

```

using System;
using System.Collections.Generic;

namespace IdentifiableObject
{
    public class IdentifiableObject
    {
        private readonly List<string> _idents = [];

        public IdentifiableObject(string[] idents)
        {
            foreach (string s in idents)
            {
                AddIdentifier(s);
            }
        }

        public bool AreYou(string id)
        {
            return _idents.Contains(id.ToLower());
        }

        public string FirstID
        {
            get
            {
                if (_idents.Count == 0)
                {
                    return "";
                }
                else
                {
                    return _idents[0];
                }
            }
        }

        public void AddIdentifier(string id)
        {
            _idents.Add(id.ToLower());
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using NUnit.Framework;

namespace IdentifiableObject
{
    [TestFixture]
    public class TestIdentifiableObject
    {
        [Test]
        public void TestAreYou()
        {
            string[] testIdentifiableObjectArray = ["Fred", "Bob"];
            IdentifiableObject testObject = new(testIdentifiableObjectArray);
            Assert.That(testObject.AreYou("fred"), Is.True);
        }

        [Test]
        public void TestNotAreYou()
        {
            string[] testIdentifiableObjectArray = ["Fred", "Bob"];
            IdentifiableObject testObject = new(testIdentifiableObjectArray);
            Assert.That(testObject.AreYou("wilma"), Is.False);
        }

        [Test]
        public void TestCaseSensitive()
        {
            string[] testIdentifiableObjectArray = ["Fred", "Bob"];
            IdentifiableObject testObject = new(testIdentifiableObjectArray);
            Assert.That(testObject.AreYou("bOB"), Is.True);
        }

        [Test]
        public void TestFirstID()
        {
            string[] testIdentifiableObjectArray = ["Fred", "Bob"];
            IdentifiableObject testObject = new(testIdentifiableObjectArray);
            StringAssert.AreEqualIgnoringCase("fred", testObject.FirstID);
        }

        [Test]
        public void TestFirstIDWithNoIDs()
        {
            string[] testIdentifiableObjectArray = [];
            IdentifiableObject testObject = new(testIdentifiableObjectArray);
            StringAssert.AreEqualIgnoringCase("", testObject.FirstID);
        }

        [Test]
        public void TestAddID()
        {
            string[] testIdentifiableObjectArray = ["Fred", "Bob"];
            IdentifiableObject testObject = new(testIdentifiableObjectArray);
            testObject.AddIdentifier("Wilma");
            Assert.Multiple(() =>
            {
                Assert.That(testObject.AreYou("fred"), Is.True);
                Assert.That(testObject.AreYou("bob"), Is.True);
            }
        }
    }
}

```

```

    Assert.That(testObject.AreYou("wilma"), Is.True);
  });
}
}
}

```

Test Explorer

Test run finished: 6 Tests (6 Passed, 0 Failed, 0 Skipped) run in 297 ms

Search (Ctrl+I)

0 Warnings 0 Errors

Test	Duration	Traits	Error Message
IdentifiableObjectTest (6)	67 ms		
IdentifiableObject (6)	67 ms		
TestIdentifiableObject (6)	67 ms		
TestAddId	59 ms		
TestAreYou	< 1 ms		
TestCaseSensitive	< 1 ms		
TestFirstID	8 ms		
TestFirstIDWithNoIDs	< 1 ms		
TestNotAreYou	< 1 ms		

Group Summary

IdentifiableObjectTest

Tests in group: 6

Total Duration: 67 ms

Outcomes

6 Passed

17°C
Cố mây

Search

ENG
US

8:04 PM
2/04/2024