

# Unit Testing

<https://fluentassertions.com/introduction>

```
//Date
result.Should().BeAfter(1.January(2010));
result.Should().BeBefore(1.January(2005));
```

```
//integer
result.Should().Be(expected);
result.Should().BeGreaterThanOrEqualTo(2);
result.Should().NotBeInRange(-10000, 0);
```

```
//object
return new PingOptions()
{
    DontFragment = true,
    Ttl = 1
};

//Arrange
var expected = new PingOptions()
{
    DontFragment = false,
    Ttl = 5
}

//Act
var result = _pingService.GetPingOptions();
result.
//Assert
result.Should().BeOfType<PingOptions>();
result.Should().BeEquivalentTo(expected);
result.Ttl.Should().Be(1);
```

```
//list
public IEnumerable<PingOptions> ReturnListOfPings()
{
    IEnumerable<PingOptions> pingOpts = new List<PingOptions>() {
        new PingOptions()
        {
            DontFragment = true,
            Ttl = 1
        },
        new PingOptions()
        {
            DontFragment = true,
            Ttl = 4
        },
        new PingOptions()
        {
            DontFragment = true,
            Ttl = 5
        },
        new PingOptions()
        {
            DontFragment = true,
```

```

        Ttl = 10
    }
    };
    return pingOpts;
}

[Fact]
private void Calculator_ReturnListOfPings()
{
    var expected = new PingOptions()
    {
        DontFragment = true,
        Ttl = 1
    };
    //Act
    var result = _obj.ReturnListOfPings();

    //Assert
    result.Should().BeOfType<List<PingOptions>>();
    result.Should().ContainEquivalentOf(expected);
    result.Should().Contain(x => x.DontFragment == true);
}

```

```

[Fact]
private void Calculator_ReturnListOfPings()
{
    var expected = new PingOptions()
    {
        DontFragment = true,
        Ttl = 1
    };
    //Act
    var result = _obj.ReturnListOfPings();

    //Assert
    result.Should().BeOfType<List<PingOptions>>();
    result.Should().ContainEquivalentOf(expected);
    result.Should().Contain(x => x.DontFragment == true);
}

```