

# Java y xpath

---

1 El XML de ejemplo

2 El código java

3 Expresiones de búsqueda con XPath

3.1 Buscar por nombre de tag

3.2 Buscando por atributos

---

XPath es una sintaxis que permite buscar rápidamente datos y nodos dentro de un fichero XML. Java, en su paquete `javax.xml.xpath` tiene clases que implementan XPath, por lo que en java disponemos de una forma fácil de buscar datos y nodos dentro de un fichero XML. Veamos aquí algunos ejemplos

# El XML de ejemplo

Supongamos el siguiente fichero XML de ejemplo que usaremos para buscar cosas en él usando Java y Xpath.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<espacio>
```

```
  <galaxia nombre="Via Lactea">
```

```
    <estrella nombre="Sol">
```

```
      <planeta nombre="Mercurio" />
```

```
      <planeta nombre="Venus" />
```

```
      <planeta nombre="Tierra">
```

```
        <satelite nombre="Luna" />
```

```
      </planeta>
```

```
      <planeta nombre="Marte" />
```

```
<planeta nombre="Jupiter">  
    <satelite nombre="Io" />  
    <satelite nombre="Europa" />  
    <satelite nombre="Ganimedes" />  
    <satelite nombre="Calisto" />  
</planeta>
```

```
<planeta nombre="Saturno">  
    <satelite nombre="Titan" />  
    <satelite nombre="Mimas" />  
    <satelite nombre="Encelado" />  
    <satelite nombre="Tetis" />  
    <satelite nombre="Dione" />  
    <satelite nombre="Rea" />  
    <satelite nombre="Hiperion" />
```

```
        <satelite nombre="Japeto" />
    </planeta>
    <planeta nombre="Urano" />
    <planeta nombre="Neptuno" />
    <!-- Lo siento, soy de la vieja escuela -->
    <planeta nombre="Pluton" />
</estrella>
<sistema_estelar nombre="Alfa Centauri">
    <estrella nombre="Alfa Centauri A"></estrella>
    <estrella nombre="Alfa Centauri B">
        <planeta nombre="Alfa Centauri Bb"></planeta>
    </estrella>
    <estrella nombre="Proxima Centauri"></estrella>
</sistema_estelar>
```

</galaxia>

<galaxia nombre="Enana del Can Mayor" />

<galaxia nombre="Enana de Sagitario" />

<galaxia nombre="Gran Nube de Magallanes" />

<galaxia nombre="Enana del Can Mayor" />

</espacio>

# El código java

El código java para buscar algo ahí dentro puede ser como este

```
package com.chuidiang.ejemplos.xpath;
```

```
import java.io.File;
```

```
import javax.xml.parsers.DocumentBuilder;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import javax.xml.xpath.XPath;
```

```
import javax.xml.xpath.XPathConstants;
```

```
import javax.xml.xpath.XPathFactory;
```

```
import org.w3c.dom.Document;
```

```
import org.w3c.dom.NodeList;
```

```
public class PruebaXPath {
```

```
    public static void main(String[] args) throws Exception {
```

```
        // La expresion xpath de busqueda
```

```
        String xpathExpression = "//satelite[@nombre='Luna']";
```

```
        // Carga del documento xml
```

```
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
```

```
        DocumentBuilder builder = factory.newDocumentBuilder();
```

```
        Document documento = builder.parse(new File("prueba.xml"));
```

```
        // Preparación de xpath
```



```
XPath xpath = XPathFactory.newInstance().newXPath();
```

```
// Consultas
```

```
NodeList nodos = (NodeList) xpath.evaluate(xpathExpression, documento, XPathConstants.NODESET);
```

```
for (int i=0;i<nodos.getLength();i++){
```

```
    System.out.println(nodos.item(i).getNodeName()+" : " +
```

```
    nodos.item(i).getAttributes().getNamedItem("nombre"));
```

```
}
```

```
}
```

```
}
```

# EJECUCIÓN DEL PROGRAMA

En la variable `xPathExpression` hemos puesto una expresión típica de XPath. No te preocupes por su contenido, puesto que es lo que vamos a ver a lo largo de este tutorial. De alguna forma, ese contenido es el que indica qué queremos buscar, en este caso, satélites que se llamen "Luna".

Bajo el comentario "Carga del documento", abrimos el documento XML que hemos llamado `prueba.xml` y cuyo contenido es el que hemos mostrado más arriba. En la variable `documento` nos queda cargado dicho fichero.

Creamos una instancia de XPath con `XpathFactory.newInstance().newXPath()`. Esta instancia será la que usemos para hacer las búsquedas dentro del documento que hemos cargado.

`xpath.evaluate()` será el método al que hay que llamar para hacer la búsqueda. Pasamos la expresión de búsqueda (la variable `xPathExpression` anterior), el documento recién cargado y una constante que le indique al método `evaluate()` qué tipo de cosa estamos buscando. Con `XPathConstants.NODESET` le estamos indicando que queremos que nos devuelva una lista de nodos, por eso podemos hacer el cast a `NodeList` del resultado.

Después de ejecutar `xpath.evaluate()` y obtener algún resultado, sólo nos queda escribir los resultados.

### Expresiones de búsqueda con XPath

Veamos cuales son los posibles contenidos de esa variable `xPathExpression`. Hay toda una sintaxis para esas expresiones, veremos aquí algunos ejemplos.

#### Buscar por nombre de tag

Para buscar nodos con un nombre de tag concreto, debemos construir la cadena para llegar hasta ese tipo de nodos usando `/` como separador. Por ejemplo, si ponemos

`/espacio/galaxia/estrella/planeta`

nos saldrán todos los planetas de nuestro sistema solar. No saldrán los planetas extrasolares, como Alfa Centauri Bb, puesto que cuelga de `/espacio/galaxia/sistema_estelar/estrella/planeta`.

Si no nos importa de dónde cuelguen, sino que queremos cualquier planeta, podemos usar // (doble barra) como un comodín para indicar cualquier camino. Por ejemplo

//planeta

nos daría todos los planetas, independientemente de en qué parte del fichero XML se encuentren.

Podemos usar la // en cualquier parte del camino, por ejemplo

/espacio/galaxia//planeta

nos daría todos los planetas que estén por debajo de /espacio/galaxia, a cualquier nivel de profundidad e independientemente del camino para llegar a ellos, siempre que el camino empiece por /espacio/galaxia

Si no sabemos el tipo de nodo o nos da igual cual sea su nombre, podemos usar \*. Por ejemplo

/espacio/galaxia/\*

nos daría todos los nodos que cuelguen de una galaxia que a su vez cuelgue de un espacio, es decir, nos daría las estrellas y sistemas estelares que cuelguen de una galaxia. No nos daría, por ejemplo, las estrellas Alfa Centauri, puesto que cuelgan de un sistema estelar.

Podemos seleccionar nodos de varios path usando el operado | de esta forma

/espacio/galaxia/estrella | /espacio/galaxia/sistema\_estelar/estrella

que nos daría todas las estrellas que cuelguen de una galaxia o de un sistema estelar.

Buscando por atributos

Podemos buscar nodos que cumplan el tener algún atributo de alguna manera. Por ejemplo,

/espacio[@nombre]

nos dará todos los espacios que tengan un atributo nombre, ninguno en nuestro ejemplo, pero

```
/espacio/galaxia[@nombre]
```

nos dará todas las galaxias que tengan atributo nombre. Podemos poner una condición con los operadores habituales (el == aquí es un simple =), de forma que

```
//satelite[@nombre='Luna']
```

nos dará todos los satelites cuyo nombre sea "Luna" mientras que

```
//*[@nombre!='Luna']
```

nos dará todos los nodos (satelites o no), cuyo nombre no sea "Luna".