



Selenium

Teste automatizado para JavaScript



Funcionamento

- O Selenium simula a interação que o usuário faria em um website, realizando testes de funcionalidades e de compatibilidade entre o browser e diferentes plataformas. É possível criar scripts de testes adicionando uma extensão.
- Roda em qualquer navegador baseado na web e suporta várias linguagens de programação, como C#, JavaScript, Python, entre outras.

Uso do Selenium

O ecossistema de Selenium pode ser usado por meio dos seguintes recursos:

- **IDE:** o usuário pode criar testes de forma rápida. Ele é um ambiente integrado de desenvolvimento para scripts de testes automatizados.
- **WebDriver:** ele permite a interação entre o script de teste e o browser que está sendo utilizado. Ele utiliza o próprio drive do navegador para a automação. Geralmente, é aplicado em testes mais complexos.
- **Grid:** o Grid é recomendado para clusterização, assim, você pode realizar os testes em várias máquinas remotamente. Desse modo, você ganha em escala, pois ele pode ser usado por várias máquinas ao mesmo tempo.

Aplicação no desenvolvimento de testes

- Uma das formas mais comuns é usar o Selenium WebDriver, que é uma API que permite interagir com o navegador através de código em uma linguagem de programação, como Python, Java ou Ruby.
- Com o Selenium WebDriver, é possível escrever testes automatizados para verificar a funcionalidade do software, a integração com outros sistemas, a compatibilidade com diferentes navegadores e dispositivos, entre outros aspectos.
- Por exemplo, um teste automatizado pode simular a interação de um usuário com um formulário web, preenchendo os campos necessários e verificando se as respostas estão sendo processadas corretamente.

Selenium WebDriver



Vantagens

- Redução de erros de teste pelo testador;
- Melhor utilização do tempo para o desenvolvedor/testador;
- Menor regressão por meio das características informadas nos testes, já que ele executa sempre os mesmos passos descritivos sequencialmente;
- Feedback rápido para desenvolvedores;
- Suporte para metodologias ágeis e extremas de desenvolvimento;
- Relatório de defeitos personalizado.

Desvantagens

- Só pode ser usado para testar aplicações web;
- Os usuários precisam contar com a ajuda da comunidade, pois não há serviço de suporte;
- Testar imagens é mais complicado, pois é preciso integrar o Selenium a outra ferramenta, que é o Sikuli;
- Para relatório nativo, é preciso integrar o Selenium com o TestNG ou JUnit.

Criação de testes

- A Selenium-IDE fornece uma interface para configuração dos testes e sua execução. **É possível criar os testes de forma manual ou então utilizar o recurso de gravação de testes.**
- Esse recurso permite gravar toda a interação realizada com a página a ser testada de forma automatizada.
- Geralmente, os testes de estresse são feitos executando alguns testes escritos com Selenium simulando diferentes usuários utilizando uma função específica.

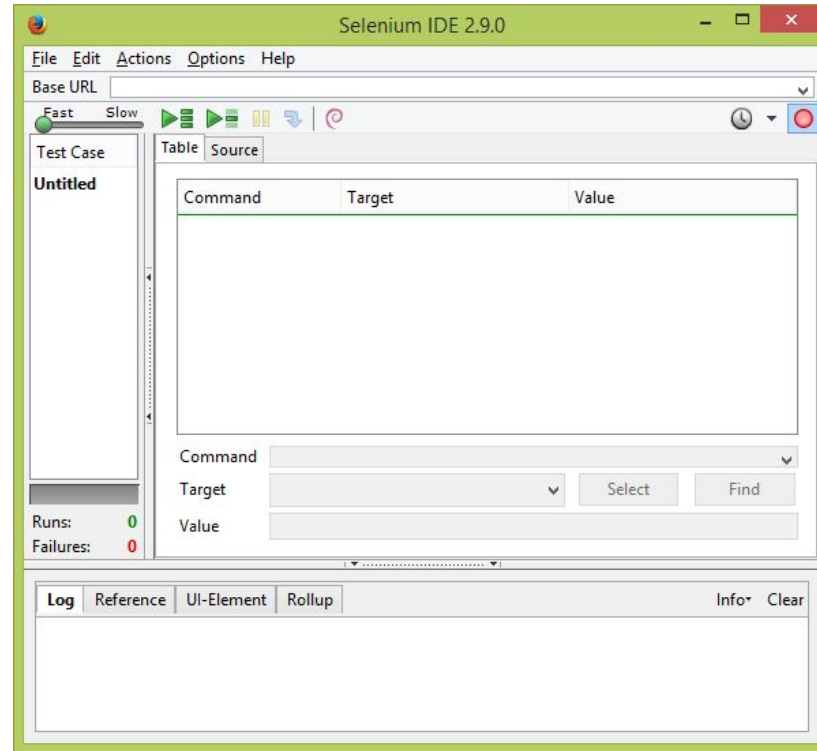
Tipos de testes criados

- Teste de aceitação.
- Teste de performance/desempenho.
- Teste regressivo.
- Desenvolvimento orientado a testes (TDD).
- Desenvolvimento orientado a comportamento (BDD).

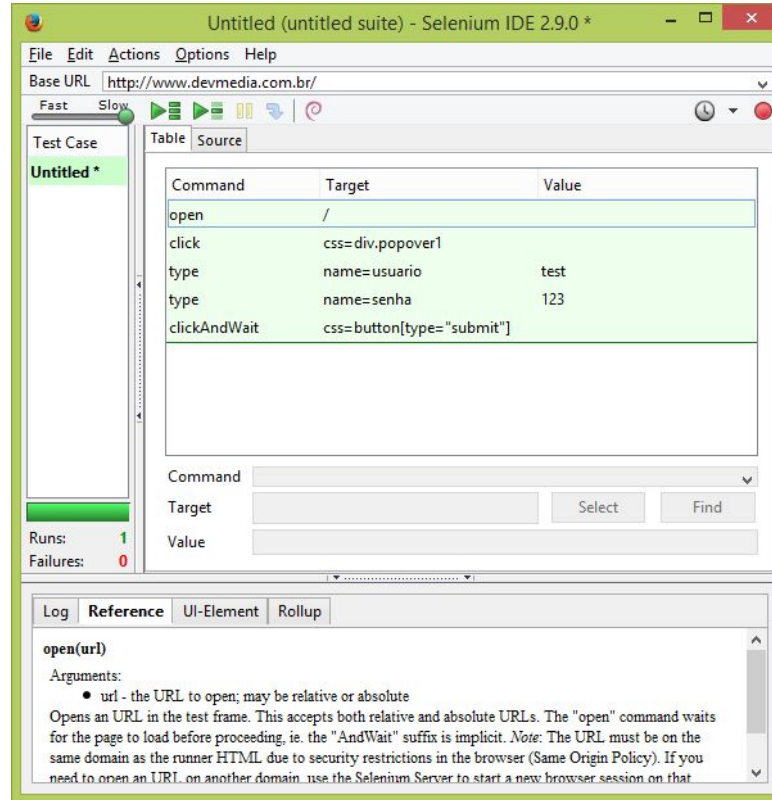
Ferramentas de controle de versão

- Não é possível integrar testes com o SELENIUM IDE no Github. Já o Selenium Webdriver pode ser integrado à API do GitHub.
- A diferença entre o Selenium IDE e o Selenium Webdriver é que você terá que programar o código em alguma linguagem.

Exemplo



Exemplo



Untitled (untitled suite) - Selenium IDE 2.9.0 *

File Edit Actions Options Help

Base URL

Fast Slow

Test Case

Untitled *

Runs: 1

Failures: 0

Command	Target	Value
open	/	
click	css=div.popover1	
type	name=usuario	test
type	name=senha	123
clickAndWait	css=button[type="submit"]	

Command

Target

Value

Log Reference UI-Element Rollup

open(url)

Arguments:

- url - the URL to open; may be relative or absolute

Opens an URL in the test frame. This accepts both relative and absolute URLs. The "open" command waits for the page to load before proceeding, ie. the "AndWait" suffix is implicit. *Note:* The URL must be on the same domain as the runner HTML due to security restrictions in the browser (Same Origin Policy). If you need to open an URL on another domain, use the Selenium Server to start a new browser session on that