

xUnit.net Framework

xUnit.net je moderní testovací nástroj pro platformu .NET, navržený s důrazem na čistotu kódu, rozšiřitelnost a soulad s moderními programovacími praktikami. Jeho autory jsou James Newkirk (původní autor NUnit) a Brad Wilson.

1. Hlavní filozofie xUnit

Na rozdíl od starších frameworků přináší xUnit několik zásadních změn, které vedou k lepším testovacím návykům:

- **Bez atributu [TestFixture]:** Třída s testy nepotřebuje speciální označení. Každá veřejná třída může obsahovat testy.
- **Izolace testů:** Pro každý jednotlivý test se vytváří **nová instance třídy**. Tím je zaručeno, že testy neovlivňují jeden druhého skrze sdílený stav (vlastnosti třídy).
- **Konstruktor místo [SetUp]:** Namísto speciálních metod pro přípravu prostředí se používá standardní konstruktor třídy. Pro úklid (cleanup) se používá rozhraní `IDisposable``.

2. Základní atributy a syntaxe

xUnit rozlišuje dva hlavní typy testů:

[Fact]

Používá se pro testy, které jsou vždy pravdivé (nemají žádné parametry).

```
[Fact]
public void Je_Sude_Cislo_Vraci_True() {
    // Arrange
    int cislo = 4;
    // Act & Assert
    Assert.True(cislo % 2 == 0);
}
```

[Theory]

Používá se pro data-driven testy (testy s parametry). To umožňuje spustit stejný test s různými vstupními daty.

```
[Theory]
[InlineData(4, true)]
```

```
[InlineData(5, false)]
public void Test_Sudosti(int cislo, bool ocekavanyVysledek) {
    bool vysledek = (cislo % 2 == 0);
    Assert.Equal(ocekavanyVysledek, vysledek);
}
```

3. Assertace (Ověřování)

xUnit používá statickou třídu `Assert`. Mezi nejpoužívanější metody patří:

- `Assert.Equal(expected, actual)` - Porovnání hodnot.
- `Assert.Throws<ExceptionType>(() => ...)` - Ověření, že kód vyvolá očekávanou výjimku.
- `Assert.Contains(substring, string)` - Kontrola, zda text obsahuje podřetězec.
- `Assert.Empty(collection)` - Kontrola, zda je kolekce prázdná.

4. Sdílení kontextu (Fixtures)

Pokud potřebujete sdílet nákladnou operaci (např. připojení k databázi) mezi více testy, xUnit k tomu nabízí mechanismus **Class Fixtures**:

- Vytvoříte třídu `DatabaseFixture`.
- Testovací třída implementuje rozhraní `IClassFixture<DatabaseFixture>`.
- Framework předá instanci fixture do konstruktoru testovací třídy.

5. Srovnání s NUnit a MSTest

Funkce	xUnit	NUnit	MSTest
Značka testu	<code>[Fact]</code>	<code>[Test]</code>	<code>[TestMethod]</code>
Příprava (Setup)	Konstruktor	<code>[SetUp]</code>	<code>[TestInitialize]</code>
Izolace	Nová instance pro každý test	Stejná instance	Stejná instance
Parametrické testy	<code>[Theory]</code>	<code>[TestCase]</code>	<code>[DataTestMethod]</code>

6. Integrace v IDE

xUnit je plně integrován do:

- **Visual Studio:** Pomocí Test Exploreru.
 - **VS Code:** Pomocí rozšíření .NET Core Test Explorer.
 - **Příkazová řádka:** Pomocí příkazu ``dotnet test``.
-

Související články:

- [Unit Testing a kvalita kódu](#)
- [Principy SOLID](#)
- [Mockování v .NET \(Moq, NSubstitute\)](#)

Tagy: `programming dot-net testing xunit unit-test csharp`

From:

<http://serviceit.cz/> - **IT ENCYKLOPEDIÉ**

Permanent link:

<http://serviceit.cz/doku.php?id=xunit>

Last update: **2026/01/02 18:55**

