

Relatório: Automação de Preenchimento de Formulário com Python

1. Introdução

Este relatório descreve o desenvolvimento de um script em Python que automatiza o preenchimento de um formulário online. O objetivo é demonstrar como a automação pode ser usada para realizar tarefas repetitivas de forma eficiente, economizando tempo e reduzindo erros.

O desafio consistia em preencher um formulário na página <https://www.rpachallenge.com/> com dados de uma planilha Excel. O formulário muda a posição dos campos após cada envio, o que exigiu que o código fosse capaz de identificar dinamicamente os campos corretos.

2. Ferramentas Utilizadas

Para desenvolver a solução, utilizamos as seguintes ferramentas e tecnologias:

- **Python:** Linguagem de programação usada para escrever o script.
 - **Selenium:** Biblioteca que permite controlar um navegador (como o Microsoft Edge) de forma automatizada.
 - **OpenPyXL:** Biblioteca para ler dados de uma planilha Excel.
 - **Microsoft Edge:** Navegador utilizado para interagir com o formulário online.
 - **GitHub:** Plataforma usada para armazenar e compartilhar o código.
-

3. Funcionamento do Código

O código desenvolvido realiza as seguintes etapas:

3.1. Preparação do Ambiente

- O script abre o navegador Microsoft Edge e acessa o site do desafio.
- Ele clica no botão "Start" para iniciar o desafio.

3.2. Leitura dos Dados

- O script lê os dados de uma planilha Excel (challenge.xlsx), que contém informações como nome, sobrenome, empresa, cargo, endereço, e-mail e telefone.

- Esses dados são armazenados em uma lista para serem usados no preenchimento do formulário.

3.3. Preenchimento do Formulário

- Para cada linha da planilha, o script preenche os campos do formulário com os dados correspondentes.
- Como a posição dos campos muda após cada envio, o código identifica dinamicamente os campos corretos usando seletores CSS.

3.4. Envio do Formulário

- Após preencher todos os campos, o script clica no botão "Submit" para enviar o formulário.
- O processo se repete para todas as linhas da planilha.

3.5. Finalização

- Após o término do preenchimento, o script exibe uma mensagem de sucesso e mantém o navegador aberto para visualização dos resultados.
-

4. Benefícios da Automação

A automação traz diversas vantagens para tarefas repetitivas, como:

- **Economia de tempo:** O processo que levaria minutos ou horas para ser feito manualmente é concluído em segundos.
 - **Redução de erros:** A automação elimina erros humanos, como digitação incorreta ou esquecimento de campos.
 - **Escalabilidade:** O mesmo código pode ser adaptado para preencher formulários maiores ou mais complexos.
-

5. Exemplo de Aplicação

Imagine que uma empresa precise preencher diariamente um formulário online com dados de clientes. Com o script desenvolvido, essa tarefa pode ser automatizada, liberando os funcionários para atividades mais estratégicas.

6. Conclusão

O código desenvolvido demonstra como a automação pode ser aplicada para resolver problemas reais de forma eficiente. Utilizando ferramentas como Python e Selenium, foi possível criar uma solução que preenche formulários online de forma rápida e precisa.

Essa abordagem pode ser adaptada para diversas outras tarefas, como coleta de dados, testes de software e integração entre sistemas.

7. Próximos Passos

Para aprimorar a solução, podemos:

- Adicionar tratamento de erros para lidar com situações inesperadas, como falhas de conexão ou mudanças no layout do formulário.
 - Integrar o script com outras ferramentas, como bancos de dados ou sistemas de gestão.
 - Expandir a automação para outras tarefas, como envio de e-mails ou geração de relatórios.
-

8. Agradecimentos

Agradecemos a todos pelo apoio e pela oportunidade de apresentar essa solução. Esperamos que este relatório tenha sido claro e inspirador para explorar as possibilidades da automação.