

Determining Passing and Failing Tests with Asserts



Jason Roberts

.NET MVP

@robertsjason dontcodetired.com



Overview



An overview of asserts

How many asserts per test

Booleans

Strings

Numeric values

Nulls

Collections

Object types and object instances

Thrown exceptions

Raised events



Asserts

Evaluate and verify the outcome of a test, based on a returned result, final object state, or the occurrence of events observed during execution. An assert can either pass or fail. If all asserts pass, the test passes; if any assert fails the test fails.





Boolean values

- True / false

String values

- Equality, inequality
- Empty
- Starts with / ends with
- Contains substring
- Matches regular expression

Numeric values

- Equality, inequality
- In a given range
- Floating point precision



Collection contents

- Equality with another collection
- Contains / does not contain
- Contains item satisfying predicate
- All items satisfy an assert

Raised events

- Custom events
- Framework events (e.g. PropertyChanged)

Object type

- Is / not exact specified type
- Is / not type or derived type

```
public class PlayerCharacterShould
{
    [Fact]
    public void IncreaseHealthAfterSleeping()
    {
        PlayerCharacter sut = new PlayerCharacter();

        sut.Sleep();

        Assert.InRange(sut.Health, 101, 200);
        Assert.True(sut.IsNoob);
    }
}
```



Summary



How many asserts per test

```
Assert.True(sut.IsNoob);
```

```
Assert.StartsWith("Sarah", sut.FullName);
```

```
Assert.InRange(sut.Health, 101, 200);
```

```
Assert.Null(sut.Nickname);
```

```
Assert.Contains("Long Bow", sut.Weapons);
```

```
Assert.IsType<BossEnemy>(enemy);
```

```
Assert.NotSame(enemy1, enemy2);
```

```
Assert.Throws<ArgumentNullException>(()  
=> sut.Create(null));
```

```
Assert.raises<EventArgs>(...)
```



Next:

Understanding and Controlling Test Execution

