CENTRO UNIVERSITÁRIO UNIFECAF

ANÁLISE E DESENVOLVIMENTO EM SISTEMAS

NICOLAS MACHADO NASCIMENTO

PORTFÓLIO PYTHON DEVELOPMENT

Aplicativo de Previsão do Tempo com Selenium e Tkinter

Taboão da Serra, SP 2024

Documentação do Código Python: Aplicativo de Previsão do Tempo com Selenium e Tkinter

Este script implementa uma aplicação desktop com interface gráfica usando Tkinter. Ele permite ao usuário capturar informações meteorológicas (temperatura, umidade e horário) da busca do Google e salvá-las em um arquivo CSV. A automação da busca e captura dos dados é feita utilizando Selenium WebDriver.

Estrutura do Código

Bibliotecas Utilizadas

```
import selenium
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import csv
from tkinter import *
from tkinter import messagebox
```

Essas bibliotecas são utilizadas para:

- Automação web (Selenium);
- Escrita de arquivos (CSV);
- Criação da interface gráfica (Tkinter).

Classe aplicacao

```
10    #Criando interface gráfica com tkinter
11    class aplicacao:
12        def __init__(self):
13            self.layout = Tk()
14            self.layout.title("Previsão do tempo")
15            self.layout.geometry("400x150")
16            self.tela = Frame(self.layout)
17
18            self.login = Label(self.tela, text="Clique para realizar a Previsao do tempo:")
19            self.login.place(x=10, y=10)
20            self.botao = Button(self.tela, text="Buscar previsão", command=self.buscar_previsao)
21
22            self.tela.pack()
23            self.login.pack()
24            self.botao.pack()
25
26            mainloop()
27
```

Responsável por montar a interface gráfica e integrar a funcionalidade de busca de dados.

Método __init__(self)

Cria a janela principal da aplicação com:

- Um *Label* com instruções;
- Um *Button* que dispara a função de busca de dados meteorológicos.

Método buscar_previsao(self)

Automatiza o navegador para:

1. Acessar o site de busca do Google com a palavra-chave "clima"
2. Capturar os dados de com os seguintes *ID da página*:
    - Temperatura (wob_tm);
    - Umidade (wob_hm);
    - Horário (wob_dts).
3. Salvar esses dados em um arquivo CSV;
4. Exibir mensagens de sucesso ou erro ao usuário.

Tratamento de Erros

- Utiliza *try/except* para lidar com falhas na captura de elementos HTML e exibir mensagens via *messagebox no Tkinter.*
- O navegador é encerrado corretamente com driver.quit() tanto em caso de sucesso quanto erro

Execução do Programa

tl = aplicacao ()

Instancia a *classe* aplicacao e inicia o *loop principal* do Tkinter.

Conclusão

Este projeto demonstra de forma clara como é possível unir diferentes bibliotecas do Python para resolver um problema do mundo real. Ao integrar automação web (Selenium), interface gráfica (Tkinter) e manipulação de arquivos (CSV), o código proporciona uma experiência interativa ao usuário final.