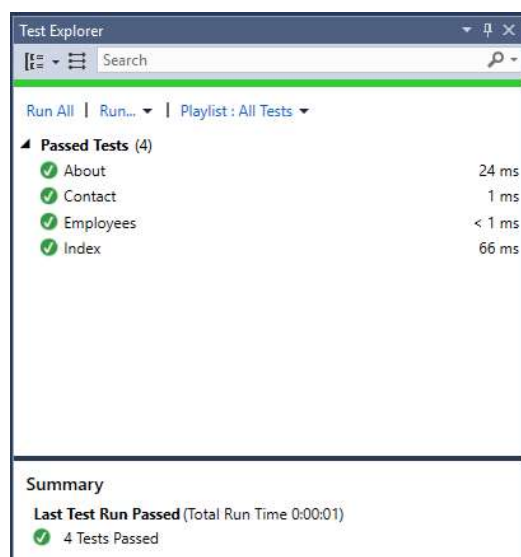
// Assert

Assert.IsNotNull(result);
}
}

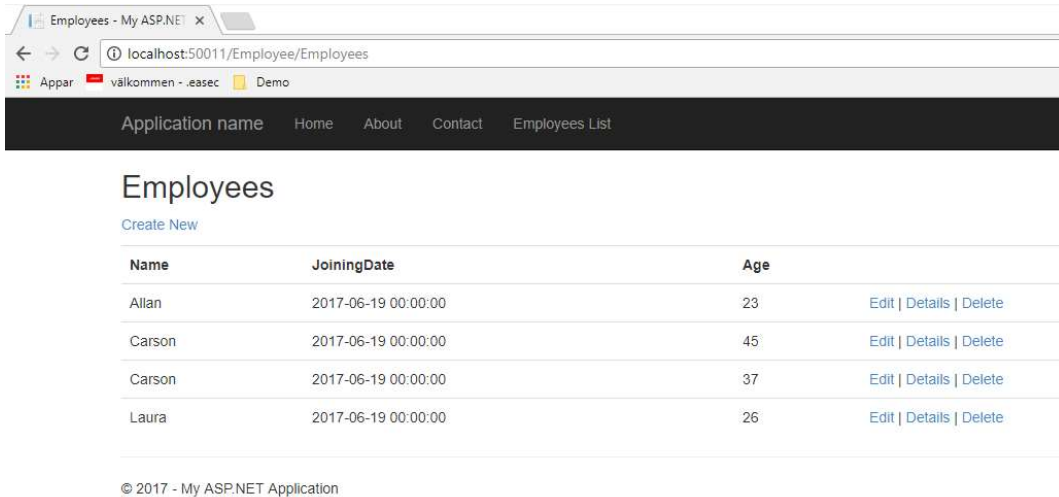
```



Steg 2: Under Test menu, klicka på Run – All tests, alla testen bör gå igenom.



Steg 3: Klicka på Debug – Start Debugging.



The screenshot shows a web browser window with the following details:

- Tab: Employees - My ASP.NET
- Address bar: localhost:50011/Employee/Employees
- Page title: Application name
- Navigation menu: Home, About, Contact, Employees List
- Section header: Employees
- Link: Create New
- Table with columns: Name, JoiningDate, Age, and a list of actions (Edit | Details | Delete).
- Table data:

Name	JoiningDate	Age	
Allan	2017-06-19 00:00:00	23	Edit Details Delete
Carson	2017-06-19 00:00:00	45	Edit Details Delete
Carson	2017-06-19 00:00:00	37	Edit Details Delete
Laura	2017-06-19 00:00:00	26	Edit Details Delete

© 2017 - My ASP.NET Application

Steg 4: Klicka på Employees List, lista över anställda skall visas.

Steg 5: Stäng ner webbläsare.

Steg 6: Avsluta debugging.

Steg 7: Spara undan din lösning.