

Documento de Mapeamento de Elementos — Challenging DOM - Autobots

1. Informações Gerais da Página

Item	Valor
URL	https://the-internet.herokuapp.com/challenging_dom
Título visível da página	<i>Challenging DOM</i>
Objetivo da página	Demonstrar desafios de automação: tabela sem IDs/descriptivos, botões com seletores frágeis, canvases, botões “dinâmicos” no topo

2. Principais Elementos a Serem Testados / Mapeados

Elemento	Seletor Recomendado	Ações Possíveis / Verificações	Observações sobre fragilidade
Cabeçalho principal	XPath: <code>//h3[text()='Challenging DOM']</code> CSS: <code>h3</code> (se único)	Verificar presença e visibilidade; texto exato	Relativamente estável; pode mudar se traduzirem ou alterarem o título
Parágrafo descritivo	XPath: <code>//p[contains(text(), 'The hardest part in automated web testing')]</code>	Presença; texto contém frase-base; não vazio	Texto longo; cuidado se reescreverem
Botões do topo (qux , baz , foo)	XPath: <code>//a[normalize-</code>	Presença; clique funciona; verificar	Textos podem mudar; prefira classes estáveis

	<code>space(text())= 'qux']</code> CSS: <code>.button-class</code> (se existir)	estilo/efeito visual	se existirem
Canvas	XPath: <code>//canvas</code> CSS: <code>canvas</code>	Verificar existência; visível; largura/altura > 0	Difícil testar conteúdo visual; mudanças na renderização podem quebrar teste
Tabela inteira	XPath: <code>//table</code> CSS: <code>table</code>	Verificar presença; cabeçalho, corpo, número de linhas	Mudanças na tabela podem quebrar contagens fixas
Cabeçalho da tabela (th)	XPath: <code>//table//thead//th</code> CSS: <code>table thead th</code>	Número de colunas; textos esperados; sequência correta	Se colunas mudarem de nome ou ordem, seletores baseados em texto podem quebrar
Linhas da tabela (tr)	XPath: <code>//table//tbody//tr</code> CSS: <code>table tbody tr</code>	Número de linhas; verificar células; presença de botões edit/delete	Evitar usar índice fixo; use âncora textual para localizar linhas
Células da tabela (td)	XPath: <code>//tr[td[contains(text(), 'Iuvaret4')]]/td[contains(text(), 'Adipisci4')]</code> CSS: <code>tr:has(td:contains('Iuvaret4'))</code>	Verificar texto correto; não vazio; formato esperado	Seletores por posição são frágeis; prefira referência textual

	<pre>) td:contains('Ad ipisci4')</pre>		
Botão “edit” por linha	<p>XPath:</p> <pre>//tr[td[contains(text(), 'Iuvaret8')]]// a[@class='edit'] (ou text()='edit' se não houver classe)</pre>	Presença; clicável; comportamento esperado	Evitar usar texto fixo se houver classe; pode quebrar se renomearem
Botão “delete” por linha	<p>XPath:</p> <pre>//tr[td[contains(text(), 'Iuvaret7')]]// a[@class='delete']</pre>	Presença; clicável; ação de exclusão ou confirmação	Mesmo cuidado que “edit”
Rodapé / Link “Powered by Elemental Selenium”	<p>XPath:</p> <pre>//a[contains(text(), 'Elemental Selenium')]</pre> <p>CSS:</p> <pre>a[href*='elementalselenium.com']</pre>	Presença; link abre corretamente	Texto ou link podem mudar; não é crítico funcionalmente

Exemplos de XPath e CSS Prontos

Elemento	XPath	CSS
Título “Challenging DOM”	<code>//h3[text()='Challenging DOM']</code>	<code>h3</code>

Botão topo “qux”	//a[normalize-space(text()='qux']	.button-class (se existir)
Canvas	//canvas	canvas
Cabeçalho da tabela	//table//thead//th	table thead th
Linha específica (Iuvaret4)	//tr[td[contains(text(), 'Iuvaret4')]]	tr:has(td:contains('Iuvaret4'))
Célula específica (Adipisci4)	//tr[td[contains(text(), 'Iuvaret4')]]/td[contains(text(), 'Adipisci4')]	tr:has(td:contains('Iuvaret4')) td:contains('Adipisci4')
Botão “edit” linha Iuvaret8	//tr[td[contains(text(), 'Iuvaret8')]]//a[@class='edit']	(sem CSS simples se não houver classe)
Botão “delete” linha Iuvaret7	//tr[td[contains(text(), 'Iuvaret7')]]//a[@class='delete']	(mesmo que edit)