



SISTEMAS DE INFORMAÇÕES

Niceu Santos Biriba

Atividade 1 – Testes Unitários e o Stack Overflow

SÃO CRISTÓVÃO
2024

1. Introdução

Este documento descreve a solução de um problema relacionado a testes de unidade, conforme solicitado pelo professor. A atividade envolve a reprodução de um problema encontrado no Stack Overflow através da String de busca "[unit-testing] or [junit] or [pytest]", a aplicação da solução proposta e a criação de um tutorial detalhado sobre o processo.

2. Escolha do Problema

Pergunta Selecionada:

Título: I need to write a unit test for a method that takes a stream which comes from a text file. I would like to do something like this:

```
Stream s = GenerateStreamFromString("a,b \n c,d");
```

c# unit-testing string stream

URL: <https://stackoverflow.com/questions/1879395/how-do-i-generate-a-stream-from-a-string>

Número de Votos: 400+

Resposta Aceita:

```
public static Stream GenerateStreamFromString(string s)
{
    var stream = new MemoryStream();
    var writer = new StreamWriter(stream);
    writer.Write(s);
    writer.Flush();
    stream.Position = 0;
    return stream;
}
```

Outras Soluções Consideradas:

1 - É uma solução eficiente e direta para converter uma string em um stream, mas não permite customização do encoding e não oferece suporte a operações adicionais de escrita.

```
public static MemoryStream GenerateStreamFromString(string value)
{
    return new MemoryStream(Encoding.UTF8.GetBytes(value ?? ""));
}
```

2 - O StreamWriter adiciona um nível extra de abstração e permite escrever dados em um stream de forma mais flexível, mas é excessivo para o caso dessa solução que é uma conversão simples.

```
public static Stream ToStream(this string str)
{
    MemoryStream stream = new MemoryStream();
    StreamWriter writer = new StreamWriter(stream);
    writer.Write(str);
    writer.Flush();
    stream.Position = 0;
    return stream;
}
```

3 - É semelhante a primeira abordagem e é uma forma direta e eficiente, mas não inclui opções para customização do encoding.

```
public static Stream ToStream(this string str, Encoding enc = null)
{
    enc = enc ?? Encoding.UTF8;
    return new MemoryStream(enc.GetBytes(str ?? ""));
}
```

And then I use it like this:

```
String someStr="This is a Test";
Encoding enc = getEncodingFromSomewhere();
using (Stream stream = someStr.ToStream(enc))
{
    // Do something with the stream...
}
```

```
var ms = new MemoryStream("some string".u8.ToArray());
```

4. É também uma solução flexível que permite suportar diferentes encodings, tornando-a útil quando se lida com strings em vários formatos de codificação.

```
private Stream GenerateStreamFromString(String p)
{
    Byte[] bytes = UTF8Encoding.GetBytes(p);
    MemoryStream strm = new MemoryStream();
    strm.Write(bytes, 0, bytes.Length);
    return strm;
}
```

5. É uma ótima e elegante solução para o problema, pois é extensível, o que melhora a legibilidade e reutilização do código ao adicionar funcionalidades diretamente às strings.

```
public static class StringExtensions {

    public static Stream ToStream(this string s) {
        return s.ToStream(Encoding.UTF8);
    }

    public static Stream ToStream(this string s, Encoding encoding) {
        return new MemoryStream(encoding.GetBytes(s ?? ""));
    }
}
```

3. Reproduzindo e Resolvendo o Problema

3.1 Configuração do Projeto

Criação do Projeto:

1. Utilizei a IDE do Microsoft Visual Studio.
2. Criei um novo projeto do tipo ASP .NET Core Web em C# chamada "Aplication".
3. Adicionei um projeto de testes utilizando MSTest Test Project ao mesmo repositório chamada "TestProject".

3.2 Adição da Função *GenerateStreamFromString*:

Adicionei uma página do tipo .cs e inseri a resolução do teste que foi respondido na pergunta.

```
using System.Text;

namespace Teste_Software_2024_Biriba_Niceu
{
    1 reference
    public static class StreamHelper
    {
        1 reference
        public static Stream GenerateStreamFromString(string s)
        {
            var stream = new MemoryStream();
            var writer = new StreamWriter(stream);
            writer.Write(s);
            writer.Flush();
            stream.Position = 0; // Reposiciona o stream para o início
            return stream;
        }
    }
}
```

3.3 Criação do Projeto de Testes:

No projeto de testes adicionei uma classe de testes.

Código da Classe de Teste:

```
using Teste_Software_2024_Biriba_Niceu;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System.IO;
using System.Text;

namespace Teste_Software_2024_Biriba_Niceu.Tests
{
    [TestClass]
    0 references
    public class StreamHelperTests
    {
        [TestMethod]
        0 references
        public void GenerateStreamFromString_ShouldReturnStreamContainingInputString()
        {
            // Arrange
            string inputString = "a,b\\nc,d";
            Stream expectedStream = new MemoryStream(Encoding.UTF8.GetBytes(inputString));

            // Act
            Stream actualStream = StreamHelper.GenerateStreamFromString(inputString);

            // Assert
            using (StreamReader expectedReader = new StreamReader(expectedStream))
            using (StreamReader actualReader = new StreamReader(actualStream))
            {
                string expectedContent = expectedReader.ReadToEnd();
                string actualContent = actualReader.ReadToEnd();

                Assert.AreEqual(expectedContent, actualContent);
            }
        }
    }
}
```

3.4 Execução dos Testes:

Executei o teste para conferir se estava tudo ok.

