第 31 章 XPath

学习要点：

1.IE 中的 XPath
2.W3C 中的 XPath
3.XPath 跨浏览器兼容

主讲教师：李炎恢
合作网站：http://www.ibeifeng.com
讲师博客：http://hi.baidu.com/李炎恢

XPath 是一种节点查找手段，对比之前使用标准 DOM 去查找 XML 中的节点方式，大大降低了查找难度，方便开发者使用。但是，DOM3 级以前的标准并没有就 XPath 做出规范；直到 DOM3 在首次推荐到标准规范行列。大部分浏览器实现了这个标准，IE 则以自己的方式实现了 XPath。

一．IE 中的 XPath

在 IE8 及之前的浏览器，XPath 是采用内置基于 ActiveX 的 XML DOM 文档对象实现的。在每一个节点上提供了两个方法：selectSingleNode()和 selectNodes()。

selectSingleNode()方法接受一个 XPath 模式（也就是查找路径），找到匹配的第一个节点并将它返回，没有则返回 null。

```
var user = xmlDom.selectSingleNode('root/user');      //得到第一个 user 节点
alert(user.xml);                                      //查看 xml 序列
alert(user.tagName);                                  //节点元素名
alert(user.firstChild.nodeValue);                     //节点内的值
```

上下文节点：我们通过 xmlDom 这个对象实例调用方法，而 xmlDom 这个对象实例其实就是一个上下文节点，这个节点指针指向的是根，也就是 root 元素之前。那么如果我们把这个指针指向 user 元素之前，那么结果就会有所变化。

```
//通过 xmlDom，并且使用 root/user 的路径
var user = xmlDom.selectSingleNode('root/user');
alert(user.tagName);                                  //user

//通过 xmlDom.documentElement，并且使用 user 路径，省去了 root
var user = xmlDom.documentElement.selectSingleNode('user');
alert(user.tagName);                                  //user

//通过 xmlDom，并且使用 user 路径，省去了 root
var user = xmlDom.selectSingleNode('user');
alert(user.tagName);                                  //找不到了，出错
```

PS：xmlDom 和 xmlDom.documentElement 都是上下文节点，主要就是定位当前路径查找的指针，而 xmlDom 对象实例的指针就是在最根上。

XPath 常用语法

```
//通过 user[n]来获取第 n+1 条节点，PS：XPath 其实是按 1 为起始值的
var user = xmlDom.selectSingleNode('root/user[1]');
alert(user.xml);

//通过 text()获取节点内的值
var user = xmlDom.selectSingleNode('root/user/text()');
alert(user.xml);
alert(user.nodeValue);

//通过//user 表示在整个 xml 获取到 user 节点，不关心任何层次
var user = xmlDom.selectSingleNode('//user');
alert(user.xml);

//通过 root//user 表示在 root 包含的层次下获取到 user 节点，在 root 内不关心任何层次
var user = xmlDom.selectSingleNode('root//user');
alert(user.tagName);

//通过 root/user[@id=6]表示获取 user 中 id=6 的节点
var user = xmlDom.selectSingleNode('root/user[@id=6]');
alert(user.xml);
```

PS：更多的 XPath 语法，可以参考 XPath 手册或者 XML DOM 手册进行参考，这里只提供了最常用的语法。

```
selectSingleNode()方法是获取单一节点，而 selectNodes()方法则是获取一个节点集合。
var users = xmlDom.selectNodes('root/user');          //获取 user 节点集合
alert(users.length);
alert(users[1].xml);
```

二．W3C 下的 XPath

在 DOM3 级 XPath 规范定义的类型中，最重要的两个类型是 XPathEvaluator 和 XPathResult。其中，XPathEvaluator 用于在特定上下文对 XPath 表达式求值。

XPathEvaluator 的方法

方法	说明
createExpression(e, n)	将 XPath 表达式及命名空间转化成 XPathExpression
createNSResolver(n)	根据 n 命名空间创建一个新的 XPathNSResolver 对象
evaluate(e, c, n ,t ,r)	结合上下文来获取 XPath 表达式的值

W3C 实现 XPath 查询节点比 IE 来的复杂，首先第一步就是需要得到 XPathResult 对象的实例。得到这个对象实例有两种方法，一种是通过创建 XPathEvaluator 对象执行 evaluate()

方法，另一种是直接通过上下文节点对象(比如 xmlDom)来执行 evaluate()方法。

```
//使用 XPathEvaluator 对象创建 XPathResult
var eva = new XPathEvaluator();
var result = eva.evaluate('root/user', xmlDom, null,
                          XPathResult.ORDERED_NODE_ITERATOR_TYPE, null);
alert(result);

//使用上下文节点对象(xmlDom)创建 XPathResult
var result = xmlDom.evaluate('root/user', xmlDom, null,
                          XPathResult.ORDERED_NODE_ITERATOR_TYPE, null);
alert(result);
```

相对而言，第二种简单方便一点，但 evaluate 方法有五个属性：1.XPath 路径、2.上下文节点对象、3.命名空间求解器(通常是 null)、4.返回结果类型、5 保存结果的 XPathResult 对象(通常是 null)。

对于返回的结果类型，有 10 中不同的类型

常量	说明
XPathResult.ANY_TYPE	返回符合 XPath 表达式类型的数据
XPathResult.ANY_UNORDERED_NODE_TYPE	返回匹配节点的节点集合，但顺序可能与文档中的节点的顺序不匹配
XPathResult.BOOLEAN_TYPE	返回布尔值
XPathResult.FIRST_ORDERED_NODE_TYPE	返回只包含一个节点的节点集合，且这个节点是在文档中第一个匹配的节点
XPathResult.NUMBER_TYPE	返回数字值
XPathResult.ORDERED_NODE_ITERATOR_TYPE	返回匹配节点的节点集合，顺序为节点在文档中出现的顺序。这是最常用到的结果类型
XPathResult.ORDERED_NODE_SNAPSHOT_TYPE	返回节点集合快照，在文档外捕获节点，这样将来对文档的任何修改都不会影响这个节点列表
XPathResult.STRING_TYPE	返回字符串值
XPathResult.UNORDERED_NODE_ITERATOR_TYPE	返回匹配节点的节点集合，不过顺序可能不会按照节点在文档中出现的顺序排列
XPathResult.UNORDERED_NODE_SNAPSHOT_TYPE	返回节点集合快照，在文档外捕获节点，这样将来对文档的任何修改都不会影响这个节点列表

PS：上面的常量过于繁重，对于我们只需要学习了解，其实也就需要两个：1.获取一个单一节、2.获取一个节点集合。

1.获取一个单一节点

```
var result = xmlDom.evaluate('root/user', xmlDom, null,
                          XPathResult.FIRST_ORDERED_NODE_TYPE, null);
if (result !== null) {
```

```
    alert(result.singleNodeValue.tagName);          //singleNodeValue 属性得到节点对象
}

2.获取节点集合
var result = xmlDom.evaluate('root/user', xmlDom, null,
                                XPathResult.ORDERED_NODE_ITERATOR_TYPE, null);
var nodes = [];
if (result !== null) {
    while ((node = result.iterateNext()) !== null) {
        nodes.push(node);
    }
}
```

PS：节点集合的获取方式，是通过迭代器遍历而来的，我们保存到数据中就模拟出 IE 相似的风格。

三．XPath 跨浏览器兼容

如果要做 W3C 和 IE 的跨浏览器兼容，我们要思考几个问题：1.如果传递一个节点的下标，IE 是从 0 开始计算，W3C 从 1 开始计算，可以通过传递获取下标进行增 1 减 1 的操作来进行。2.独有的功能放弃，为了保证跨浏览器。3.只获取单一节点和节点列表即可，基本可以完成所有的操作。

```
//跨浏览器获取单一节点
function selectSingleNode(xmlDom, xpath) {
    var node = null;

    if (typeof xmlDom.evaluate != 'undefined') {
        var patten = /\[(\d+)\]/g;
        var flag = xpath.match(patten);
        var num = 0;
        if (flag !== null) {
            num = parseInt(RegExp.$1) + 1;
            xpath = xpath.replace(patten, '[' + num + ']');
        }
        var result = xmlDom.evaluate(xpath, xmlDom, null,
                                    XPathResult.FIRST_ORDERED_NODE_TYPE, null);
        if (result !== null) {
            node = result.singleNodeValue;
        }
    } else if (typeof xmlDom.selectSingleNode != 'undefined') {
        node = xmlDom.selectSingleNode(xpath);
    }

    return node;
}
```

```
//跨浏览器获取节点集合
function selectNodes(xmlDom, xpath) {
    var nodes = [];

    if (typeof xmlDom.evaluate != 'undefined') {
        var patten = /\[(\d+)\]/g;
        var flag = xpath.match(patten);
        var num = 0;
        if (flag !== null) {
            num = parseInt(RegExp.$1) + 1;
            xpath = xpath.replace(patten, '[' + num + ']');
        }
        var node = null;
        var result = xmlDom.evaluate('root/user', xmlDom, null,
                                XPathResult.ORDERED_NODE_ITERATOR_TYPE, null);
      if (result !== null) {
        while ((node = result.iterateNext()) !== null) {
            nodes.push(node);
        }
      }
    } else if (typeof xmlDom.selectNodes != 'undefined') {
        nodes = xmlDom.selectNodes(xpath);
    }

    return nodes;
}
```

PS：在传递 xpath 路径时，没有做验证判断是否合法，有兴趣的同学可以自行完成。在 XML 还有一个重要章节是 XSLT 和 EX4，由于在使用频率的缘故，我们暂且搁置。

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(yc60.com)联合提供：

本次主讲老师：李炎恢

我的博客：hi.baidu.com/李炎恢/

我的邮件：yc60.com@gmail.com