Fernanda Valdevino junho/2020

O que é XPath?

Conceitos Básicos

- XPath → XML Path Language
- Linguagem de consulta (Query Language) para selecionar nós em um documento XML

XPath - Absoluto ou Relativo?

Conceitos Básicos

XPath Absoluto vs XPath Relativo

 XPath Absoluto → usa o caminho completo, do elemento raiz ao elemento desejado. Sempre começa com "/html".

Ex:

/html/body/div[3]/main/div[1]/div[1]/div[4]/div/div[1]/div[2]/div[1]/divv/section/ol/li[1]/div/article/div[1]/div[2]/div (Trivago)

 XPath Relativo → pode-se referenciar um elemento diretamente através de algum atributo, e seguir o mapeamento a partir dele.

Ex: //*[@id="13750756"]/div/article/div[1]/div[2]/div [Trivago]

- Vantagens:
 - Caminho direto para um componente, identificando-o rapidamente
- Desvantagens:
 - Fragilidade
 - Manutenção frequente e onerosa (tempo)

Comandos Básicos

- nodename → Seleciona todos os nós filhos do nome determinado no contexto.
- √. → Seleciona o nó atual.
- → Seleciona o pai do nó atual.
- $_{ ilde{}^{-}}$ / $_{ ilde{ ilde{o}}}$ ightarrow Seleciona o atributo.
- ✓ / → Seleciona o nó raiz ou filho do nó atual. Representa um caminho absoluto para o elemento.
- // → Seleciona nós em qualquer lugar no documento ou descendentes do nó atual, se precedido pelo nó

Comandos Básicos - Exemplos

Expressão	Resultado
./cd	Seleciona o nó cd atual
/@title	Seleciona o atributo title do nó pai
cd[@title="Great est Hits"]	Seleciona os nós cd que tem o atributo title = "Greatest Hits"
/catalog	Seleciona o nó raiz de catalog
catalog/cd	Seleciona os nós cd que são filhos de todos os nós de catalog
//cd	Seleciona os elementos cd no documento em todo o XML
catalog//cd	Seleciona todos os nós cd que são descendentes dos nós de catalog

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<catalog>
 <cd country="USA">
   <title>Empire Burlesque</title>
   <artist>Bob Dylan</artist>
   <price>10.90</price>
 </cd>
 <cd country="UK">
   <title>Hide your heart</title>
   <artist>Bonnie Tyler</artist>
   <price>9.90</price>
 </cd>
 <cd country="USA">
   <title>Greatest Hits</title>
   <artist>Dolly Parton</artist>
   <price>9.90</price>
 </cd>
</catalog>
```

Selecionando elementos desconhecidos

O asterisco pode ser usado como um caractere coringa:

Expressão	Resultado
catalog/cd/*	Seleciona todos os elementos filhos de todos os elementos cd do elemento catalog
catalog/*/price	Seleciona todos os elementos price que são <u>netos</u> do elemento catalog
/*/*/price	Seleciona todos os elementos price que têm dois ancestrais
/catalog/*	Seleciona todos os nós filhos do elemento raiz catalog
///*/~/~/	Retorna uma sequência de todos os elementos do XML

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<catalog>
 <cd country="USA">
   <title>Empire Burlesque</title>
   <artist>Bob Dylan</artist>
   <price>10.90</price>
 </cd>
 <cd country="UK">
   <title>Hide your heart</title>
   <artist>Bonnie Tyler</artist>
   <price>9.90</price>
 </cd>
 <cd country="USA">
   <title>Greatest Hits</title>
   <artist>Dolly Parton</artist>
   <price>9.90</price>
 </cd>
</catalog>
```

Selecionando seções

Os colchetes se referem a um elemento adiante:

Expressão	Resultado
catalog/cd[1]	Seleciona o primeiro cd filho do elemento catalog
catalog/cd[last[]]	Seleciona o último elemento cd do elemento catalog. Obs: Não existe a função first()
catalog/cd[price]	Seleciona todos os elementos cd que tem um elemento price
catalog/cd[price =10.90]	Seleciona todos os elementos cd que tem um elemento price com valor 10.90
catalog/cd[price =10.90]/price	Seleciona todos os elementos price de todos os elementos cd que tem um elemento price com valor 10.90

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<catalog>
 <cd country="USA">
   <title>Empire Burlesque</title>
   <artist>Bob Dylan</artist>
   <price>10.90</price>
 </cd>
 <cd country="UK">
   <title>Hide your heart</title>
   <artist>Bonnie Tyler</artist>
   <price>9.90</price>
 </cd>
 <cd country="USA">
   <title>Greatest Hits</title>
   <artist>Dolly Parton</artist>
   <price>9.90</price>
 </cd>
</catalog>
```

Selecionando vários caminhos

O pipe pode selecionar vários caminhos:

Expressão	Resultado
/catalog/cd/title /catalog/cd/artist	Seleciona todos os elementos title e artist do elemento cd do elemento catalog
//title //artist	Seleciona todos os elementos title e artist do XML
//title //artist //price	Seleciona todos os elementos title, artist e price do XML
/catalog/cd/title //artist	Seleciona todos os elementos title do elemento cd , e todos os elementos artist do XML

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<catalog>
 <cd country="USA">
   <title>Empire Burlesque</title>
   <artist>Bob Dylan</artist>
   <price>10.90</price>
 </cd>
 <cd country="UK">
   <title>Hide your heart</title>
   <artist>Bonnie Tyler</artist>
   <price>9.90</price>
 </cd>
 <cd country="USA">
   <title>Greatest Hits</title>
   <artist>Dolly Parton</artist>
   <price>9.90</price>
 </cd>
</catalog>
```

Selecionando vários caminhos - Atributos

A arroba é o identificador de um atributo:

Expressão	Resultado
//@country	Seleciona todos os atributos do XML chamados country
//cd[@country]	Seleciona todos os elementos cd que tem um atributo chamado country
//cd[@*]	Seleciona todos os elementos cd que tem algum atributo
//cd[@country=' UK']	Seleciona todos os elementos cd que tem um atributo chamado country com valor " UK "

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<catalog>
 <cd country="USA">
   <title>Empire Burlesque</title>
   <artist>Bob Dylan</artist>
   <price>10.90</price>
 </cd>
 <cd country="UK">
   <title>Hide your heart</title>
   <artist>Bonnie Tyler</artist>
   <price>9.90</price>
 </cd>
 <cd country="USA">
   <title>Greatest Hits</title>
   <artist>Dolly Parton</artist>
   <price>9.90</price>
 </cd>
</catalog>
```

Níveis de Localização

Sintaxe para um nível de localização:

nomedoeixo::noteste[predicado]

- <u>Eixo</u> → relação de árvore entre os nós selecionados pelo nível de localização e o nó atual
- Nó Teste → tipo de nó e nome expandido dos nós selecionados pelo nível de localização
- Predicado → expressões usadas para refinar o conjunto de nós selecionados pelo nível de localização

Ex: child::price[price=9.90] \rightarrow [todos os elementos filhos do nó atual com preço 9.90]

Níveis de Localização - Eixo

<u>Eixo</u> → define um conjunto de nós relativo ao nó atual

Nome do Eixo	Descrição
ancestor	Seleciona todos os ancestrais do nó atual (pais, avôs, etc). Inclui o raiz, se não for ele.
ancestor-or-self	Seleciona o nó atual e o seus ancestrais (pai, avô, etc)
attribute	Seleciona todos os atributos do nó atual
child	Seleciona todos os filhos do nó atual
descendant	Seleciona todos os descendentes do nó atual (filhos, netos, etc). Não contém namespace.
descendant-or-self	Seleciona o nó atual e o seus descendentes (filhos, netos, etc)
following	Seleciona tudo no documento depois da tag de fechamento do nó atual

Níveis de Localização - Eixo

<u>Eixo</u> → define um conjunto de nós relativo ao nó atual

Nome do Eixo	Descrição
following-sibling	Seleciona todos os irmãos depois do nó atual. Obs: Se o nó atual é um nó atributo ou um nó namespace, este eixo vem vazio.
namespace	Seleciona todos os nós namespace do nó atual
parent	Seleciona o pai do nó atual
preceding	Seleciona tudo no documento que está antes da tag de abertura do nó atual
preceding-sibling	Seleciona todos os irmãos ante do nó atual. Obs: Se o nó atual é um nó atributo ou um nó namespace, este eixo vem vazio.
self	Seleciona o nó atual

Níveis de Localização - Nó Teste

Nó Teste → usado para identificar um nó dentro de um eixo

Pode ser executado por nome ou por tipo

Nome do Eixo	Descrição
child::cd	Seleciona todos os elementos cd que são filhos do nó atual. Se o nó atual não tem filhos, retorna vazio
attribute::src	Seleciona o atributo src do nó atual. Se o nó atual não tem atributo src, retorna vazio
child::*	Seleciona todos os elementos filhos do nó atual
attribute::*	Seleciona todos os atributos do nó atual
child::text[]	Seleciona o nó texto filho do nó atual
child::node()	Seleciona todos os filhos do nó atual
descendant::cd	Seleciona todos os elementos cd descendentes do nó atual

Níveis de Localização - Nó Teste

Nó Teste → usado para identificar um nó dentro de um eixo

Pode ser executado por nome ou por tipo

Nome do Eixo	Descrição	
ancestor::cd	Seleciona todos os elementos cd ancestrais do nó atual	
ancestor-or-self::cd	Seleciona todos os elementos cd ancestrais do nó atual. Se o nó atual é um elemento cd , seleciona o nó atual também	
child::*/child::price	Seleciona todos os preços netos do nó atual	
	Seleciona a raiz do documento	

Níveis de Localização - Predicado

<u>Predicado</u> → refina uma consulta dentro de um conjunto de dados já restrito

Escrito dentro de colchetes [[]].

Nome do Eixo	Descrição
child::price[price=9.90]	Seleciona todos os elementos cd que são filhos do nó atual. Se o nó atual não tem filhos, retorna vazio
child::cd[position[]=1]	Seleciona o primeiro cd filho do nó atual
child::cd[position()=last()]	Seleciona o último cd filho do nó atual
child::cd[position()=last()-1]	Seleciona o penúltimo cd filho do nó atual
child::cd[position() < 6]	Seleciona os primeiros 5 cd filhos do nó atual
/descendant::cd[position()=7]	Seleciona o sétimo elemento cd no documento
child::cd[attribute::type="classic"]	Seleciona todos os cd filhos do nó atual que tem um atributo tipo com valor "classic"

Níveis de Localização - Predicado - Operadores Aritméticos

Estes são os operadores aritméticos utilizados no XPath:

Operadores	Descrição
	Dois ou mais conjuntos de nós
\	Soma
	Subtração
	Multiplicação
div	Divisão
mod	Resto da divisão
and	Operador lógico E
or	Operador lógico OU

Operadores	Descrição
-/-	Igualdade
!=	Diferença
/ > /	Maior que
>=	Maior ou igual que
<	Menor que
<=	Menor ou igual que

Funções para conjuntos de nós

Funções utilizadas para operar em conjuntos de nós:

Função	Descrição	Sintaxe
count()	Número de nós num conjunto de nós	number=count(node)
id()	Seleciona elementos pelo id único	node=id(value)
last[]	Posição do último nó do conjunto de nós	number=last()
local-name()	Retorna a parte local de um nó. Um nó consiste em prefixo::nome-local	string=local-name(node)
name[]	Nome do nó	string=name(node)
namespace-url()	namespace URI de um nó específico	uri=namespace-url(node)
position()	Posição de um nó num conjunto de nós	number=position()

Funções de Strings

Funções utilizadas com Strings:

Função	Descrição	Sintaxe
concat[]	Concatena todos os argumentos	string=concat(str1,str2,)
contains[]	Se str2 está contido em str1, retorna true; senão, retorna false	bool=contains(str1,str2)
normalize-space()	Remove espaços antes e depois de uma string e deixa apenas um espaço em branco entre as palavras	string=normalize-space(string)
starts-with()	Se str1 começa com str2, retorna true; senão, retorna false	bool=starts-with(str1,str2)
endswith()	Se str1 termina com str2, retorna true; senão, retorna false	string=ends-with(str1,str2)
string()	Converte o argumento em string	string(value)

⊸ Funções de Strings

Funções utilizadas com Strings:

Função	Descrição	Sintaxe
string-lenght[]	Retorna o nº de caracteres de uma string	number=string-lenght(string)
substring()	Retorna uma parte da string	string=substring(string,star t,lenght)
substring-after()	Retorna uma parte da string que aparece depois do argumento str1	string=substring-after(string,str1)
substring-before()	Retorna uma parte da string que aparece antes do argumento str1	string=substring-before(string,str1)
translate()	Realiza uma substituição caracter por caracter	string=translate(value,str1, str2)

Funções Numéricas

Funções numéricas utilizadas:

Função	Descrição	Sintaxe
ceiling()	Retorna o menor inteiro maior que o valor (arredonda pra cima)	number=ceiling(3.14) → 4
floor[]	Retorna o maior inteiro menor que o valor (arredonda pra baixo)	number=floor(3.14) \rightarrow 3
number[]	Converte o valor em número	number=number('100') → 100
round[]	Arredonda o argumento para o inteiro mais próximo	number=round(3.14) → 3
sum[]	Retorna o valor total de um conjunto de valores numéricos, num conjunto de nós	number=sum(/cd/price) → soma dos preços dos nós cd

Funções Booleanas

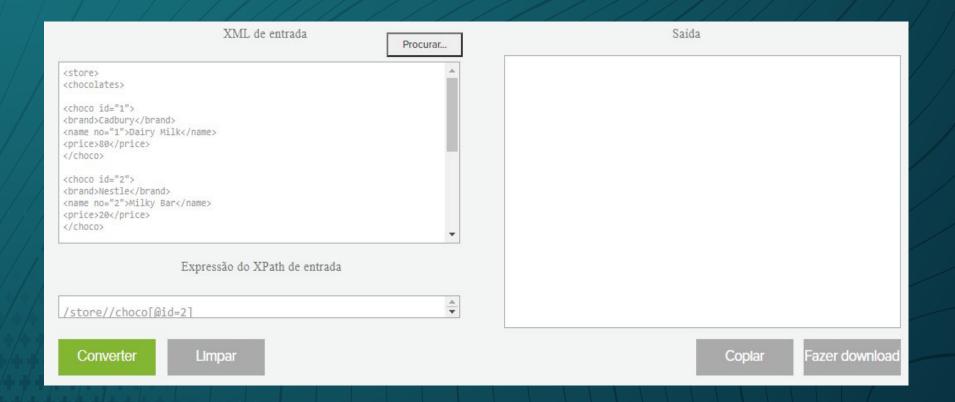
Funções booleanas utilizadas:

Função	Descrição	Sintaxe
boolean()	Converte o argumento em booleano e retorna true ou false	bool=boolean(value)
true[]	Retorna verdadeiro	bool=number(true()) → 1
false()	Retorna falso	bool=number(false()) → O
lang[]	Se o valor do argumento "language" é igual ao idioma do elemento, retorna true; senão, retorna false	bool=lang(language)
not()	Se o argumento "condition" é falso, retorna true; senão, retorna false	bool=not(condition)

Avaliador de XPath

Avaliador de XPath

https://www.site24x7.com/pt/tools/xpath-avaliador.html



Mapeando XPaths - Exemplos

Exemplos Práticos

- '//div[contains(@class, "input-element") and
 descendant::input[contains(@id, "elementId")]]'
- " '//input[contains(@id, "inputId") and
 ancestor::div[contains(@class,"elementClass")]]'
- '//div[descendant::component and contains(@class, "elementClass1")
 and ancestor::div[contains(@class, "elementClass2")]]'
- '//div[contains(@class,"elementClass1") and
 child::div[contains(@class,"elementClass2")]]'
- "'//div[contains(@class, "elementClass1") and descendant::p[@class =
 "elementClass2"]][1]'

Dúvidas



Dúvidas?????





Obrigada!!!