



Урок 4

Парсинг HTML, XPath

Введение

Основы

Узлы

- Атомарные значения

Отношения узлов

- Родитель

- Прямые потомки

- Элементы данных одного уровня

- Предки

- Потомки

Синтаксис XPath

- Выбор узлов

- Предикаты

- Выбор неизвестных заранее узлов

- Выбор нескольких путей

- Прочие обозначения в XPath

- Оси XPath

- Выражение пути выборки

- Примеры

Введение

XPath (XML Path Language) — язык запросов к элементам XML-документа. Разработан для организации доступа к частям документа XML в файлах трансформации XSLT и является стандартом консорциума W3C. XPath призван реализовать навигацию по DOM в XML. В XPath используется компактный синтаксис, отличный от принятого в XML. В 2007 году завершилась разработка версии 2.0, которая теперь является составной частью языка XQuery 1.0. В декабре 2009 года началась разработка версии 2.1, которая использует XQuery 1.1.

На данный момент самой популярной версией является XPath 1.0. Это связано с отсутствием поддержки XPath 2.0 со стороны открытых библиотек. В частности, речь идёт о LibXML, от которой зависит поддержка языка в браузерах, с одной стороны, и поддержка со стороны серверного интерпретатора, с другой.

ОСНОВЫ

XML имеет древовидную структуру. В самостоятельном XML-документе всегда имеется один корневой элемент (инструкция `<?xml version="1.0"?>` к дереву элементов отношения не имеет), в котором допустим ряд вложенных элементов, некоторые из которых тоже могут содержать вложенные элементы. Так же могут встречаться текстовые узлы, комментарии и инструкции. Можно считать, что XML-элемент содержит массив вложенных в него элементов и массив атрибутов.

У элементов дерева бывают элементы-предки и элементы-потомки (у корневого элемента предков нет, а у тупиковых элементов (листьев дерева) нет потомков). Каждый элемент дерева находится на определённом уровне вложенности (далее — «уровень»). Элементы упорядочены в порядке расположения в тексте XML, и поэтому можно говорить об их предыдущих и следующих элементах. Это очень похоже на организацию каталогов в файловой системе.

Строка XPath описывает способ выбора нужных элементов из массива элементов, которые могут содержать вложенные элементы. Начинается отбор с переданного множества элементов, на каждом шаге пути отбираются элементы, соответствующие выражению шага, и в результате оказывается отобрано подмножество элементов, соответствующих данному пути.

Для примера возьмём следующий XHTML документ:

```
1 <html>
2 <body>
3   <div>Первый слой
4     <span>блок текста в первом слое</span>
5   </div>
6   <div>Второй слой</div>
7   <div>Третий слой
8     <span class="text">первый блок в третьем слое</span>
9     <span class="text">второй блок в третьем слое</span>
10    <span>третий блок в третьем слое</span>
11  </div>
12  <span>четвёртый слой</span>
13  <img />
14 </body>
15 </html>
```

XPath-путь `/html/body/*/span[@class]` будет соответствовать в нём двум элементам исходного документа —

```
<span class="text">первый блок в третьем слое</span>
```

и

```
<span class="text">второй блок в третьем слое</span>.
```

Элементы пути преимущественно пишутся в XPath в краткой форме. Полная форма приведённого выше пути имеет вид `/child::html/child::body/child::*/child::span[attribute::class]`

Чтобы выбрать узлы или наборы узлов в документе, XPath использует выражения пути.

Такие выражения очень похожи на выражения пути, которые используются в традиционных файловых системах:



Путь состоит из *шагов адресации*, которые разделяются символом «косая черта» `/`.

Каждый шаг адресации состоит из трёх частей:

- ось (по умолчанию `child::`, ось элементов). Кроме отбора по оси вложенных элементов, можно отбирать разным другим осям элементов и по оси атрибутов (`attribute::`, она же обозначается символом `@`) (см. ниже).
- выражение, определяющее отбираемые элементы (в примере отбор делается по соответствию элементов документа именам `html`, `body`, `span`, и используется символ `*`, который отберёт все элементы оси)
- предикаты (в данном примере это `attribute::class`) - дополнительные условия отбора. Их может быть несколько. Каждый предикат заключается в квадратные скобки, и подразумевает логическое выражение для проверки отбираемых элементов. Если предиката нет, то отбираются все подходящие элементы.

Анализ пути ведётся слева направо, и начинается либо в контексте первого элемента корневого узла (в данном примере это элемент `html`), и тогда по оси `child::` будут вложенные в него элементы (в данном примере это один элемент `body`), что удобно в случае обработки обычного XML-документа с одним корневым узлом, либо, если в начале XPath указан символ `/`, в контексте со всеми корневыми элементами переданного XML по оси `child::` (в данном примере это будет один элемент `html`). На каждом шаге адресации в текущем контексте отбираются элементы, подходящие под указанные в шаге условия, и их перечень берётся как контекст для следующего шага или как возвращаемый результат.

Таким образом, первый шаг `/child::html` явным образом делает текущим контекстом для следующего шага перечень из одного элемента `html`, что было бы и так сделано неявно, если этот шаг не был обозначен.

На втором шаге адресации в данном примере (шаг `child::body`) контекстом является перечень из одного элемента `html`. Ось `child::` говорит о том, что необходимо смотреть на имена вложенных элементов в текущем контексте, а условие проверки `body` говорит о том, что в формируемый набор элементов нужно включить те узлы, у которых имя `body`. Таким образом, в ходе второго шага адресации получаем набор узлов, состоящий всего из одного элемента `body`, который и становится контекстом для третьего шага.

Третий шаг адресации: `child::*`. Ось `child::` содержит всех непосредственных потомков элемента `body`, а условие проверки `*` говорит о том, что в формируемый перечень нужно включить элементы основного типа с любым именем. В ходе этого шага получаем перечень, состоящий из трёх элементов `div`, одного `span` и одного элемента `img` — итого, пять элементов.

Четвёртый шаг адресации: `child::span/@class`. Его контекстом является перечень из пяти элементов, поэтому исходящий перечень создаётся в пять проходов (за пять итераций). При первой

итерации узлом контекста становится первый `div`. Согласно заданной оси `child::` и правилу проверки `span`, в набор должны включаться непосредственные потомки этого `div`-а, имя которых равно `span`. Там такой один. При второй итерации в набор ничего добавляться не будет, так как у второго `div` нет потомков. Третья итерация увидит сразу три элемента `span`. Четвёртая ничего не увидит, так как у элемента `span` нет потомков `span`, а то что он сам `span` — не важно, ведь просматриваются именно потомки. Пятая тоже ничего не увидит, у элемента `img` тоже нет потомков `span`. Итак, в ходе проверки мог бы быть получен набор узлов, состоящий из четырёх элементов `span`. Это и было бы контекстом для последующей обработки, не будь на этом шаге указано предиката.

Но так как предикат на четвёртом шаге есть, по мере выполнения каждого из пяти проходов будет производиться дополнительная фильтрация отбираемых элементов. В данном случае у предиката ось `attribute::` говорит о необходимости проверить, есть ли у отбираемого узла атрибуты, а условие `class` требует оставить лишь те узлы, у которых задан атрибут с именем `class`. И поэтому на первой итерации единственный найденный `span` фильтрацию предикатом не пройдёт, на третьей итерации фильтрацию пройдут два элемента из трёх, и в итоге, несмотря на то, что фильтрация происходит за пять итераций, в окончательный набор попадают только два элемента `span`.

Узлы

Всего в XPath выделяется семь типов узлов:

1. элемент,
2. атрибут,
3. текст,
4. пространство имен,
5. инструкция обработки,
6. комментарий,
7. узлы документа.

Документ XML рассматривается, как дерево узлов. Элемент, находящийся в самом верху этого дерева, называется корневым элементом.

Рассмотрим на следующий XML документ:

```
<?xml version="1.0" encoding="UTF-8"?>
<messages>
  <note id="1">
    <to>Джону</to>
    <from>Дженни</from>
    <heading>Напоминание</heading>
    <body>Купи хлеба!</body>
  </note>
</messages>
```

Здесь:

- **<messages>** - узел корневого элемента
- **<to>Джону</to>** - узел элемента
- **note id="1"** - узел атрибута

Атомарные значения

Узлы, у которых нет родителей и потомков, называются атомарными значениями.

Пример:

Джону

Атомарные значения или узлы называются элементами данных.

Отношения узлов

Родитель

У каждого элемента есть один родитель.

В следующем примере элемент **note** является родителем элементов **to**, **from**, **heading** и **body**:

```
<note>  
  <to>Джону</to>  
  <from>Дженни</from>  
  <heading>Напоминание</heading>  
  <body>Купи хлеба!</body>  
</note>
```

Прямые потомки

У элемента может быть ноль, один или больше прямых потомков.

В следующем примере элементы **to**, **from**, **heading** и **body** являются прямыми потомками элемента **note**:

```
<note>  
  <to>Джону</to>  
  <from>Дженни</from>  
  <heading>Напоминание</heading>  
  <body>Купи хлеба!</body>  
</note>
```

Элементы данных одного уровня

Узлы, у которых один и тот же родитель, являются элементами данных одного уровня.

В следующем примере элементы **to**, **from**, **heading** и **body** являются элементами данных одного уровня:

```
<note>  
  <to>Джону</to>  
  <from>Дженни</from>  
  <heading>Напоминание</heading>  
  <body>Купи хлеба!</body>  
</note>
```

Предки

Предками называются родители узлов, родители родителей и т.д.

В следующем примере предками элемента **to** являются элементы **note** и **messages**:

```
<messages>
  <note>
    <to>Джону</to>
    <from>Дженни</from>
    <heading>Напоминание</heading>
    <body>Купи хлеба!</body>
  </note>
</messages>
```

Потомки

Потомками называются прямые потомки узлов, прямые потомки прямых потомков и т. д.

В следующем примере потомками элемента **messages** являются элементы **note**, **to**, **from**, **heading** и **body**:

```
<messages>
  <note>
    <to>Джону</to>
    <from>Дженни</from>
    <heading>Напоминание</heading>
    <body>Купи хлеба!</body>
  </note>
</messages>
```

Синтаксис Xpath

Выбор узлов

Чтобы выбрать узлы в XML документе, XPath использует выражения пути. Узел выбирается следуя по заданному пути. Наиболее полезные выражения пути:

Выражение	Результат
<i>имя_узла</i>	Выбирает все узлы с именем "имя_узла"
/	Выбирает от корневого узла
//	Выбирает узлы от текущего узла, соответствующего выбору, независимо от их местонахождения
.	Выбирает текущий узел

Выражение	Результат
..	Выбирает родителя текущего узла
@	Выбирает атрибуты

В следующей таблице приводятся некоторые выражения XPath, позволяющие сделать некоторые выборки по демонстрационному XML документу:

Выражение XPath	Результат
messages	Выбирает все узлы с именем "messages"
/messages	Выбирает корневой элемент сообщений Примечание: Если путь начинается с косой черты (/), то он всегда представляет абсолютный путь к элементу!
messages/note	Выбирает все элементы note, являющиеся потомками элемента messages
//note	Выбирает все элементы note независимо от того, где в документе они находятся
messages//note	Выбирает все элементы note, являющиеся потомками элемента messages независимо от того, где они находятся от элемента messages
//@date	Выбирает все атрибуты с именем date

Предикаты

Предикаты позволяют найти конкретный узел или узел с конкретным значением.

Предикаты всегда заключаются в квадратные скобки.

В следующей таблице приводятся некоторые выражения XPath с предикатами, позволяющие сделать выборки по демонстрационному XML документу:

Выражение XPath	Результат
/messages/note[1]	Выбирает первый элемент note, который является прямым потомком элемента messages. Примечание: В IE 5,6,7,8,9 первым узлом будет [0], однако согласно W3C это должен быть [1]. Чтобы решить

Выражение XPath	Результат
	эту проблему в IE, нужно установить опцию SelectionLanguage в значение XPath. В JavaScript: <code>xml.setProperty("SelectionLanguage","XPath");</code>
<code>/messages/note[last()]</code>	Выбирает последний элемент note, который является прямым потомком элемента messages.
<code>/messages/note[last()-1]</code>	Выбирает предпоследний элемент note, который является прямым потомком элемента messages.
<code>/messages/note[position()=2]</code>	Выбирает второй по счету элемент note, который являются прямым потомком элемента messages.
<code>//heading[@date]</code>	Выбирает все элементы heading, у которых есть атрибут date
<code>//heading[@date="10/01/2008"]</code>	Выбирает все элементы heading, у которых есть атрибут date со значением "10/01/2008"

Выбор неизвестных заранее узлов

Чтобы найти неизвестные заранее узлы XML документа, XPath позволяет использовать специальные символы.

Спецсимвол	Описание
*	Соответствует любому узлу элемента
@*	Соответствует любому узлу атрибута
node()	Соответствует любому узлу любого типа

В следующей таблице приводятся некоторые выражения XPath со спецсимволами, позволяющие сделать выборки по демонстрационному XML документу:

Выражение XPath	Результат
<code>/messages/*</code>	Выбирает все элементы, которые являются прямыми потомками элемента messages

Выражение XPath	Результат
<code>//*</code>	Выбирает все элементы в документе
<code>//heading[@*]</code>	Выбирает все элементы heading, у которых есть по крайней мере один атрибут любого типа

Выбор нескольких путей

Использование оператора `|` в выражении XPath позволяет делать выбор по нескольким путям.

В следующей таблице приводятся некоторые выражения XPath, позволяющие сделать выборки по демонстрационному XML документу:

Выражение XPath	Результат
<code>//note/heading //note/body</code>	Выбирает все элементы heading И body из всех элементов note
<code>//heading //body</code>	Выбирает все элементы heading И body во всем документе

Прочие обозначения в XPath

Обозначение	Описание
<code>*</code>	Обозначает <i>любое</i> имя или набор символов по указанной оси, например: <code>*</code> — любой дочерний узел; <code>@*</code> — любой атрибут
<code>\$name</code>	Обращение к переменной. <code>name</code> — имя переменной или параметра
<code>{i}</code>	Дополнительные условия выборки (или предикат шага адресации). Должен содержать логическое значение. Если содержит числовое, считается что это порядковый номер узла, что эквивалентно приписыванию перед этим числом выражения <code>position() =</code>
<code>{ }</code>	Если применяется внутри тега другого языка (например HTML), то XSLT-процессор рассматривает содержимое фигурных скобок как XPath
<code>/</code>	Определяет уровень дерева, т. е. разделяет шаги адресации
<code> </code>	Объединяет результат. Т. е., в рамках одного пути можно написать несколько путей разбора через знак <code> </code> , и в результат такого выражения войдёт всё, что будет найдено любым из этих путей

Оси XPath

Ось определяет отношение узлового набора по отношению к текущему узлу.

Название оси	Результат
ancestor	Выбирает всех предков текущего узла
ancestor-or-self	Выбирает всех предков текущего узла и сам текущий узел
attribute	Выбирает все атрибуты текущего узла
child	Выбирает всех прямых потомков текущего узла
descendant	Выбирает всех потомков текущего узла
descendant-or-self	Выбирает всех потомков текущего узла и сам текущий узел
following	Выбирает все элементы в документе после закрывающего тега текущего узла
following-sibling	Выбирает все элементы одного уровня после текущего узла
namespace	Выбирает все узлы пространства имен текущего узла
parent	Выбирает родителя текущего узла
preceding	Выбирает все узлы, которые появляются перед текущим узлом, за исключением предков, узлов атрибутов и пространства имен
preceding-sibling	Выбирает все элементы одного уровня до текущего узла
self	Выбирает текущий узел

Выражение пути выборки

Путь выборки может быть абсолютным или относительным.

Абсолютный путь выборки начинается с символа косой черты (/), а относительный нет. В обоих случаях такой путь содержит один или более шагов, разделенных косой чертой:

Абсолютный путь:

/шаг/шаг/...

Относительный путь:

шаг/шаг/...

Каждый шаг вычисляется по отношению к узлам в текущем узловом наборе.

Шаг содержит:

- ось (определяет древовидное отношение между выбранными узлами и текущим узлом)
- проверка узла (определяет узел внутри оси)
- ноль или больше предикатов (для более точного указания выбранного узлового набора)

Синтаксис для шага выборки имеет следующий вид:

```
имяОси : : проверкаУзла [ предикат ]
```

Примеры

Пример	Результат
child::note	Выбирает все узлы note, которые являются прямыми потомками текущего узла
attribute::date	Выбирает атрибут date текущего узла
child::*	Выбирает всех прямых потомков текущего узла
attribute::*	Выбирает все атрибуты текущего узла
child::text()	Выбирает все текстовые узлы текущего узла
child::node()	Выбирает всех прямых потомков текущего узла
descendant::note	Выбирает всех потомков note текущего узла
ancestor::note	Выбирает всех предков note текущего узла
ancestor-or-self::note	Выбирает всех предков note текущего узла, а также сам текущий узел, если это узел note
child::* / child::heading	Выбирает всех прямых потомков прямых потомков ("внуков") heading текущего узла

Домашнее задание

Написать приложение, которое собирает основные новости с сайтов mail.ru, lenta.ru, yandex-новости.

Для парсинга использовать xpath. Структура данных должна содержать:

- название источника,
- наименование новости,
- ссылку на новость,
- дата публикации

Используемая литература

1. <https://ru.wikipedia.org/wiki/XPath>
2. <https://wm-help.net/lib/b/book/3998533050/156>
3. <https://msiter.ru/tutorials/xpath/syntax>