Unit Testing in ASP.NET MVC Applications

.NET Framework 4 2 out of 3 rated this helpful

A significant benefit of using the MVC pattern in ASP.NET is that you can easily implement unit tests for your Web application. This is particularly true in comparison to the ASP.NET Web Forms page model, where unit testing is more complex because it is difficult to isolate specific functionality, and because testing Web Forms pages requires that you invoke the Web server and run the complete page pipeline. ASP.NET MVC has been architected for testability without dependencies on the IIS server, on a database, or on external classes.

When you create a new ASP.NET MVC project in Visual Studio, the **Create Unit Test Project** dialog box is displayed. If you select **Yes** and create unit tests, a test project will be created in your ASP.NET MVC solution that contains unit tests for the account controller and the home controller. These test classes provide a good introduction to MVC unit testing.

ASP.NET MVC unit tests directly call methods of your MVC controllers. When a unit test calls an action method in a controller, you can validate that the correct view is returned (although you do not validate the HTML) and that view data is returned. You can also test whether a method correctly redirects to another controller or view.

Visual Studio provides the Visual Studio Unit Test framework in all editions except Standard Edition and Express Edition. However, you might already be familiar with third-party test frameworks such as NUnit, MbUint, or XUnit, and with third-party mock-object libraries such as Rhino mocks, Type mocks, or NMock. In versions of Visual Studio that support unit-test projects, you can create a custom test project template that will then be available as a project option when you create new ASP.NET MVC projects. The custom test project can use a unit-test framework that you specify. In addition, you can include other libraries in the custom test project, such as a mock-object framework, a personal library of unit-test code, and so on.

Related Topics

Title	Description
Walkthrough: Using TDD with ASP.NET MVC	Provides step-by-step procedures that show you how to get started with test-driven development (TDD) in ASP.NET MVC.
Building Testable ASP.NET MVC Applications	An article in MSDN Magazine online that provides an introduction to using TDD with MVC. (Does not include information about the Design by Example pattern.)
How to: Add a Custom ASP.NET MVC Test Framework in Visual Studio	Explains how to build testable MVC applications using third-party tools.
Using Mocks And Tests To	Explains how to test object interactions without implementing the

Design Role-Based Objects	object.
Unit Testing Framework	Explains the unit-testing framework that is integrated with Visual Studio.

Community Additions