

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha **aquí...**

[Ver cursos que ofrece Autentia](#)

[Descargar comics en PDF y alta resolución](#)



[iNUEVO!] 2008-12-01

2008-11-17

2008-09-01

2008-07-31

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la seccion [comic](#).

Tutorial desarrollado por

Catálogo de servicios de Autentia



Víctor Javier Madrid

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática por la Universidad de Alcalá de Henares.

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(17 MB\)](#)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [teoriaXPath.pdf](#)

Catálogo de servicios Autentia (PDF 6,2MB)



[En formato comic...](#)



☐ Web
☒ www.adictosaltrabajo.com

Últimos tutoriales

2009-03-04
[Persistencia con Spring](#)

2009-02-26
[Vistas materializadas](#)

2009-02-03
[Instalación de MySQL 5.1 en Windows](#)

2009-03-03
[Instalación de Java Virtual Machine](#)

2009-03-03
[Primeros Pasos con Liferay 5.2.1](#)

2009-02-27
[Edición de video MPEG2](#)

2009-02-26
[Introducción teórica a XPath](#)

2009-02-26
[Integración Selenium / Maven 2 / Surefire / Cargo / Tomcat 6](#)

2009-02-24
[Selenium Remote Control](#)

2009-02-22
[Integración de Groovy, JRuby y BeanShell con Spring 2](#)

Últimas ofertas de empleo

2009-02-21
[Otras - Estética/Peluquería - MADRID.](#)

Fecha de creación del tutorial: 2009-02-26

Introducción teórica a XPath.

0. Índice de contenidos.

1. Introducción.
2. Qué es XPath?
3. Terminología en XPath.
4. Tipos de nodo.
5. Expresiones XPath.
6. Localización (Location Path).
7. Pasos de localización.
8. Predicado.
9. Ejes.
10. Funciones.
11. Integración con: Selenium y Consola Firebug.
12. Conclusiones.

1. Introducción

Esta más que demostrada la gran utilidad que tiene los documentos XML, de hecho, desde su aparición estos documentos han ido adquiriendo mayor importancia y han sido aplicados en más sitios. Recordar que en muchos casos sirven de contenedores de propiedades ,ficheros configurables de diversas cosas, etc..Seguro que a lo largo de vuestra vida laboral os habéis tenido que enfrentar con uno de ellos en alguna ocasión.

Antes de meterme de lleno con XPath trararé de dejar claro una serie de conceptos necesarios para aprovechar este tutorial.

XML (eXtensible Markup Language) es un Lenguaje Extensible de Etiquetado cuya función principal es la de generar documentos de texto con etiquetas, estos documentos contienen esencialmente información y hay que destacar que no se hace mención en ningún momento a los detalles de presentación.

Ejemplo de documento XML

```
view plain print ?
01. <libro>
02. <titulo></titulo>
03. <capitulo>
04. <titulo></titulo>
05. <seccion>
06. <titulo></titulo>
07. </seccion>
08. </capitulo>
09. </libro>
```

Por otro lado, estos documentos organizan la información de manera **jerárquica** en función de las etiquetas utilizadas. Dicha información es comprensible de un "simple" vistazo pero hay que tener en cuenta que estos ficheros serán **procesados**.

Por lo tanto el procesamiento de un documento XML requiere acceder a cada una de las partes que lo componen,esto nos va a permitir obtener formas de representación más adecuadas ya que vamos a poder acceder a cada uno de los elementos de forma independiente.

Importante :XML describe los datos pero no se encarga de mostrarlos.

Para poder seleccionar información dentro de un documento XML vamos a hacer uso de XPath, este lenguaje nos va a permitir seleccionar y acceder a cualquier información contenido en un fichero XML.

2. Qué es XPath?

XPath (XML Path Language) es un lenguaje que permite recuperar información de un documento XML. Para ello define una sintaxis para establecer partes en un documento XML, permitiendo navegar a través de sus elementos y atributos, además permite manipular de forma básica booleanos, números y cadenas.

En líneas generales se podría decir que se basa en relaciones de "parentesco" entre esos elementos. Esta parte se verá más en detalle a lo largo del desarrollo de los siguientes apartados.

Utiliza un tipo de notación similar a las rutas de los ficheros, pero haciendo referencia a los nodos de un XML.

XPath fue creado para su uso con el estándar XSTL (es el mayor componente en XSTL).

Estándar XSTL

XSTL (eXtensible Stylesheet Language for Transformations) es un estándar que define la forma de transformar un documento XML en otro documento XML (DTD Origen -> DTD Destino).Además define también la forma de transformar un XML para que su salida sea un documento HTML o texto.

Para realizar la transformación se utilizan "hojas XSTL" (vendrían a ser como hojas de estilo) que se componen de una serie de reglas.El documento XML origen junto con esta serie de reglas son las que utiliza un procesador de XSTL para generar la salida indicada.

XSTL es por tanto un lenguaje de programación para la transformación de documentos XML en "otros" documentos, esta definido en XML y es una recomendación de W3C.

La principal utilidad de XPath dentro de este estándar es la examinar y seleccionar la estructura del documento XML de entrada que se utilizará en la transformación.

Para aquel que quierá ver algún ejemplo práctico integrado con Java se aconseja visitar el siguiente [tutorial](#).

XPath se convirtió en estándar recomendado por la W3C en 1999 (aprobada el mismo día que XSLT).

El objetivo de XPath es la **generación de expresiones** o mejor dicho, expresiones de trayectorias que navegan por un documento XML (recordar que este tipo de documentos disponen de una estructura jerárquica), para ello modela dicho documento como un árbol de nodos.También incorpora una biblioteca con funciones estándar.

Ayuda : Sería algo similar a seleccionar partes de un texto plano mediante el uso de expresiones regulares.

XPath ha servido de base para el desarrollo de múltiples herramientas para el tratamiento de documentos XML.

- 1. **XLink**
- 2. **XPointer**
- 3. **XQL**

1. XLink

XLink (Lenguaje de Enlace XML) es un lenguaje que permite generar enlaces en los ficheros XML, permitiendo establecer relaciones cruzadas (enlaces) entre diferentes elementos (2 o más) sin que estos sepan que estan enlazados.

Hay que diferenciar 2 tipos de enlaces:

- Simples** : Son los enlaces de un recurso local a uno remoto.
- Extendidos** : Son los enlaces que permiten vincular muchos recursos entre sí.

También es una recomendación de W3C.

2. XPointer

XPointer (Lenguaje de Direccionamiento XML) es un lenguaje que permite identificar univocamente partes de un documento XML para establecer vínculos.Lo que hace es establecer un tipo direccionamiento del documento XML en base a su estructura interna (elementos, atributos, etc.). Entre sus aplicaciones estaría el mostrar partes concretas de un documento XML.

También es una recomendación de W3C.

3. XQL

XQL (Lenguaje de consulta XML) es un lenguaje que permite manejar los documentos XML como si fueran base de datos, es decir, "SQL" para los documentos XML.

3. Terminología en XPath.

Ejemplo de documento XML

2009-02-13
[T. Información - Otros no catalogados - MADRID.](#)

2009-02-13
[T. Información - Otros no catalogados - MADRID.](#)

2009-02-13
[T. Información - Otros no catalogados - MADRID.](#)

2009-02-13
[T. Información - Diseñador Gráfico - MADRID.](#)

Anuncios Google

```

view plain print ?
01. <universidad>
02.   <carrera>
03.     <asignatura>
04.       <alumno>Fran</alumno>
05.       <alumno>Juan Pablo</alumno>
06.       <alumno>Marta</alumno>
07.       <alumno>Víctor</alumno>
08.     </asignatura>
09.   </carrera>
10.   <carrera>
11.     <asignatura>
12.       <alumno>Tamara</alumno>
13.     </asignatura>
14.   </carrera>
15.   <carrera>
16.     <asignatura>
17.       <alumno>Fer</alumno>
18.     </asignatura>
19.   </carrera>
20. </universidad>

```

Construcción del Árbol de Nodos

Un árbol de nodos se genera tras procesar un documento XML mediante un parser o analizador.

Ejemplo del árbol de nodos referente al ejemplo del comienzo de este apartado:

```

view plain print ?
01. /
02. |
03. +---universidad
04. |
05.   +---carrera
06.   |
07.   +---asignatura
08.   |
09.   +---alumno
10.   |
11.   +---(texto)Fran
12.   |
13.   +---alumno
14.   |
15.   +---(texto)Juan Pablo
16.   |
17.   +---alumno
18.   |
19.   +---(texto)Marta
20.   |
21.   +---alumno
22.   |
23.   +---(texto)Víctor
24. |
25. +---carrera
26. |
27. +---asignatura
28. |
29. +---alumno
30. |
31. +---(texto)Tamara
32. |
33. +---carrera
34. |
35. +---asignatura
36. |
37. +---alumno
38. |
39. +---(texto)Fer

```

Este árbol de nodos tiene una estructura básica que es común a todos ellos:

- Comienza por un elemento raíz.
- Despliega una serie de elementos intermedios que dependen del elemento raíz.
- Finaliza en varios elementos finales (también denominados nodos hoja).

Nota : El funcionamiento de XPath se basa en este tipo de representación del documento.

Nodo actual (Current)

Es el nodo que está seleccionado cuando se evalúa una expresión XPath, es decir, es el comienzo desde donde se evalúa dicha expresión.

En el ejemplo, si el nodo actual fuera el nodo alumno con texto "Fer" y se buscaran el resto de alumnos, se perderían los que son anteriores.

Nodo contexto (Context)

Son aquellos nodos que son evaluados de forma parcial, para obtener el resultado de la evaluación de la expresión XPath. Hay que tener en cuenta que en cada evaluación de las subexpresiones se obtienen un nuevo conjunto de nodos que pasa a ser el nuevo contexto a evaluar en las siguientes subexpresiones.

Tamaño del contexto

Es el número de nodos que se están evaluando en la expresión XPath en un momento dado.

También se denomina tamaño contextual.

Ojo : Siempre se corresponde con un entero positivo (no nulo).

Posición del contexto

Es la posición que ocupa dentro del documento el nodo contexto.

También se denomina posició contextual.

Ojo :Siempre se corresponde con un entero positivo (no nulo) y es siempre es menor o igual que el tamaño contextual.

4. Tipos de nodo.

Estos son los diferentes tipo de nodos que podemos distinguir:

Nodo raíz (Root)

Es común a cualquier árbol de nodos.

Se reconoce por su identificador "/"

Es muy importante no confundirlo con el elemento raí del documento. Por lo tanto y aplicado a nuestro ejemplo:

- **Nodo raíz :** es el elemento "/" del árbol.
- **Elemento raíz :** es el elemento "universidad".

Se puede observar que el elemento raíz esta contenido en el nodo raíz

Nodo elemento (Element)

Cualquier elemento del documento es un nodo elemento del árbol de nodos.

Características:

- Cada elemento (nodo elemento) tiene un nodo padre.
- El nodo padre de un elemento es también un elemento excepto el padre de elemento raíz que es el nodo raíz.
- Cada nodo elemento tiene hijos que pueden ser otros nodos elemento o nodos hoja.
- Identificador único (Si se acompaña de un DTD que especifique dicho atributo).
- Disponen de propiedades (nombre/atributos/etc.)

En nuestro ejemplo, si seleccionamos el elemento "asignatura"

- **Nodo padre :** "carrera".
- **Nodo/s hijo/s:** los 6 nodos elemento "alumnos".

Nodo atributo (Attribute)

Es un tipo especial de nodo.

Características:

- Un nodo no tienen un número determinado de atributos.
- Atributo: es una etiqueta incorporada al elemento que lo contiene. (Se compone de un nombre y un valor (Formato cadena).
- Los atributos pueden tener valores por defecto si se especifican en el DTD.

Ejemplo de documento XML con atributos para diferenciar las carreras en función del atributo "nombre"

view plain print ?

```
01. <universidad>
02.   <carrera nombre="Ing. Informática">
03.     <asignatura>
04.       <alumno>Fran</alumno>
05.       <alumno>Juan Pablo</alumno>
06.       <alumno>Marta</alumno>
07.       <alumno>Víctor</alumno>
08.     </asignatura>
09.   </carrera>
10.   <carrera nombre="Económicas">
11.     <asignatura>
12.       <alumno>Tamara</alumno>
13.     </asignatura>
14.   </carrera>
15.   <carrera nombre="Ing. Industriales">
16.     <asignatura>
17.       <alumno>Fer</alumno>
18.     </asignatura>
19.   </carrera>
20. </universidad>
```

Nodo texto (Text)

Un nodo texto es un nodo de los considerados como hoja.

Características:

- Referencia a todos los caracteres que no están delimitados por etiquetas.
- No tiene hijos (el texto que lo compone no se condidera hijos suyos).

En nuestro ejemplo, los nodos texto serían los delimitados por la etiqueta "alumno"

- **Nodo texto 1 :**El nodo que contiene el texto "Fran"
- **Nodo texto 2 :**El nodo que contiene el texto "Juan Pablo"
- **Nodo texto 3 :**El nodo que contiene el texto "Marta"
- **Nodo texto 4 :**El nodo que contiene el texto "Víctor"
- **Nodo texto 5 :**El nodo que contiene el texto "Tamara"
- **Nodo texto 6 :**El nodo que contiene el texto "Fer"

Nodo comentario (Comment)

Se puede acceder con la propiedad "string-value".

Nodo tipo instrucción de procesamiento (processing instruction)

Se puede acceder con la propiedad "string-value".

Relaciones entre nodos

- **Padre** : Cada elemento y atributo tiene un padre.
- **Hijo** : Los nodos elemento pueden tener cero, uno o más hijos.
- **Hermanos** : Los nodos que tienen el mismo padre.
- **Ancestros** : Un nodo padre , el padre de ese nodo, etc.
- **Descendientes** : Uno nodo hijo, los nodos hijo de ese nodo, etc.

5. Expresiones XPath.

En XPath una expresión es el elemento base que se utiliza para producir un resultado.

Estableciendo una similitud con cualquier lenguaje de programación, una expresión vendría a ser una instrucción (recordar que XPath es un lenguaje declarativo).

Mediante las expresiones se van a poder **seleccionar** nodos o conjuntos de nodos.

Al evaluar una o varias expresiones se genera un resultado, este resultado podrá ser de uno de los siguientes tipos:

- **Conjunto de nodos (Node-set)** : Un conjunto de nodos (Nota :sin duplicados y sin orden).
- **Cadena** : Un conjunto de caracteres.
- **Número** : Punto flotante.
- **Booleano** : Verdadero / Falso.

Conjunto de nodos (Node-set)

Los node-set son como se ha dicho antes, **conjuntos de nodos** no ordenados que se generan como el resultado de evaluar una expresión XPath. Estos nodos pueden ser de cualquiera de las tipologías anteriormente descritas, pero suelen corresponderse con los tipos : elemento, atributo y texto.

Importante : Los elementos que componen un node-set son siempre **hermanos** con independencia de lo que fueran antes de la ejecución de la expresión XPath.

Sus hijos originales no están incluidos, pero se puede acceder a ellos.

Para poder evaluar una o varias expresiones también se tiene en cuenta el "contexto", es decir, se tienen en cuenta:

- **Nodo contextual.**
- **Posición contextual.**
- **Tamaño contextual**
- **Asignación de variables**
- **Biblioteca de funciones**
- **Declaraciones de espacios de nombres aplicables a la expresión**

Una expresión XPath genera como resultado una lista de referencias (vacía / un nodo / varios nodos) a los elementos que encajan en el patrón buscado.

Importante : NO devuelve la lista con los elementos.

6. Localización (Location Path).

Una ruta de localización (location path) es uno de los tipos de expresiones más importantes que se pueden especificar en XPath, el resultado generado siempre es un node-set.

Importante : Se devuelven los nodos considerados, no los hijos de los nodos.

Su sintaxis es muy similar a la usada para describir la ruta de los ficheros en Linux o Unix. También se pueden indicar que serían similares a las rutas de Windows y MS-DOS sin considerar los 2 excepciones:

- Omitir la unidad de disco. Por ejemplo : "C:".
- Modificar las barras "\" por barras "/".

Hay que hacer notar que sólo se parecen en sintaxis ya que el significado de las expresiones varía mucho de un caso a otro. Mirar los ejemplos:

- Ejemplo de path en Unix:

/usr/home/carrera/assinatura

Explicación : Hace referencia al directorio "assinatura" que se encuentre en "/usr/home/carrera"

- Ejemplo de expresión en XPath:

/universidad/carrera/assinatura

Explicación : Hace referencia a todos los elementos "assinatura" que dependen de cualquier elemento "carrera" que a su vez dependan de cualquier elemento "universidad".

Por lo tanto XPath indica la ruta a varios nodos en base a la estructura del documento XML.

Relación con el nodo contextual

Una ruta de localización (location path) tiene un nodo contextual.

La ruta de localización siempre comenzará a aplicarse desde el nodo contextual a menos que se indique lo contrario estableciendo una ruta explícita.

Tipología de rutas de localización:

- **Absolutas** : Comienzan por "/", lo que significa que se aplicará desde el nodo raíz.
- **Relativas** : Comienzan desde el nodo contextual (no parte del nodo raíz). El nodo contextual cambia con cada "/", que actúa de separador entre los pasos de localización -> En cada paso se selecciona un nuevo node-set que pasa a ser el nuevo nodo contextual.
- **Patrones** : Obligan al uso de los ejes "child" y attribute.

7. Pasos de localización.

Paso de localización es el nombre con el que se denomina a cada cambio que se produce en una ruta de localización. Es decir, a cada cambio de "/" que se produce.

Un paso de localización se compone de:

- **Eje (Axis)** : Es la relación entre el nodo contextual y el paso.
- **Nodo prueba (Node test)** : Es el elemento filtrado (En el caso de rutas de directorios sería el nombre del directorio).
- **Predicado (Predicate)** : Expresión XPath entre corchetes.

El Eje a veces está implícito.

El nodo prueba se suele identificar por el nombre, aunque poniendo "*" simboliza cualquier nombre.

El predicado es opcional.

Por lo tanto la sintaxis de un paso de localización sería:

Eje::NodoPrueba[Predicado]

Selección de nodos

En este apartado se pretende mostrar las diferentes formas (pasos de localización) con las que se pueden seleccionar los nodos.

Sintaxis	Descripción
/	Selecciona desde el nodo raíz.
//	Selecciona nodos desde el nodo contextual (Sin importar donde se encuentren).
.	Selecciona el nodo contextual.
..	Selecciona el padre del nodo contextual.
nombre	Selecciona los nodos hijos de nodo nombrado.
@prueba	Selecciona todos los atributos "prueba".

Estas serían las formas de poder seleccionar nodos que no se conocen.

Sintaxis	Descripción
//*	Selecciona todos los nodos del documento.
node()	Selecciona todos los nodos del documento.
*	Selecciona todos los nodos elemento.
@*	Selecciona todos los nodos atributos.
/<NodoEjemplo>/*	Selecciona todos los los nodos hijos del nodo ejemplo.
//<NodoEjemplo>[@*]	Selecciona todos los atributos del nodo ejemplo .

Operadores

Estos son los operadores que se permiten utilizar en las expresiones XPath.

Operador	Descripción
	Unión entre 2 node-sets
+	Suma
-	Resta
*	Multiplicación
div	División
=	Igualdad
!=	Diferencia
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
or	Disyunción
and	Conjunción
mod	Módulo (resto)

8. Predicados.

Un predicado es una "condición" que permiten seleccionar un nodo con unos determinados atributos o características.Como se ha visto en puntos anteriores incorpora un tipo de verificación durante el paso de localización.

Su sintaxis viene dada entre corchetes ([predicado]) y tiene un valor booleano.

Ejemplo de predicado:

/universidad/carrera/asignatura[@nombre="Base de datos"]/alumno

Hace referencia a todos los alumnos que pertenecen a las asignaturas que tiene como atributo nombre "Base de datos".

En el apartado anterior se dijo que los predicados son opciones y que en caso de existir se representan entre corchetes. Entre los corchetes se pueden colocar

Elemento corchete	Descripción
Número	Selecciona el elemento cuya aparición es la indicada en el número.
Función	Selecciona el elemento cuya aparición es el resultado de ejecutar la función. Por ejemplo: last()

Condición position()	Selecciona la cantidad de elementos seleccionada en la condición . Por ejemplo position()<6 (Selecciona 5 elementos)
Condición	Selecciona los elementos que cumplen la condición.Por ejemplo: dinero<1000
@atributo='XXXX'	Selecciona los elementos cuyo atributo coincide en valor.Por ejemplo: nombre='Esther'

9. Ejes.

Un eje se define como la relación existente entre un paso de localización y su nodo contextual. También se puede definir como un conjunto de nodos relativos al nodo contextual.

Recordar la sintaxis de un paso de localización:

Eje::**NodoPrueba**[Predicado]

Como se puede observar

- El eje y el nodo de prueba se encuentran separados por el elemento "::**"**.
- Se puede traducir por : **NodoPrueba**[Predicado] "que es un " Eje.

Estos son los ejes que se pueden utilizar

Sintaxis	Descripción	Caso Especial	Resultado
ancestor	Selecciona todos los ancestros del nodo contextual.	/ancestor:: node() /parent:: node()	/
ancestor-or-self	Selecciona todos los ancestros del nodo contextual y el mismo.		
attribute	Selecciona todos los atributos del nodo contextual.		
child	Selecciona todos los hijos del nodo contextual.(Suele estar implícito)	child:: nombreNodo	nombreNodo
descendant	Selecciona todos los descendientes del nodo contextual.	/descendant-or-self:: node() /	//
descendant-or-self	Selecciona todos los descendientes del nodo contextual y el mismo.		
following	Selecciona todo el documento que se encuentra a partir del nodo contextual.		
following-sibling	Selecciona todos los hermanos posteriores del nodo contextual.		
namespace	Selecciona todos los nodos con el namespace del nodo contextual.		
parent	Selecciona el nodo padre del nodo contextual.	parent:: node()	..
preceding	Selecciona todo en el documento que esta antes del nodo actual.		
preceding-sibling	Selecciona todos los hermanos anteriores al nodo actual.		
self	Selecciona el nodo actual. (Esta implícito para el nodo raíz)	self:: node()	.

Ejemplo de expresión XPath:

/universidad/carrera

Ejemplo de equivalencia con ejes:

/self::**universidad**/child::**carrera**

10. Funciones.

En XPath esta permitido el uso de funciones que nos van a facilitar poder realizar algunas operaciones necesarias de una manera sencilla.

Podemos encontrar una amplia variedad de funciones como por ejemplo:

- boolean()** : Convierte a booleano. Al aplicarlo sobre un conjunto de nodos devuelve true si esta vacío.
- position()** : Devuelve la posición de un nodo en su contexto
- last()** : Devuelve la última posición
- count** : Devuelve el nº de nodos e un conjunto de nodos.
- not()** : Devuelve el contrario de un booleano dado.

Nota : Existen más funciones de las que aquí se presentan.

11. Integración con: Selenium y Consola Firebug.

Este punto es muy importante porque vamos a relacionar XPath con el paquete de herramientas Selenium y con el plugin Firebug.

Firebug

Si se desconoce esta herramienta se aconseja visitar el siguiente [tutorial](#).

Para aquel que ya sepa un poco de que va Firebug, recordar que podremos ser capaces de extraer la expresión XPath de un elemento inspeccionado. De esta forma podremos utilizarla para identificar los elementos en el paquete Selenium, pero con una condición, las expresiones que se obtiene por defecto no son todo lo claras que deberían de ser.

Ejemplo de expresión por defecto extraída desde Firebug :

/html/body/div[3]/div[5]/div[2]/table/tbody/tr/td/div/div/h1/a

Ejemplo de la anterior expresión adaptada :

//a[@ href="detalle-noticia.php?noticia=96"]

Como se puede comprobar en el ejemplo, el segundo caso es mucho más claro y esta más simplificado. En la mayoría de los casos tendréis que adaptar la expresión a las necesidades.

Importante :Intentar incorporar identificadores en todos los elementos para que sea más sencillos localizarlos con XPath.

Aún así, otra posible aplicación sería la de poder utilizar expresiones XPath en la consola de Firebug, de esta forma seremos capaces de encontrar diferentes elementos de la página web. Para ello se utiliza el siguiente comando:

\$x("expresion XPath")

Por ejemplo :

\$x("//a") Devolvería todos los links dentro de una página

Paquete Selenium

El paquete Selenium es un conjunto de herramientas que nos van a permitir realizar pruebas sobre aplicaciones web. Para más información visitar los siguientes tutoriales:

- [Selenium IDE](#)
- [Selenium Core](#)
- [Selenium RC](#)

XPath es una de las formas que nos va a permitir localizar elementos en un script realizado por Selenium, de hecho, es considerado un sistema de localización de elementos válido. De esta manera, vamos a poder identificar elementos de una forma más "completa" mediante una de sus expresiones.

Una vez realizado un script con la aplicación Selenium IDE, podemos modificar los localizadores de los elementos, para ello utilizaremos Firebug con el cual inspeccionaremos el elemento y realizaremos una expresión a partir de uno de sus atributos que a poder ser sea único (por ejemplo id).Para comprobar la validez de nuestra nueva expresión la sustituiremos por el localizador anterior y pulsaremos el botón FIND en la tabla habilitada por Selenium IDE, en caso de ser correcta nuestra expresión se iluminará mediante un recuadro verde el elemento seleccionado.

12. Conclusiones.

En resumen, XPath es un lenguaje de "ayuda" para localizar elementos dentro de cualquier fichero XML, como véis tiene muchas utilidades pero seguro que vosotros sois capaces de sacarles muchas más. Tenerlo muy en cuenta cuando uséis el paquete Selenium ya que os falicitará en gran medida el trabajo y sobre todo permitirá que no se produzcan errores en los scripts cuando estos se basan en la posición de los elementos.

Ojo: No me voy a cansar de repetir que los elementos HTML que utiliceis usen identificadores únicos, que luego pasa lo que pasa....

Considero que no esta demás saber utilizar XPath porque uno nunca sabe cuando lo va a tener que utilizar ;-).

Un saludo.

Víctor

<mailto:vjmadrid@autentia.com>

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ivota!

- Muy malo Malo Regular Bueno Muy bueno
- ☐ ☐ ☐ ☐ ☐

Votar

- Puedes opinar sobre este tutorial [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- Añadir a favoritos Technorati.  ADD THIS BLOG TO MY **Technorati FAVORITES**



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

[Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Gestión de contenidos

Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subcripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Selenium Core : El motor de Selenium.	Selenium Core es un aplicación perteneciente al juego de herramientas SeleniumHQ que permite realizar juegos de pruebas sobre aplicaciones web.	2009-02-16	437	Muy bueno	4	
Desarrollo de aplicaciones Web con Struts 1	En este tutorial analizaremos una aplicación Web realizada con Struts 1 que cubre los conceptos más importantes de este Framework.	2009-01-14	1632	Muy bueno	8	
Manual Avanzado de Firebug	Firebug es una extensión gratuita para Mozilla Firefox especialmente diseñada para todos aquellos programadores que se dedican al desarrollo web. En este tutorial os enseñaremos en profundidad su funcionamiento.	2009-02-09	733	Muy bueno	6	
Integración Selenium / Maven 2 / Surefire / Cargo / Tomcat 6	Con este tutorial se pretende integrar en nuestro proyecto : Maven, Selenium, Surefire, Cargo y Tomcat 6 con el objetivo de incluir y ejecutar las pruebas de integración dentro del ciclo de vida de Maven.	2009-02-26	175	Muy bueno	3	
Executor : Un programa para ejecutarlos a todos.	Nuestro amigo Víctor nos enseña éste utilísimo programa para programar la ejecución de aplicaciones de manera sencilla y rápida	2009-01-19	868	Muy bueno	6	
Primeros pasos con Selenium IDE	Nuestro amigo y compañero Víctor nos enseña, para acabar bien el año, el uso de una herramienta muy interesante para la realización de pruebas sobre aplicaciones web	2008-12-31	1468	Muy bueno	19	
Manual Básico de FireBug	Este tutorial nos va a enseñar una herramienta llamada FireBug, que es un plugin de Firefox que nos brinda un paquete de utilidades para el desarrollo de páginas y aplicaciones Web	2007-05-18	6096	Bueno	11	
Selenium Remote Control	Selenium Remote Control es una herramienta que permite automatizar las pruebas sobre aplicaciones web	2009-02-24	275	Bueno	3	
Reorganización estratégica	Manuel Lopez nos hace reflexionar sobre algunas cuestiones, conceptos, circunstancias y vivencias, del día a día laboral, empresarial y profesional, que le puedan ayudar a mejorar sus planteamientos y enfoques organizativos, empresariales, y laborales.	2008-12-09	530	Bueno	5	
Introducción a XPath	Este tutorial pretende ser una introducción a XPath mediante unos sencillos ejemplos.	2006-11-02	6743	-	-	

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.