

Лекция 7. XPath

Web System Design and Development

План лекции

- Понятие XPath.
- Основы XPath.
- Операторы и специальные символы.
- Примеры.

Технология XML

XML (Extensible Markup Language)

- XML – язык разметки
- XML отделяет хранение данных от их представления (HTML)
- XML упрощает обмен данными
- На базе XML легко создаются новые языки
- XML-документ имеет древовидную структуру (root element, child element)
- элемент XML состоит из:
 - **названия элемента**;
 - **контента**;
 - **возможно, атрибута элемента**;
 - **возможно, значения атрибута элемента**

```
<?xml version = "1.0">
<item number="00001">
  <name>
    <first>Jane</first>
    <middle>Q</middle>
    <last>Public</last>
  </name>
  <phone type="voice">
    <areacode>407</areacode>
    <number>555-1212</number>
  </phone>
  <phone type="fax">
    <areacode>407</areacode>
    <number>555-1213</number>
  </phone>
  <email>jpublic@gmail.com</email>
</item>
```

Понятие XPath

XPath (XML Path Language)

- язык запросов к элементам XML-документа

Используется для навигации по DOM в XML.

Представляет собой некий гибрид между указанием пути в командной строке и регулярными выражениями.

XPath представляет XML-документ в виде дерева из **узлов** различных типов – **элементов, атрибутов и текста**.

Выражения XPath могут идентифицировать эти узлы по их типу, имени и значениям, а также по взаимоотношениям узлов в документе.

Например, запрос **"найти элементы 'author' с атрибутом 'period' и значением 'classical', содержащиеся в элементе 'authors' "** может быть записан:
`"/authors/author[@period='classical']"`

Основы XPath

Выражение XPath строится подобно пути к файлу в файловой системе, откуда и происходит название языка "XML Path", сокращенно XPath.

- Язык XPath представляет документ в виде **дерева**, корнем которого служит корневой элемент документа.
- От корня отходят ветви, заканчивающиеся узлами. Узлами служат, например, вложенные элементы, их атрибуты и тексты, составляющие содержимое корневого элемента.
- От каждого вложенного элемента отходят свои ветви, рекурсивно повторяющие ветви корневого элемента.

У каждого узла может быть только один узел - "предок", причем предком может быть только узел корневого или другого элемента дерева, но не атрибут и не текстовый узел.

XPath включает в себя более 100 встроенных функций.

Основы XPath

Запрос XPath состоит из пути к элементу в дереве или вызова функции, результатом запроса будет набор всех элементов, соответствующих пути.

XPath:

```
/wikimedia/projects/project/editions/*[2]
```

XML document:

```
<?xml version="1.0" encoding="utf-8"?>
<wikimedia>
  <projects>
    <project name="Wikipedia" launch="2001-01-05">
      <editions>
        <edition language="English">en.wikipedia.org</edition>
        <edition language="German">de.wikipedia.org</edition>
        <edition language="French">fr.wikipedia.org</edition>
        <edition language="Polish">pl.wikipedia.org</edition>
      </editions>
    </project>
    <project name="Wiktionary" launch="2002-12-12">
      <editions>
        <edition language="English">en.wiktionary.org</edition>
        <edition language="French">fr.wiktionary.org</edition>
        <edition language="Vietnamese">vi.wiktionary.org</edition>
      </editions>
    </project>
  </projects>
</wikimedia>
```

Основы XPath

Пути адресации

Относительные

Состоят из последовательности одного или нескольких шагов адресации, отделенных друг от друга символом /

step/step/...

Абсолютные

Состоят из символа / (находит корневой узел документа) , за которым может следовать относительный путь адресации.

/step/step/...

Контекст — это некая точка отсчета, относительно которой рассчитывается следующий шаг адресации.

На каждом шаге адресации мы получаем новый набор узлов документа, и этот набор становится контекстом для следующего шага адресации.

Основы XPath

XPath путь делится на **шаги адресации (location step)**, разделенные символом /

`/html/body/*/span[@class]`

полный синтаксис: `/child::html/child::body/child::*/child::span[attribute::class]`

ось::условие_проверки_узлов [предикат]

Ось

Указывает общее направление, в котором выполняется шаг определения расположения по отношению к узлу контекста (в примере `child::`).

**условие
проверки
узла**

Определяет тип узла либо развернутое имя узлов, которые изначально выбираются шагом определения расположения (в примере - имена элементов документа `html`, `body`, `span`, а символ `*` означает элемент с любым именем).

предикат

Фильтр, который определяет критерий выбора для дальнейшего уточнения списка узлов-кандидатов (`attribute::class`).

Основы XPath

Ось поиска говорит о том, **КУДА** мы движемся, условие проверки узла показывает, **ЧТО** мы ищем, а предикат уточняет, **КАКИМИ** свойствами должен обладать искомый объект.

ОСИ

Полная запись	Сокращенная запись	Результат
ancestor::		Возвращает множество предков.
ancestor-or-self::		Возвращает множество предков и текущий элемент.
attribute::	@	Возвращает множество атрибутов текущего элемента.
child::	часто просто опускают	Возвращает множество потомков на один уровень ниже.
descendant::	//	Возвращает полное множество потомков.

ОСНОВЫ XPath

Полная запись	Сокращенная запись	Результат
descendant-or-self::		Возвращает полное множество потомков и текущий элемент.
following::		Возвращает необработанное множество, ниже текущего элемента.
following-sibling::		Возвращает множество элементов на том же уровне, следующих за текущим.
namespace::		Возвращает множество, имеющее пространство имён (то есть присутствует атрибут xmlns).
parent::	..	Возвращает предка на один уровень назад.
preceding::		Возвращает множество обработанных элементов исключая множество предков.
preceding-sibling::		Возвращает множество элементов на том же уровне, предшествующих текущему.
self::	.	Возвращает текущий элемент.

ОСНОВЫ XPath

Примеры

child::node()	Выберет всех потомков узла контекста независимо от их типа узла.
attribute::name	Выберет атрибут name узла контекста.
descendant::para	Выберет потомков элемента <para> узла контекста.
ancestor::div	Выберет всех предков элемента <div> узла контекста.
ancestor-or-self::div	Выберет предков элемента <div> узла контекста и, если узлом контекста является элемент <div>, выберите также узел контекста.
descendant-or-self::para	Выберет потомков элемента <para> узла контекста и, если узлом контекста является элемент <para>, выберите также узел контекста.
self::para	Выберет узел контекста, если им является элемент <para>, в противном случае не выбирайте ничего.
child::chapter/descendant::para	Выберет потомков элемента <para>, являющихся дочерними элементами по отношению к элементу <chapter> узла контекста.

ОСНОВЫ XPath

Функции XPath

Функции для наборов
узлов (Node-Set)

ID - выбирает элементы по их уникальному идентификатору.

Функции с
множествами

last() - возвращает номер последнего элемента в множестве.

Строковые функции

string - преобразует объект в строку.

Логические функции

boolean - преобразует аргумент в логическое значение.

Числовые функции

round - возвращает ближайшее к аргументу целое значение.

http://www.w3schools.com/XPath/xpath_functions.asp

Операторы и специальные символы

Выражения XPath создаются с помощью операторов и специальных символов:

/

Оператор «дочерний элемент»: выбирает непосредственные дочерние элементы коллекции, указанной слева. Если этот оператор пути стоит в начале шаблона, будут выбраны дочерние элементы корневого узла.

/AAA

- выбирается корневой узел AAA

```
<AAA>
  <BBB/>
  <CCC/>
  <BBB/>
  <BBB/>
  <DDD>
    <BBB/>
  </DDD>
  <CCC/>
</AAA>
```

/AAA/DDD/BBB

- выбираются все элементы BBB, являющиеся детьми элементов DDD, которые в свою очередь являются детьми корневого узла AAA

```
<AAA>
  <BBB/>
  <CCC/>
  <BBB/>
  <BBB/>
  <DDD>
    <BBB/>
  </DDD>
  <CCC/>
</AAA>
```

Операторы и специальные символы

//

Рекурсивный спуск; поиск заданного элемента на любой глубине. Если этот оператор пути стоит в начале шаблона, рекурсивный спуск будет вестись из корневого узла.

//BBB

- будут выбраны все элементы BBB

```
<AAA>
  <BBB/>
  <CCC/>
  <BBB/>
  <DDD>
    <BBB/>
  </DDD>
  <CCC>
    <DDD>
      <BBB/>
      <BBB/>
    </DDD>
  </CCC>
</AAA>
```

//DDD/BBB

- будут выбраны все элементы BBB, являющиеся детьми DDD

```
<AAA>
  <BBB/>
  <CCC/>
  <BBB/>
  <DDD>
    <BBB/>
  </DDD>
  <CCC>
    <DDD>
      <BBB/>
      <BBB/>
    </DDD>
  </CCC>
</AAA>
```

Операторы и специальные символы

```
<AAA>
  <XXX>
    <DDD>
      <BBB/>
      <BBB/>
      <EEE/>
      <FFF/>
    </DDD>
  </XXX>
  <CCC>
    <DDD>
      <BBB/>
      <BBB/>
      <EEE/>
      <FFF/>
    </DDD>
  </CCC>
  <CCC>
    <BBB>
      <BBB>
        <BBB/>
      </BBB>
    </BBB>
  </CCC>
</AAA>
```

.	Указывает текущий контекст.
..	Родитель узла текущего контекста.
*	Символ подстановки; выбирает все элементы независимо от их имени.

/AAA/CCC/DDD/*

- будут выбраны все элементы,
являющиеся прямыми потомками
/AAA/CCC/DDD

//*

- будут выбраны все элементы

```
<AAA>
  <BBB/>
  <CCC/>
  <BBB/>
  <BBB/>
  <DDD>
    <BBB/>
  </DDD>
  <CCC/>
</AAA>
```

Операторы и специальные символы

@	Атрибут.
@*	Символ подстановки для атрибута; выбирает все атрибуты независимо от имени.

//@id

- будут выбраны все атрибуты @id

```
<AAA>
  <BBB id = "b1"/>
  <BBB id = "b2"/>
  <BBB name = "bbb"/>
  <BBB/>
</AAA>
```

//BBB[@*]

- будут выбраны все элементы BBB, имеющие хоть какой-нибудь атрибут

```
<AAA>
  <BBB id = "b1"/>
  <BBB id = "b2"/>
  <BBB name = "bbb"/>
  <BBB/>
</AAA>
```

//BBB[@id]

- будут выбраны элементы BBB, имеющие атрибут id

```
<AAA>
  <BBB id = "b1"/>
  <BBB id = "b2"/>
  <BBB name = "bbb"/>
  <BBB/>
</AAA>
```


Операторы и специальные символы

:	Разделитель пространства имени; отделяет префикс пространства имени от собственно имени элемента или атрибута.
()	Группирует операции для явного задания порядка их выполнения.
[]	Применяет шаблон фильтра.

/AAA/BBB[1]

- будет выбран первый потомок
BBB элемента AAA

```
<AAA>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
</AAA>
```

/AAA/BBB[last()]

- будет выбран последний потомок
BBB элемента AAA

```
<AAA>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
</AAA>
```

Примеры

Выражение	Ссылается на
author/first-name	Все элементы <first-name> внутри элемента <author> текущего узла контекста.
bookstore//title	Все элементы <title> на первом или более глубоких уровнях элемента <bookstore> (наследники произвольного уровня). <u>Заметьте отличие</u> от следующего шаблона: bookstore/*/title.
bookstore/*/title	Все элементы <title>, являющиеся "внуками" элементов <bookstore>.
bookstore//book/excerpt//emph	Все элементы <emph> в любом месте внутри элементов <excerpt>, дочерних по отношению к элементам <book>, находящимся в любом месте внутри элемента <bookstore>.
./title	Все элементы <title> на первом или более глубоких уровнях текущего контекста. Только в этой ситуации требуется нотация с использованием точек.

Примеры

Выражение	Ссылается на
author/*	Все дочерние элементы элементов <author>.
book/*/last-name	Все элементы <last-name>, являющиеся "внуками" элементов <book>.
/	Все элементы-"внуки" текущего контекста.
my:book	Элемент <book> из пространства имен my.
my:*	Все элементы из пространства имен my.

*** означает все элементы, являющиеся дочерними для текущего контекста, независимо от имени тега.**

Примеры

Выражение	Ссылается на
@style	Атрибут style контекста текущего элемента.
price/@exchange	Атрибут exchange элементов <price> в текущем контексте.
book/@style	Атрибут style всех <book> элементов.
@style	Атрибут style контекста текущего элемента.
price/@exchange	Атрибут exchange элементов <price> в текущем контексте.
@*	Все атрибуты текущего контекстного узла.
@my:*	Все атрибуты из пространства имен my. Сюда не входят атрибуты с неполным именем, принадлежащие элементам из пространства имен my.

Примеры

Выражение	Ссылается на
book[last()]	Последний элемент <book> в контексте текущего узла.
book/author[last()]	Последний дочерний элемент <author> каждого элемента <book> в контексте текущего узла.
(book/author)[last()]	Последний элемент <author> из всего набора элементов <author>, дочерних по отношению к элементам <book> в контексте текущего узла.
book[excerpt]	Все элементы <book>, которые содержат хотя бы один дочерний элемент <excerpt>.
book[excerpt]/title	Все элементы <title>, дочерние по отношению к элементам <book> и содержащие хотя бы один дочерний элемент <excerpt>.
book[excerpt]/author[degree]	Все элементы <author>, содержащие по крайней мере один дочерний элемент <degree> и являющиеся дочерними для элементов <book>, содержащих по крайней мере один элемент <excerpt>.

Примеры

Выражение	Ссылается на
book[author/degree]	Все элементы <book>, у которых есть дочерние элементы <author>, у которых, в свою очередь, есть хотя бы один дочерний элемент <degree>.
author[degree][award]	Все элементы <author>, которые содержат хотя бы один элемент <degree> и хотя бы один элемент <award>.
author[degree and award]	Все элементы <author>, которые содержат хотя бы один элемент <degree> и хотя бы один элемент <award>.
author[(degree or award) and publication]	Все элементы <author>, которые содержат хотя бы один дочерний элемент <degree> или <award> и хотя бы один дочерний элемент <publication>.
author[degree and not(publication)]	Все элементы <author>, содержащие по крайней мере один дочерний элемент <degree> и не содержащие дочерних элементов <publication>.
author[not(degree or award) and publication]	Все элементы <author>, содержащие по крайней мере один дочерний элемент <publication> и не содержащие ни элементов <degree>, ни элементов <award>.

Примеры

Выражение	Ссылается на
<code>author[last-name = "Bob"]</code>	Все элементы <author>, которые содержат хотя бы один дочерний элемент <last-name> со значением Bob.
<code>author[last-name[1] = "Bob"]</code>	Все элементы <author>, первый дочерний элемент <last-name> которых имеет значение Bob. Заметьте, что это эквивалентно выражению в следующей строке.
<code>author[last-name [position()=1]= "Bob"]</code>	Все элементы <author>, первый дочерний элемент <last-name> которых имеет значение Bob.
<code>degree[@from != "Harvard"]</code>	Все элементы <degree>, у которых атрибут from не равен "Harvard".
<code>author[.= "Matthew Bob"]</code>	Все элементы <author> со значением Matthew Bob.
<code>author[last-name = "Bob" and ../price > 50]</code>	Все элементы <author>, имеющие дочерний элемент <last-name> со значением Bob и одноуровневый элемент <price> со значением выше 50.
<code>book[position() <= 3]</code>	Первые три книги (1, 2, 3).

Примеры

Выражение	Ссылается на
author[not(last-name = "Bob")]	Все элементы <author>, не содержащие дочерних элементов <last-name> со значением Bob.
author[first-name = "Bob"]	Все элементы <author>, которые содержат хотя бы один дочерний элемент <first-name> со значением Bob.
author[* = "Bob"]	все элементы author, имеющие какой-либо дочерний элемент со значением Bob.
author[last-name = "Bob" and first-name = "Joe"]	Все элементы <author>, имеющие дочерний элемент <last-name> со значением Bob и дочерний элемент <first-name> со значением Joe.
price[@intl = "Canada"]	Все элементы <price> в контекстном узле, имеющие атрибут intl со значением "Canada".
degree[position() < 3]	Первые два элемента <degree>, являющиеся дочерними для контекстного узла.
p/text()[2]	Второй текстовый узел в каждом элементе <p> контекстного узла.

Примеры

Выражение	Ссылается на
ancestor::book[1]	Ближайший элемент <book>, который является предком узла контекста.
ancestor::book[author][1]	Ближайший элемент <book>, который является предком узла контекста и содержит дочерний элемент <author>.
ancestor::author[parent::book][1]	Ближайший элемент <author>, который является предком в текущем контексте и является дочерним элементом для элемента <book>.

Дополнительные материалы

1. google.com
2. www.w3.org/TR/xpath/
3. <http://msdn.microsoft.com/ru-ru/library/ms256115%28v=vs.110%29.aspx>
4. <http://www.w3schools.com/XPath/default.asp>
5. <http://en.wikipedia.org/wiki/XPath>

... и другие учебники...



Q&A

Thank You

