# Exercise: Tesing ASP.Net WEB API

Problems for exercises and homework for the ["Back-End Technologies Basics"](https://softuni.bg/trainings/4398/back-end-technologies-basics-january-2024) course @ SoftUni.

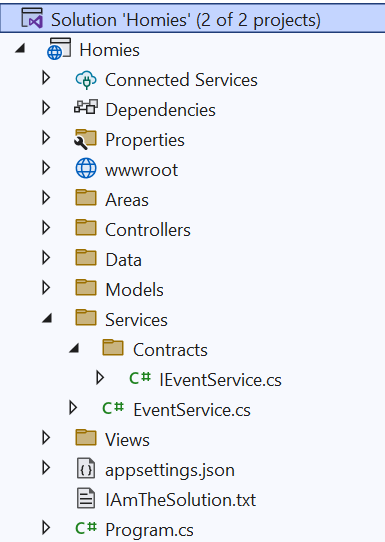
In this part of the lab, you will familiarize yourself with the **ASP.NET Web API** **application**, learn how to run it in **Visual Studio**, and **connect it to a database** using **Microsoft SQL Server Management Studio** (SSMS)..

## Set Up the Testing Environment

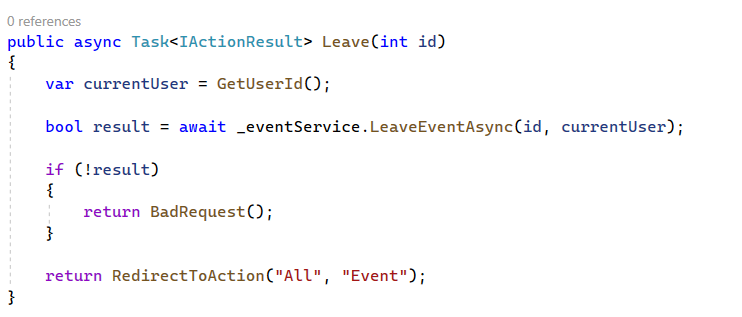
* Make sure you have a testing environment set up within your solution. This typically involves creating a separate test project where you will write your tests.

## Introduction to IEventService

* In the Homies application, we have introduced an IEventService interface to separate the business logic related to events from the controllers. This interface defines the contract that any implementation of the event service must adhere to. By using an interface, we can decouple the implementation details of the service from the components that depend on it, such as controllers or other services.

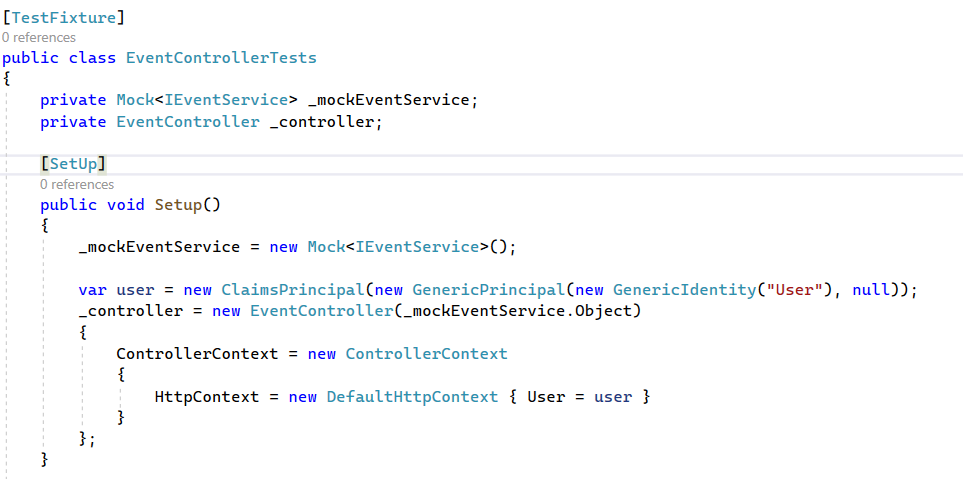


## Separation of Concerns

* One of the key benefits of using an interface like **IEventService** is that it helps to enforce the principle of **separation of concerns**. By **separating the business logic** related to events into its own service, we can keep our **controllers focused on handling HTTP requests and responses**, while delegating the complex logic of event management to the service layer.
* 

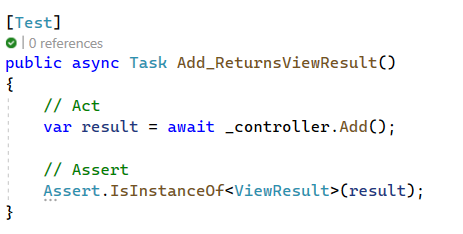
## EventControllerTests.cs

* The EventControllerTests class is a collection of unit tests designed to verify the behavior of the EventController class in the Homies application. This class utilizes the NUnit testing framework along with Moq to mock dependencies and isolate the controller for testing purposes.



### Add\_ReturnsViewResult

* The **Add\_ReturnsViewResult** test method is designed to verify the behavior of the **Add action method** in the **EventController** class. This test ensures that when the Add action method is invoked, it returns a view result to the client, indicating that the operation was successful.
* **The primary** **purpose of this test** is to ensure that the Add action method functions correctly and returns the expected result, which is a view that can be rendered in the user's browser. By validating this behavior, developers can verify that the controller action behaves as intended and provides the necessary user interface for adding new events.



* **Importance of the test:**
  + **Functional Verification**
  + **User Interface Validation**
  + **Prevents Regression**

## EventServiceTests.cs

* The **EventServiceTests** class is responsible for testing the behavior of the EventService class in the Homies application. This class contains unit tests written using the NUnit testing framework to ensure that the event service behaves as expected in different scenarios.



* **AddEventAsync\_ShouldAddEvent\_WhenValidEventModelAndUserId**
* The AddEventAsync\_ShouldAddEvent\_WhenValidEventModelAndUserId test method is responsible for verifying the behavior of the AddEventAsync method in the EventService class. This test ensures that when the AddEventAsync method is called with valid event model data and a user ID, it successfully adds the event to the database.
* **The primary purpose of this test** is to verify that the AddEventAsync method correctly adds an event to the database when valid event model data and a user ID are provided. By validating this behavior, developers can ensure that events are correctly stored in the database and that the EventService class functions as expected.



* **Importance of the test:**
  + **Database Interaction Verification**
  + **Data Integrity Validation**
  + **Business Logic Validation**