

Automação Web *Descomplicada:*

Selenium + Excel

VBA



Programando em *VBA com Selenium*



Automação de Tarefas com VBA e Selenium no Excel



Se você já trabalha com Excel, provavelmente sabe que muitas tarefas repetitivas podem ser automatizadas para economizar tempo e esforço. Quando combinamos o poder do VBA (Visual Basic for Applications) com o Selenium, uma poderosa ferramenta de automação de navegadores, podemos realizar tarefas como preencher formulários web, extrair dados da web e até interagir com sites diretamente do Excel. Neste e-book, exploraremos algumas funcionalidades essenciais dessa automação com exemplos práticos.

01

CONFIGURAÇÃO INICIAL: INTEGRANDO VBA COM SELENIUMS

Configuração Inicial: Integrando VBA com Selenium

Para começar a usar o Selenium no Excel, você precisa de uma configuração inicial:

1. Baixe o Selenium WebDriver e o "SeleniumBasic", que é uma versão para VBA.
2. Adicione a referência do Selenium ao seu projeto VBA. Isso pode ser feito indo em **Ferramentas > Referências** no editor VBA e marcando "Selenium Type Library".

Agora, vamos escrever o código básico para inicializar um navegador.

```
Sub IniciarNavegador()  
    Dim driver As New Selenium.WebDriver  
    driver.Start "chrome"  
    driver.Get "https://www.exemplo.com"  
End Sub
```

02

PREENCHENDO FORMULÁRIOS WEB

Preenchendo Formulários Web

Uma das utilidades mais comuns do Selenium é preencher formulários automaticamente. Suponha que você precise preencher um formulário de login em um site. Aqui está um exemplo:

```
Sub PreencherFormulario()  
    Dim driver As New Selenium.WebDriver  
    driver.Start "chrome"  
    driver.Get "https://www.exemplo.com/login"  
  
    driver.FindElementById("usuario").SendKeys "seu_usuario"  
    driver.FindElementById("senha").SendKeys "sua_senha"  
    driver.FindElementById("btnLogin").Click  
End Sub
```

Neste exemplo, o código preenche os campos de usuário e senha, e clica no botão de login.

03

EXTRAINDO DADOS DA WEB PARA O EXCEL

Extraindo Dados da Web para o Excel

Outro uso interessante do Selenium é para extrair dados de páginas web e importá-los para o Excel. Imagine que você precisa coletar a cotação de uma moeda diretamente de um site. Veja como fazer isso:

```
Sub ExtrairCotacao()  
    Dim driver As New Selenium.WebDriver  
    Dim cotacao As String  
  
    driver.Start "chrome"  
    driver.Get "https://www.exemplo.com/cotacoes"  
  
    cotacao = driver.FindElementByXPath("//span[@class='valor']").Text  
  
    ThisWorkbook.Sheets("Planilha1").Range("A1").Value = cotacao  
End Sub
```

Aqui, o código acessa uma página de cotações e extrai o valor de uma cotação usando XPath, armazenando o resultado na célula A1 de uma planilha.

04

NAVEGANDO ENTRE PÁGINAS

Navegando entre Páginas

Selenium também permite que você navegue entre diferentes páginas de um site. Isso é útil, por exemplo, para acessar várias páginas de resultados de pesquisa.

```
Sub NavegarEntrePaginas()  
    Dim driver As New Selenium.WebDriver  
  
    driver.Start "chrome"  
    driver.Get "https://www.exemplo.com/busca"  
  
    ' Navega para a próxima página de resultados  
    driver.FindElementByLinkText("Próxima").Click  
End Sub
```

Este exemplo acessa uma página de busca e clica no link "Próxima" para carregar os resultados da próxima página.

05

CAPTURANDO CAPTCHAS E INTERAÇÕES COMPLEXAS

Capturando Captchas e Interações Complexas

Embora o Selenium seja ótimo para automatizar muitas interações web, ele pode ter dificuldades com captchas. Contudo, em sites sem proteções avançadas, é possível automatizar quase qualquer interação.

```
Sub InteragirComPopup()  
    Dim driver As New Selenium.WebDriver  
  
    driver.Start "chrome"  
    driver.Get "https://www.exemplo.com/popup"  
  
    ' Espera o popup carregar  
    driver.Wait 5000  
  
    ' Interage com o popup  
    driver.FindElementById("fecharPopup").Click  
End Sub
```

Esse código fecha um popup que aparece ao acessar a página.

06

FECHANDO O NAVEGADOR APÓS A AUTOMAÇÃO

Fechando o Navegador Após a Automação

Ao terminar a automação, é importante fechar o navegador corretamente para liberar recursos.

```
Sub FecharNavegador()  
    Dim driver As New Selenium.WebDriver  
    driver.Start "chrome"  
    driver.Get "https://www.exemplo.com"  
  
    ' Executa ações e depois fecha o navegador  
    driver.Quit  
End Sub
```

O comando `driver.Quit` fecha o navegador após a execução das ações automatizadas..

07

CONCLUSÃO



Conclusão



O VBA combinado com o Selenium oferece um enorme potencial para automatizar tarefas diretamente do Excel. Seja preenchendo formulários, extraindo dados de sites ou navegando entre páginas, essa integração permite realizar atividades repetitivas de maneira simples e eficaz. Com esses exemplos, você já pode começar a explorar as possibilidades e otimizar seu trabalho com Excel e automação web.

Agradecimentos

Obrigado por ler até aqui!

Este Ebook foi gerado por IA e diagramado por humano.

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.



[https://github.com/IslanderBH/E-book Projeto DIO](https://github.com/IslanderBH/E-book_Projeto_DIO)