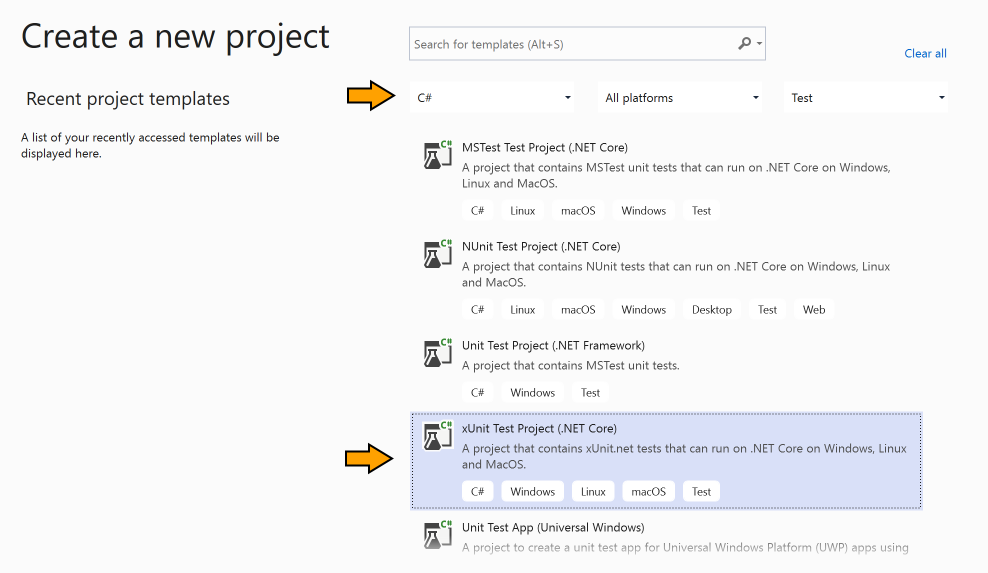
Getting Started With xUnit.net

Using .Net Core With Visual Studio

**Create a xUnit Test Project**

****

**Why are Using xUnit?**

Your First Unit Test with xUnit. The . NET Core platform supports different testing frameworks. However, xUnit has become the most popular due to its simplicity, expressiveness, and extensibility

**Why should I Use xUnit?**

Unit Test cases reveal if anything is wrong with software design, software functionality and if it is not working as expected as per the requirements. Due to the help of the Unit Testing, we can get or catch the bugs or issues early before deploying the code to another environment and fixing it.

**what is the unit testing with in xUnit in c#**

Unit Testing is a software testing approach which performs at the time of the development to test the smallest component of any software. It means rather than testing the big module in one go, you test the small part of that module.

🡺xUnit.net is a free, open source, community-focused unit testing tool for the . NET Framework. Written by the original inventor of NUnit v2, xUnit.net is the latest technology for unit testing C#

**How do I start xUnit?**

**Getting Started with xUnit.net**

1. Download the .NET SDK.
2. Download Mono (non-Windows machines)
3. Create the xunit test project.
4. Setting up a unit test MSBuild target.
5. Write your first tests.
6. Write your first theory.
7. Run tests with Text Exploer

**[Fact]** – attribute states that the method should be executed by the test runner

[**Theory**] – attribute implies that we are going to send some parameters to our testing code. So, it is similar to the [Fact] attribute, because it states that the method should be executed by the test runner, but additionally implies that we are going to send parameters to the test method

[**InlineData**] – attribute provides those parameters we are sending to the test method. If we are using the [Theory] attribute, we have to use the [InlineData] as well

**Create The xUnit Project**

**After Create Project we can come like this**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**What is Fact in unit testing C#?**

The [Fact] attribute declares a test method that's run by the test runner.

**What is inline data in xUnit?**

The InlineData attribute is commonly used to supply data to a Theory attribute-based xUnit parameterized test. Every instance of the InlineData attribute creates a separate occurrence of the test method

After we are Writing the Code In xUnit We have Excute The Text Exploer And We Have Run Must IN Graphical user interface, text, application, email

Description automatically generated

After We Run The Pogramma we Have Check Like This In Text Exploer

Graphical user interface, application

Description automatically generated

Errors will come like this

Graphical user interface, application

Description automatically generated

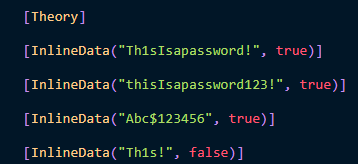
## **Write your first theory**

You may have wondered why your first unit tests use an attribute named [Fact] rather than one with a more traditional name like Test. xUnit.net includes support for two different major types of unit tests: facts and theories. When describing the difference between facts and theories, we like to say:

***Facts* are tests which are always true. They test invariant conditions.**

***Theories* are tests which are only true for a particular set of data.**

**We Can Create Theory this Like ?**

****

**By using the**[Theory]**attribute, we are stating that we are going to provide some data to this test method as a parameter. Additionally, with the**[InlineData]**attribute, we are providing concrete data for the test method.**

**TL;DR:** This article will guide you in creating automated tests with xUnit for your C# applications. You will learn the basics of automated tests and how to create unit and integration tests.

Regarding the way to structure your automated tests, a typical approach follows the so-called [**AAA** pattern](http://wiki.c2.com/?ArrangeActAssert). The name comes from the initials of the three actions usually needed to perform a test:

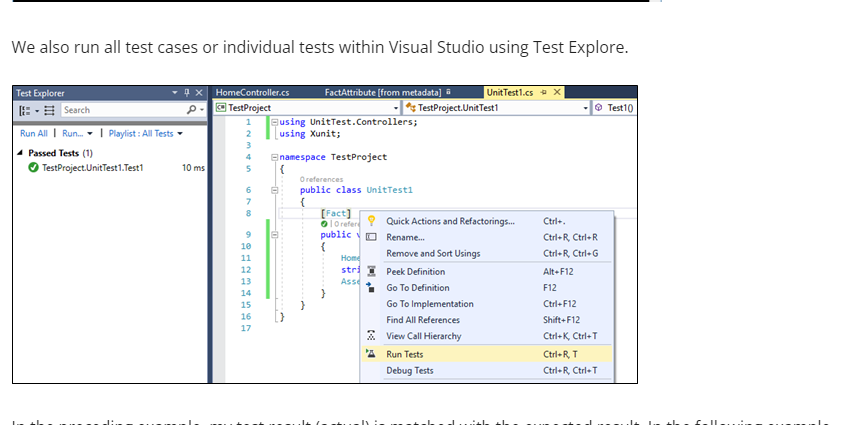
* **Arrange**. With this action, you prepare all the required data and preconditions.-🡪**Set values**
* **Act**. This action performs the actual test.
* **Assert**. This final action checks if the expected result has occurred.-🡪 **Calling the arrange and act**

## Your First Unit Test with xUnit

The .NET Core platform supports different testing frameworks. However, [xUnit](https://xunit.net/" \t "_blank) has become the most popular due to its simplicity, expressiveness, and extensibility. The project is supported by the [.NET Foundation](https://dotnetfoundation.org/), and it is part of the more recent versions of .NET Core. This means that you don't need to install anything but the [.NET Core SDK](https://dotnet.microsoft.com/download).

We Have Run these Like Also

**We also run all test cases or individual tests within Visual Studio using Test Explore.**



**Tips to write a Test Case.**

1. Don’t write Unit Test Cases in the same project, create a separate test project for Unit Testing.
2. Unit Test Cases should be well organized and maintainable.
3. Write Unit Test Cases only for small functionality.
4. If a function is performing so many operations then just write Unit Test Case for each individual function,
5. Don’t write Unit Test Cases which are dependent on another Unit Test Cas.
6. The name of the function for Unit Test Case should be self-explanatory.
7. Unit Test Cases should always be independent.
8. Performance wise, Unit Test Case should always be fast.