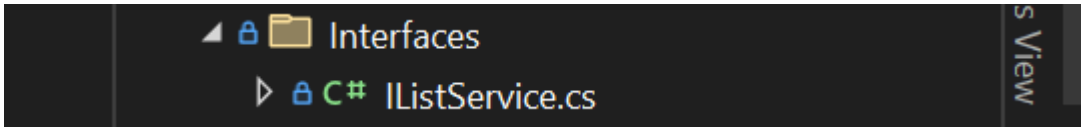
Beginner's guide to Unit Testing with NUnit.

I'll be going through a short introduction to Unit Testing Using NUnit. To make it easy, we'll be testing simple CRUD(Create, Read, Update, Delete) Operations on a List.

Let's Begin

First Create a new Console Application, I have named mine UnitTestingLists

Next in your UnitTestingLists project, click on your Solution Explorer and create a folder named Interfaces, this will hold the interface our service will use. Then add a file to that folder named IListService

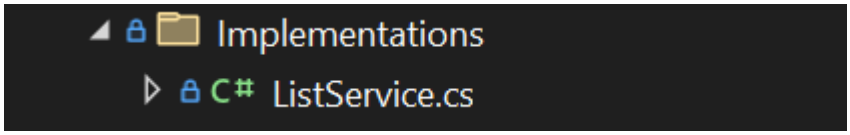


Next in your IListService, add the following code

```
public interface IListService
{
   void Add(string item);
   void Remove(string item);


   void Edit(string oldItem,  string newItem);


   List<string> GetItems();
}
```

These are all the operations we will create.

Next, in your UnitTestingLists project, create a folder named implementations. Then in this folder create a class called ListService. This is the service that Implements our Interface.



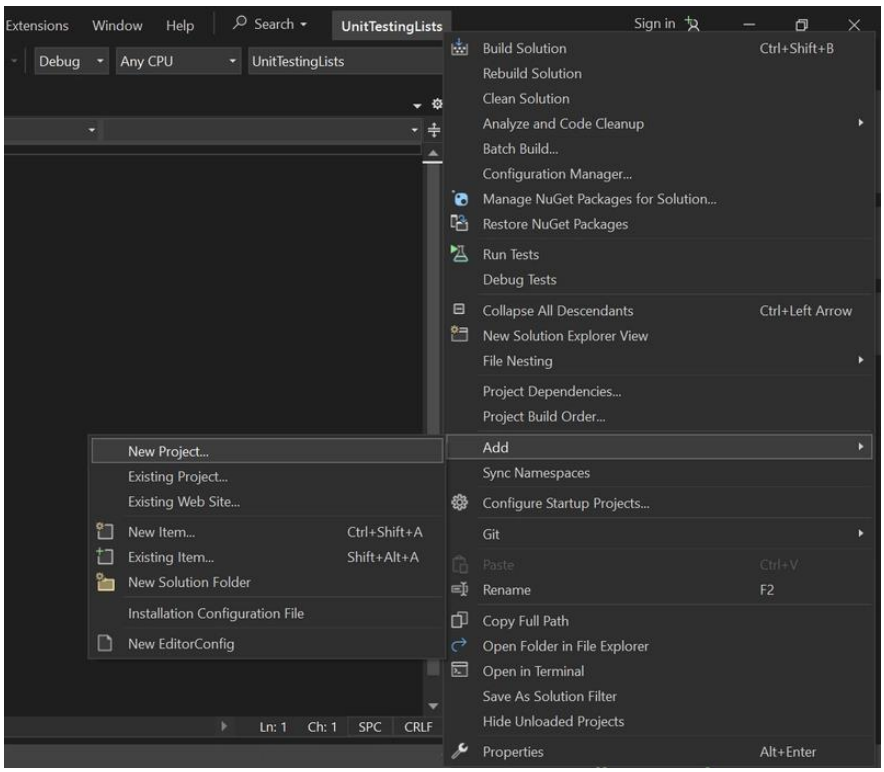
Then in your ListService class, add the following code

```
public class ListService : IListService
{
   private static List<string> _list;
   public ListService()
   {
      _list = new List<string> {"Apple", "Orange", "Pear", "Watermelon", "Lemon","Lime" };
   }
   public void Add(string item)
   {
      _list.Add(item);
   }
```

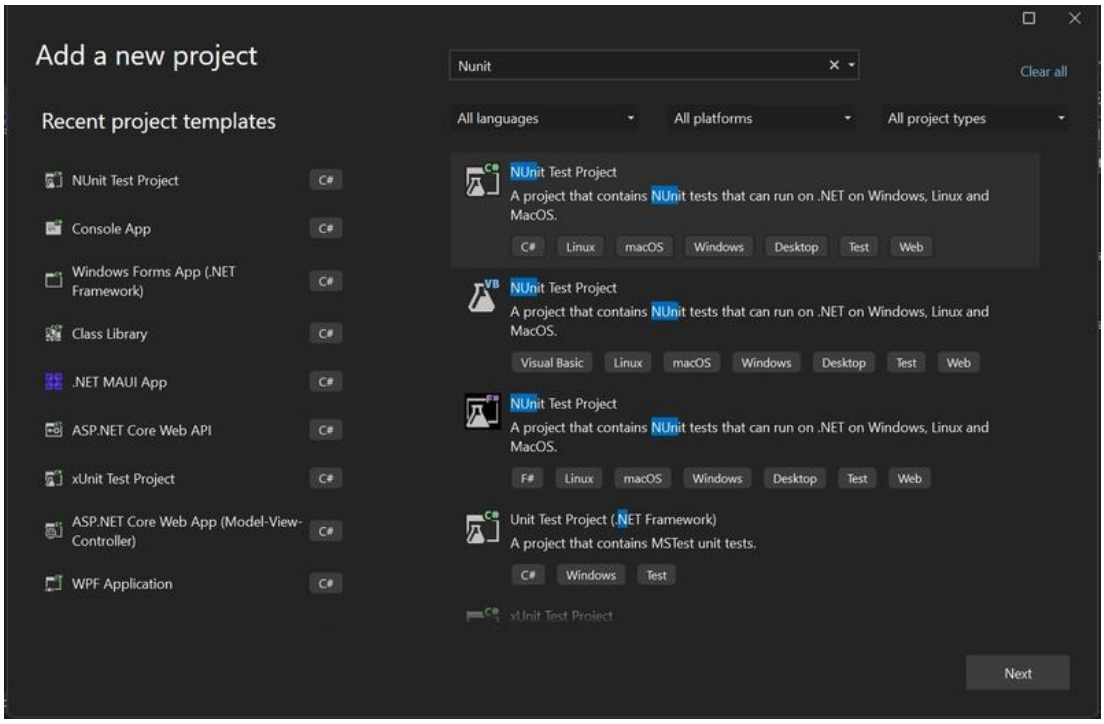
```
    public void Edit(string oldItem, string newItem)
    {
        if (_list?.Count > 0 && _list.Contains(oldItem)){
            int indexToRemove = _list.IndexOf(oldItem);
            _list[indexToRemove] = newItem;
        }
    }

    public List<string> GetItems()
    {
        return _list;
    }

    public void Remove(string item)
    {
        _list.Remove(item);
    }
}
```

We have added a lot of code so let's go over it. We created a static list that we will work with and added some items. We also added some simple methods that do different operations on our list.

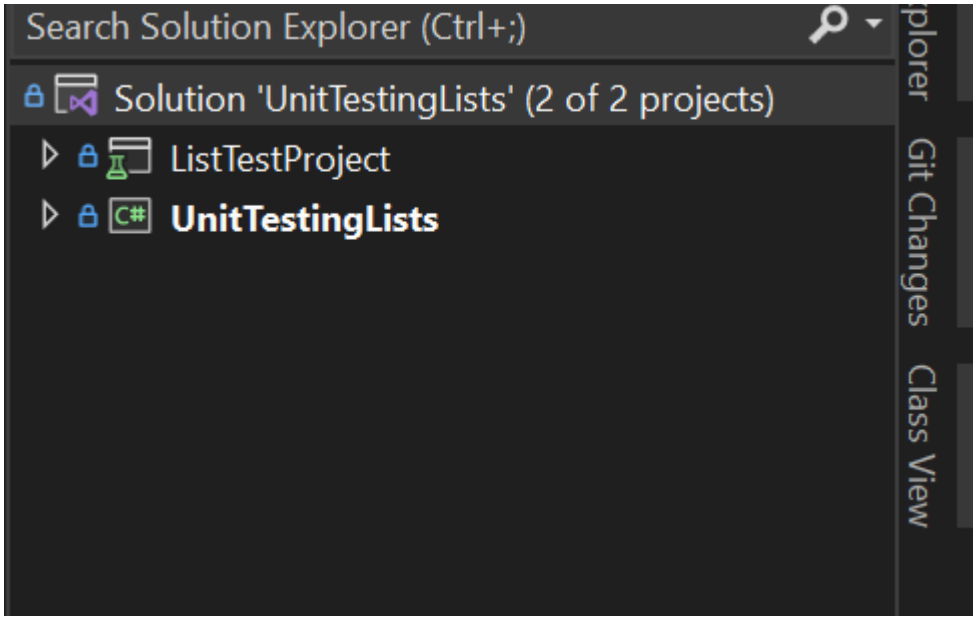
Great, Next let's create a test project. Right-click on your solution and go to Add, then click on Add New Project



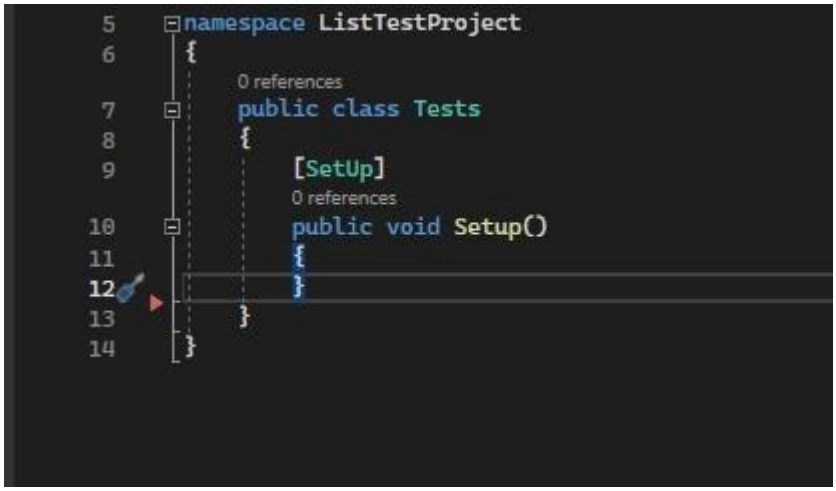
In the Add New Project window, search for "Nunit" and Click on "NUnit Test Project"



Give your project a name and keep clicking "next" and then create. I have called mine "ListTestProject". So far, your entire project should look something like this.



OK, in your ListTestProject there should be a file called UnitTest1, when you click that, you should have something like this

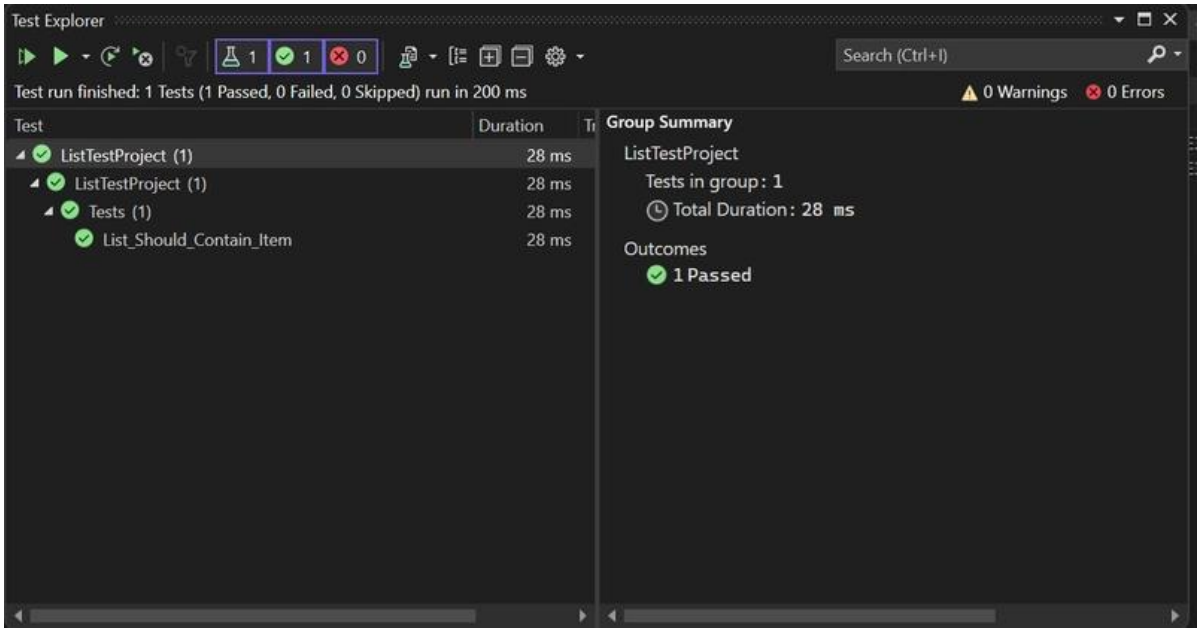


Let's go ahead and write our first test.
Every test method has the decorator [Test] to indicate it is a test, so go ahead and add this code,

```
[Test]
public void List_Should_Contain_Item()
{
    //Arrange
    var listService = new ListService();
    string fruit = "Watermelon";
    List<string> list = mockService.GetItems();

    //Act
    bool isThere = list.Contains(fruit);

    //Assert
    Assert.IsTrue(isThere);
}
```

This is a test to check that one of the items we added when creating our listService is present.

Now, go ahead and right on the test project and click on Run Tests.



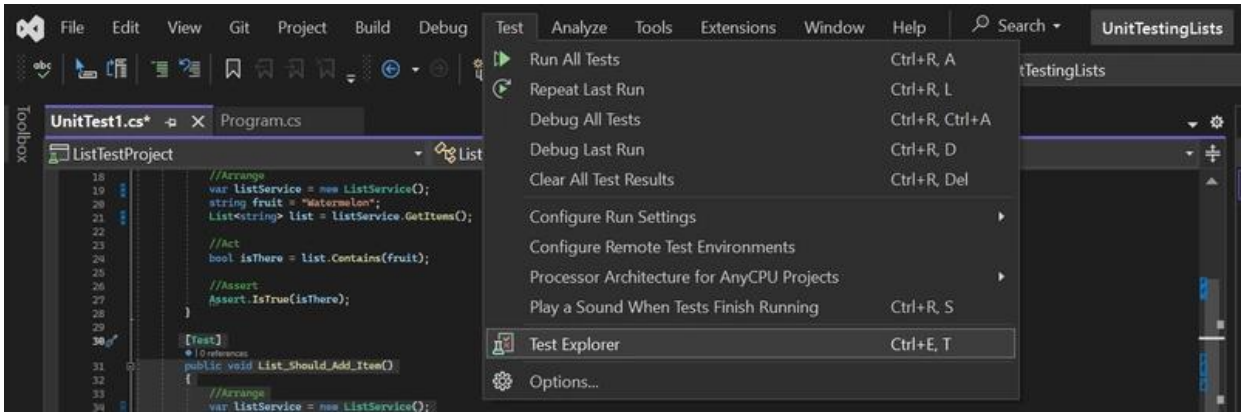
As you can see, the item is in the list so we get a passing Test

Let's add a couple more tests and run them, Add the following code to your UnitTest1 test class

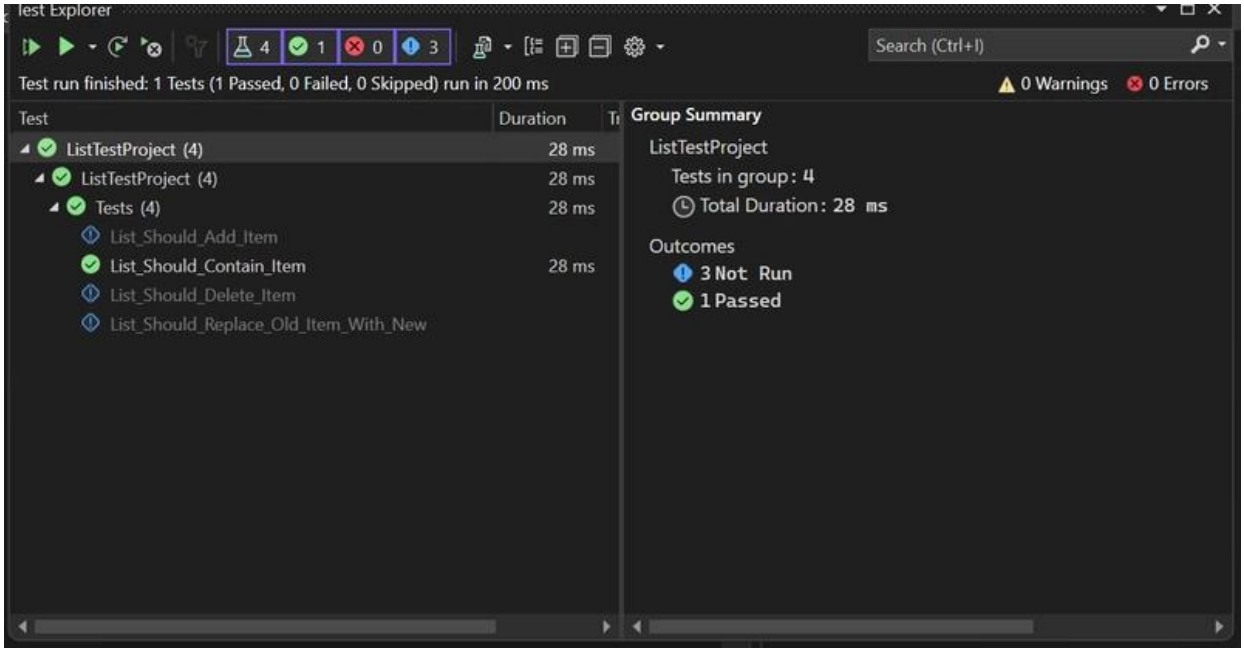
```
[Test]
public void List_Should_Add_Item()
{
    //Arrange
    var listService = new ListService();
    string fruit = "Grapes";
    List<string> list = listService.GetItems();

    //Act
    listService.Add(fruit);

    //Assert
    Assert.Contains(fruit, list);
}

[Test]
public void List_Should_Replace_Old_Item_With_New()
{
    //Arrange
    var listService = new ListService();
    List<string> list = listService.GetItems();
    int indexOfRemove = 1;
    string oldFruit = "Orange";
```

```
    string newFruit = "Strawberry";

    //Act
    listService.Edit(oldFruit, newFruit);

    //Assert
    Assert.That(list[indexOfRemove], Is.EqualTo(newFruit));
}

[Test]
public void List_Should_Delete_Item()
{
    //Arrange
    var listService = new ListService();
    List<string> list = listService.GetItems();
    string deleteFruit = "Pear";

    //Act
    listService.Remove(deleteFruit);

    //Assert
    Assert.IsFalse(list.Contains(deleteFruit));
}
```

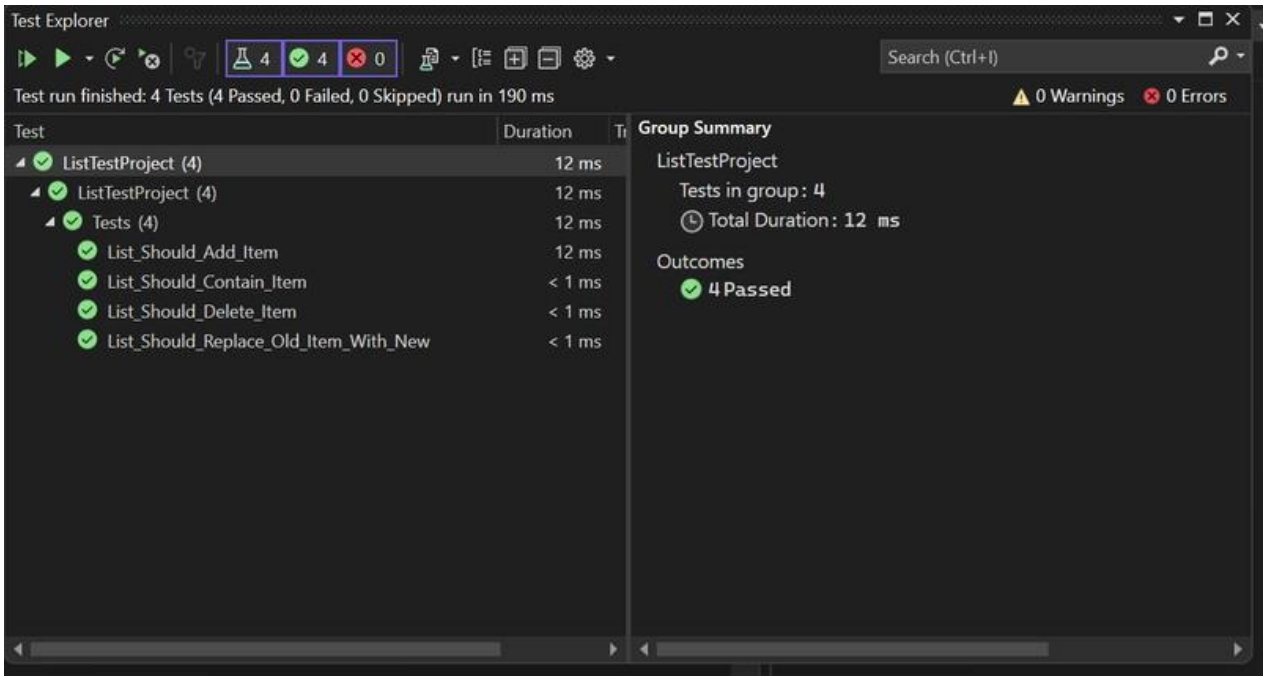
Alright, we added 3 more tests to make sure the operations in our ListService are working correctly.
So instead of running your tests from the project itself, in the top menu bar, click on test and open the Test explorer



Your test Explorer should then open this way



This way, you can then choose to run all tests or run tests Individually.
Let's Run all our tests to see.



And we see that all our tests passed.
You can decide to add more meaningful tests to ascertain other functions.
If you would like the source code used, you can get it here.