

Instruções para Configuração e Execução de um Teste Básico com Selenium, xUnit e Microsoft TestPlatform usando C# no Visual Studio Code

Passo 1: Instalar o SDK do .NET

Certifique-se de que você tem o SDK do .NET instalado no seu sistema. Você pode baixá-lo em <https://dotnet.microsoft.com/download>.

Passo 2: Instalar o VS Code

Baixe e instale o VS Code em <https://code.visualstudio.com/>.

Passo 3: Instalar a Extensão C# para o VS Code

1. Abra o VS Code.
2. Vá para a visualização de Extensões clicando no ícone de Extensões na Barra de Atividades na lateral da janela ou pressionando Ctrl+Shift+X.
3. Procure por "C#" e instale a extensão oficial da Microsoft.

Passo 4: Criar um Novo Projeto de Teste com xUnit

1. Abra um terminal no VS Code (Ctrl+` ou via o menu Terminal).
2. Crie um novo projeto de teste com xUnit executando:
`dotnet new xunit -n SeleniumReportXunitTest`
3. Navegue até o diretório do projeto:
`cd SeleniumReportXunitTest`

Passo 5: Adicionar os Pacotes NuGet para Selenium e ChromeDriver

1. No terminal, adicione o pacote Selenium WebDriver executando:
`dotnet add package Selenium.WebDriver`
2. Adicione também o pacote Selenium WebDriver ChromeDriver:
`dotnet add package Selenium.WebDriver.ChromeDriver`
3. Adicione o pacote xUnit.runner.visualstudio para executar testes e gerar relatórios:
`dotnet add package xunit.runner.visualstudio`

4. Adicione o pacote Microsoft.TestPlatform para relatórios avançados:
dotnet add package Microsoft.TestPlatform

Passo 6: Escrever o Teste Selenium com xUnit

1. Abra o arquivo UnitTest1.cs no diretório SeleniumReportXunitTest no VS Code.
2. Substitua o conteúdo com o seguinte código de exemplo:

```
using System;
using Xunit;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;

public class SeleniumTests : IDisposable
{
    private readonly IWebDriver _driver;

    public SeleniumTests()
    {
        // Configurar o ChromeDriver
        _driver = new ChromeDriver();
    }

    [Fact]
    public void GoogleSearchTest()
    {
        // Navegar para o site do Google
        _driver.Navigate().GoToUrl("https://www.google.com");

        // Encontrar a caixa de pesquisa pelo atributo name
        IWebElement searchBox = _driver.FindElement(By.Name("q"));

        // Realizar a pesquisa
        searchBox.SendKeys("Selenium WebDriver");
        searchBox.SendKeys(Keys.Enter);

        // Esperar um pouco para ver os resultados
        System.Threading.Thread.Sleep(3000);

        // Verificar se o título da página contém "Selenium WebDriver"
        Assert.Contains("Selenium WebDriver", _driver.Title);
    }

    public void Dispose()
    {

```

```
// Fechar o navegador
_driver.Quit();
_driver.Dispose();
}
}
```

Passo 7: Executar o Teste e Gerar Relatórios

1. Certifique-se de ter o Google Chrome instalado no seu sistema, pois o ChromeDriver o usará para executar o teste.
2. No terminal, execute o seguinte comando para restaurar os pacotes e compilar o projeto:
dotnet restore
dotnet build
3. Execute o teste e gere o arquivo de resultados TRX:
dotnet test --logger "trx;LogFileName=test_results.trx"

Dicas Adicionais

Explore mais métodos e funcionalidades do Selenium na documentação oficial do Selenium:
<https://www.selenium.dev/documentation/en/>.

Certifique-se de que a versão do ChromeDriver corresponde à versão do Google Chrome instalada no seu sistema.