

| | |
|-----------------------------------|-----------|
| Breve introducción a XPath | 2 |
| ¿Qué es XPath? | 2 |
| Atributos | 3 |
| Predicados | 3 |
| Operadores | 4 |
| Secuencias | 5 |
| Expresión for | 6 |
| Funciones | 7 |
| Expresión if | 8 |
| XQuery | 8 |
| Expresión FLWOR | 9 |
| Resumen | 11 |

Breve introducción a XPath

[MobileTogether](#) utiliza el lenguaje XPath para la creación de fórmulas y expresiones.

Si sabe escribir fórmulas en Excel, aprenderá fácilmente a escribir fórmulas o expresiones XPath en MobileTogether. Para ello solo es necesario comprender algunos conceptos básicos. El generador de XPath de MobileTogether facilita la construcción de expresiones XPath, al igual que el generador de fórmulas de Excel ayuda a escribir fórmulas en las hojas de cálculo.

¿Qué es XPath?

XPath es un estándar basado en XML que se sirve de expresiones en forma de rutas de acceso para identificar determinados componentes de los datos. Todas las fuentes de datos a las que acceda desde MobileTogether se representan en forma de árbol con estructura jerárquica en la ventana Fuentes de página. Cada estructura jerárquica empieza con un componente raíz (p. ej. \$XML1 o \$DB1) y sigue con los diferentes componentes de datos, llamados elementos. Estos elementos están anidados de forma lógica para que pueda expandir y contraer cada nivel de elementos y ver qué hay en cada uno de ellos. Y lo mejor de MobileTogether es que los datos se presentan siempre de la misma forma, tanto si provienen de una BD, de un documento XML o de un servicio web, etc. El acceso a los datos también se hace siempre de la misma forma.

Lo que XPath permite es navegar por estas estructuras en forma de árbol para asociar sus elementos de datos con objetos de la interfaz del usuario del dispositivo móvil de los usuarios finales. Es muy parecido a la navegación por directorios en el sistema de archivos de un equipo informático. Independientemente del sistema operativo, a los archivos siempre se puede acceder a través de una ruta de acceso. Por ejemplo:

C:\Usuarios\Spock\Documentos\ScienceRecords (Windows)

/Usuarios/Kirk/Documentos/BeautifulAlienWomen (Linux/Mac)

Y aquí puede ver un ejemplo de expresiones XPath para comparar:

\$DB1/salesdata/region/month/gadgets

Como puede ver, se trata del mismo concepto. XPath sirve para navegar por estructuras de datos en forma de árbol al igual que las rutas de acceso sirven para navegar por el sistema de archivos de un equipo informático. XPath utiliza la barra diagonal / para separar los elementos de la ruta de acceso, al igual que en Linux y MacOS ese carácter sirve para separar directorios. Además, al igual que un punto . hace referencia al directorio actual del sistema de archivos, en XPath un punto . hace referencia al elemento actual. Y si el doble punto .. hace referencia a un directorio primario del sistema de archivos, en XPath el doble punto.. hace referencia al elemento primario.

Atributos

Si su fuente de datos es un documento XML o datos XML que devuelve un servicio web, los datos de la estructura en forma de árbol pueden ser de dos tipos: elementos o atributos. Los elementos son los principales componentes de datos en XML y se pueden anidar, es decir, un elemento puede tener elementos secundarios. Los atributos, por el contrario, son valores de datos simples que se anexan a los elementos. En las estructuras de la ventana Fuentes de página los atributos llevan el signo =. El motivo es que en XML los atributos se escriben así: color="verde".

Cuando quiera hacer referencia a un atributo en una expresión XPath, deberá escribir una arroba @ antes del nombre del atributo:

\$XML1/salesdata/region/@name

Aparte de esto, los elementos y los atributos se pueden usar exactamente igual, los puede usar en cálculos o para crear un gráfico a partir de sus valores y los puede asignar a objetos de la interfaz del usuario en un diseño de MobileTogether.

Predicados

A veces no es suficiente con solo especificar la ruta de acceso de un elemento de datos. De lo que se trata es de seleccionar algunos datos que cumplan ciertos criterios para, por ejemplo, verlos en forma de gráfico o presentarlos en forma de tabla.

El equivalente en las bases de datos relacionales es la típica cláusula WHERE de una instrucción SQL. Esta cláusula permite seleccionar los datos de la BD que cumplen ciertos criterios.

El concepto equivalente en XPath se denomina "predicados". Un predicado XPath sirve para especificar los criterios de selección. Los predicados se escriben en forma de prueba lógica y entre corchetes. Por ejemplo:

\$XML1/salesdata/region[@name="East"]/Month

Es decir, en la fuente de datos \$XML1, ir al elemento "salesdata", después ir a sus elementos secundarios "region" y seleccionar solo los elementos secundarios "region" cuyo atributo "name" tenga el valor "East" y después seleccionar el elemento secundario "Month" de ese elemento "region".

Operadores

Al principio decíamos que XPath es un lenguaje de expresiones. Esto no solo significa que permite seleccionar ciertos componentes de datos, sino que además permite realizar operaciones matemáticas con los datos para calcular valores nuevos. En MobileTogether puede usar expresiones XPath para calcular valores nuevos a partir de los datos disponibles, como si fueran fórmulas de Excel.

Por ejemplo, imagine que quiere crear una solución de facturación móvil y quiere ver la cantidad total de un concepto en un formulario móvil:

`$DB1/sales/invoice/item/quantity * $DB1/sales/invoice/item/price`

Si suponemos que la estructura de datos tiene elementos distintos para la cantidad y para el precio (que pueden ser editados por el usuario), entonces podremos calcular la cantidad total de ese concepto realizando una simple multiplicación con un asterisco *.

Una de las ventajas de XPath es que podemos realizar la multiplicación dentro de la misma expresión XPath usando paréntesis. Por ejemplo, esta sería la versión abreviada de la expresión anterior:

`$DB1/sales/invoice/item/(quantity * price)`

Igualmente puede usar los operadores matemáticos + y -. Pero para la división se utiliza un símbolo diferente. Como la barra diagonal ya se usa para separar los elementos de la ruta de acceso, en la división tenemos que usar un operador diferente. El operador "div" realiza divisiones de números reales, mientras que el operador idiv se utiliza para divisiones enteras (es decir, si quiere que el resultado sea un entero en lugar de un número real).

Además se recomienda poner espacios antes y después de los operadores para evitar que se confundan con parte del elemento o del atributo. Sobre todo cuando se trata del operador -.

Secuencias

Las expresiones de rutas de acceso de directorios identifican un solo archivo o directorio, mientras que las expresiones XPath pueden hacer referencia a varios nodos de datos (elementos o atributos). De hecho, todas las expresiones XPath devuelven una secuencia de nodos.

Por ejemplo, la mayoría de las bases de datos contienen filas de datos que tienen una estructura similar, al igual que la mayoría de los documentos XML tienen una serie de elementos que tienen el mismo nombre y la misma estructura. El propósito de XPath es permitir trabajar con ese tipo de conjuntos de datos y así poder ocuparse de un conjunto de nodos entero de una sola vez. Esto tiene muchísimas ventajas, como podrá comprobar a continuación.

Al mismo tiempo, a veces puede ser necesario definir secuencias en una expresión XPath. Esto se hace enumerando los valores de esta forma:

```
(1, 2, 3, 4, 5)
```

Las secuencias siguen un orden y, por tanto, la secuencia anterior no es igual que la secuencia (3, 5, 2, 4, 1). Las secuencias pueden contener nodos de los datos y valores numéricos constantes o valores de cadena constantes.

Por ejemplo, una secuencia de cadenas podría ser:

```
("a", "b", "c", "d", "e")
```

Y, por último, se puede especificar secuencias ordenadas de valores numéricos usando el operador de intervalo "to":

```
(1 to 5)
```

Esto es igual que usar la secuencia (1, 2, 3, 4, 5), pero el operador de intervalo "to" permite construir secuencias de varios valores numéricos de manera mucho más rápida.

Expresión for

Si ha trabajado con lenguajes de programación, sabrá que el bucle "for" es una herramienta frecuente que existe en la mayoría de los lenguajes modernos y que en XPath es un componente clave.

En su forma más sencilla, una expresión "for" sirve para recorrer una secuencia de valores o nodos usando una variable de bucle temporal que se define directamente en la expresión "for". Por ejemplo, la expresión

```
for $i in (1 to 10) return 2 * $i
```

producirá la secuencia numérica (2, 4, 6, 8, 10, 12, 14, 16, 18, 20) recorriendo la secuencia original (1 to 10) usando la variable \$i y calculando 2 * \$i por cada punto de datos de la secuencia.

Ahora que sabemos que cualquier expresión XPath puede devolver una secuencia de nodos, también podemos usar la expresión "for" para recorrer esos nodos y realizar algunas operaciones matemáticas. Por ejemplo, para recorrer los conceptos de una factura y calcular la cantidad correspondiente para ese concepto, podemos usar esta expresión:

```
for $x in $DB1/sales/invoice/item return $x/price * $x/quantity
```

En este caso, \$x será nuestra variable de bucle que recorrerá cada concepto de la factura y esta expresión devolverá una secuencia que contiene las cantidades para cada concepto de la factura. Imaginemos que tenemos estos dos conceptos en la fuente de datos de la factura:

```
<item>
<quantity>5</quantity>
<price>12,50</price>
</item>
<item>
<quantity>3</quantity>
<price>27,85</price>
</item>
```

Entonces la expresión XPath anterior devolverá la secuencia (62.5 83.55) que son las cantidades de cada concepto de la factura.

Funciones

En XPath las funciones funcionan exactamente igual que en las fórmulas de Excel y que en la mayoría de los lenguajes de programación: a una función se le llama usando su nombre y pasando un conjunto de parámetros entre paréntesis. Por ejemplo, para sumar la secuencia (3, 9, 14) escribiríamos:

```
sum( (3, 9, 14) )
```

Y el resultado sería 26. Observe que en este caso los paréntesis exteriores son parte de la llamada a la función sum(), mientras que los paréntesis interiores son parte de la secuencia que pasamos como parámetro único de la función.

Igualmente, ahora podemos calcular el total de la factura sumando todos los conceptos del ejemplo anterior:

```
sum( for $x in $DB1/sales/invoice/item  
return $x/price * $x/quantity )
```

Esto puede ser muy útil a la hora de generar formularios interactivos móviles dentro de soluciones móviles donde se pueden ver valores totales que cambian dinámicamente a medida que el usuario introduce los datos en el formulario.

XPath contiene una gran cantidad de funciones integradas para valores numéricos (p. ej. round, abs, floor, etc.), para la manipulación de cadenas de texto (p. ej. concat, lower-case, contains, replace), funciones de duración, fecha y hora y operaciones de agregado para secuencias (p. ej. count, sum, avg, min, max). En el generador de XPath de MobileTogether es muy fácil crear expresiones XPath usando estas funciones porque ofrece un práctico directorio que tiene todas las funciones acompañadas de una breve descripción e información sobre los parámetros necesarios. Estas funciones están organizadas de forma lógica para que pueda encontrar la función que necesita en cada momento.

Expresión if

En los lenguajes de programación y de expresiones suele ser necesario decidirse por una forma alternativa de calcular un resultado determinado, dependiendo de si se cumplen o no ciertos requisitos. Para esto es fundamental la instrucción if, que permite definir resultados alternativos dependiendo de los criterios de prueba que quiera utilizar.

En XPath la instrucción if se utiliza de esta manera:

```
if ( expresión-de-prueba ) then expression else expression
```

siendo expresión-de-prueba una expresión XPath lógica cuyo resultado puede ser true o false.

Por ejemplo, en MobileTogether puede usar una instrucción como esta para que el color del texto de una etiqueta o de un campo de edición dependa del valor numérico de los datos.

Imagine que un elemento de la fuente de datos está asociado con un campo de edición.

Podemos editar el campo de edición y en la pestaña Propiedades podemos seleccionar la propiedad Color del texto. En lugar de definir el color con el selector de colores, podemos hacer clic en el icono XPath de la barra de herramientas y definir el color del texto por medio de una expresión XPath:

```
if ( $MT_ControlValue >= 0 ) then "green" else "red"
```

Aquí usamos una de las variables integradas de MobileTogether para referirnos al control actual y probamos si su valor es mayor o igual que cero.

Dependiendo del resultado de esta prueba, el resultado será "green" (verde) o "red" (rojo) y el color del texto cambiará dependiendo de los datos que introduzca el usuario en el control.

MobileTogether ofrece muchas variables integradas que empiezan con "\$MT_" y que sirven para tomar decisiones en función del tamaño de la pantalla del dispositivo móvil, del sistema operativo y de muchos otros factores. Todas estas variables globales integradas están en el cuadro de diálogo "Variables globales" del menú Proyecto de MobileTogether Designer.

XQuery

Hasta ahora nos hemos centrado en el lenguaje XPath porque es el lenguaje basado en XML más común y tiene unos 15 años de historia. Pero además de XPath, MobileTogether también es compatible con XQuery, un lenguaje mucho más potente. XQuery es un supraconjunto de XPath, es decir, que cada expresión que es válida en XPath también es válida en XQuery. Sin embargo, XQuery ofrece varias construcciones de gran potencia para manipular y construir documentos XML de forma instantánea.

Expresión FLWOR

La expresión XQuery más importante es la llamada expresión FLWOR. FLWOR es el acrónimo inglés de “For-Let-Where-Order-Return” y suele pronunciarse como la palabra inglesa “flower” (flor). Funciona de manera parecida a la instrucción SQL SELECT, que tiene cláusulas similares, como FROM, WHERE, ORDER BY, etc. La expresión FLWOR permite consultar datos XML y de BD con mayor eficacia que con expresiones XPath.

La parte “for” de la expresión ya la conocemos. También conocemos la cláusula “return” de la expresión “for”. Sin embargo, en XQuery la cláusula “return” es mucho más flexible porque permite crear estructuras XML enteras desde cero con ayuda de constructores. Básicamente, la cláusula “return” se evalúa en cada elemento del bucle “for” pero en lugar de producir un solo valor en una secuencia, ahora puede generar una subestructura XML entera, como veremos en los siguientes ejemplos.

El propósito de la cláusula “let” de la expresión FLWOR es definir las variables adicionales que podemos usar más tarde para generar nuestro resultado. Este puede venir de la misma fuente de datos o de otra fuente de datos distinta, pero usa la variable de bucle para seleccionar ciertos elementos de la misma fuente.

Por ejemplo, imagine que una fuente de datos contiene información sobre los departamentos de una organización y que otra fuente de datos contiene información sobre los empleados. Podría crear una expresión FLWOR que seleccione los empleados de ciertos departamentos:

```
for $d in $DEPARTMENTS/depts/deptno
let $e := $EMPLOYEES/emp[deptno = $d]
```

...

La cláusula “where” permite filtrar los datos en función de ciertos criterios (algo parecido a los predicados de las expresiones XPath). Por ejemplo, podemos filtrar los departamentos en función de su número de empleados:

```
for $d in $DEPARTMENTS/depts/deptno
let $e := $EMPLOYEES/emp[deptno = $d]
where count($e) >= 10
```

...

Por último, la cláusula "order by" permite ordenar los resultados según cualquier criterio, lo cual es muy útil a la hora de generar documentos XML nuevos.

Ahora vamos a generar toda la expresión FLWOR usando las dos fuentes de datos mencionadas anteriormente. Por ejemplo, podemos generar una tabla de datos que ofrecerá la información agregada sobre el número de empleados y el salario medio para los departamentos que tengan más de 10 empleados:

```
for $d in $DEPARTMENTS/depts/deptno
let $e := $EMPLOYEES/emp[deptno = $d]
where count($e) >= 10
order by avg($e/salary) descending
return
<big-dept>
{ $d,
<headcount>{count($e)}</headcount>,
<avgsal>{avg($e/salary)}</avgsal>
}
</big-dept>
```

O, retomando el ejemplo de la factura, podemos recorrer varias facturas emitidas después de determinada fecha para calcular su cantidad total y después generar una tabla de resultados ordenados en la que la primera factura sea la de mayor cantidad y la última sea la de menor cantidad:

```
for $inv in $DB1/sales/invoice
let $total := sum (for $x in item return $x/price * $x/quantity)
where $inv/date >= "2014-01-01"
order by $total descending
return <invtotal>{ $total }</invtotal>
```

Como puede ver, la flexibilidad y la potencia de las expresiones XQuery supera con creces a las expresiones XPath. Sin embargo, esta mayor potencia supone inevitablemente una mayor complejidad y puede que tarde algo más en dominar las expresiones XQuery.

Gracias al generador y evaluador de expresiones XQuery y XPath de MobileTogether podrá escribir expresiones correctas y podrá probarlas con sus propios datos. Combinado con el potente simulador que le permite ejecutar sus aplicaciones móviles directamente en el MobileTogether Designer, puede hacer pruebas y depurar su selección de datos mediante XPath y XQuery fácilmente y generar potentes aplicaciones móviles mucho más rápido que con cualquier otro enfoque.

Resumen

En este documento ofrecemos una breve introducción a los conceptos más importantes de XPath y de XQuery para que pueda empezar a trabajar con datos en MobileTogether. Esperamos que esta breve introducción le sirva de base para empezar a crear potentes selecciones de datos en su flujo de trabajo móvil, producir gráficos precisos y personalizar la interfaz de su solución móvil.

No obstante, debe tener en cuenta que hay muchos aspectos de XPath y XQuery que no hemos cubierto en esta breve introducción. Para obtener más información consulte nuestro curso gratuito sobre XPath, destinado a principiantes y a usuarios avanzados.

MobileTogether es compatible con XPath 3.0 y XQuery 3.0. La ventaja de usar un estándar internacional como XPath y XQuery es que muchos usuarios ya conocen XPath por su experiencia con XML, XSLT o XML Schema. Por tanto, en MobileTogether pueden aplicar todos esos conocimientos previos. Y es muy fácil aprender a usar las versiones más recientes del estándar XPath (XPath 2.0 y 3.0).

Además, XPath y XQuery son estándares desarrollados por el W3C y no son lenguajes propietarios creados por una empresa determinada.

Para consultar la especificación XPath 3.0 y XQuery 3.0 siga los enlaces que aparecen a continuación:

- [XML Path Language \(XPath\) 3.0](#)
- [XQuery 3.0: An XML Query Language](#)
- [XPath and XQuery Functions and Operators 3.0](#)
- [XQuery and XPath Data Model 3.0](#)

Por último, no olvide que puede usar XMLSpy para crear, desarrollar y depurar expresiones XPath y XQuery más complejas y utilizarlas más tarde en MobileTogether. XMLSpy ofrece una función de evaluación de XPath y un depurador para XQuery, además de un generador de perfiles para optimizar el rendimiento de las expresiones.

Consulte los [cursos en línea de Altova](#) y los archivos de ayuda para obtener más información sobre XPath 3.0 y XQuery 3.0. Y si tiene alguna pregunta sobre cómo trabajar con XPath y XQuery en MobileTogether, visite el foro de soporte técnico <https://support.mobiletogether.com/>.