

University at Buffalo
Spring 2011

CSE 586 - Distributed Systems

xml parsing with dom4j



Hanifi Gunes
PhD Candidate
Computer Science & Engineering
hanifigu{at}buffalo{dot}edu

In this tutorial we will talk about XML parsing with dom4j, an easy to use, open source library working with XML, XPath and XSLT on the Java platform. So let's get started.

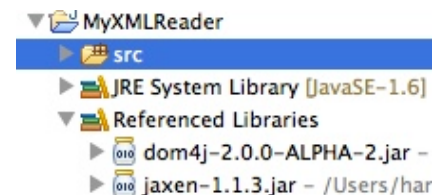
① Download and install eclipse for Java developers: <http://www.eclipse.org>

Download the dom4j jar package: <http://dom4j.sourceforge.net>

Download the jaxen jar package: <http://jaxen.codehaus.org>

② Run eclipse and locate your workspace directory if you are running it for the first time. Now, create a new Java Project, MyXMLReader, from File → New → Java Project. Notice that the project is created under the workspace directory `{workspace_dir}/MyXMLReader`.

③ Right click on your project and select Build Path → Add External Archives. Locate and add the dom4j jar archive from step 1 and repeat the same procedure to add the jaxen package to your build path. At that point, you should see these packages listed under *Referenced Libraries* as it is on the left snapshot.



④ Now that, the project setup is done we can work with Yahoo! Weather. Visit <http://developer.yahoo.com/weather/> to get familiar with the XML response schema. First thing to observe is that you can query Yahoo Weather for a particular zip code using the following pattern <http://weather.yahooapis.com/forecastrss?p=ZIPCODE>. Try substituting your own zip code into the address pattern and then navigate to that address from your browser in order to inspect the XML response.

⑤ Assuming that we are interested in retrieving the *city* and *region* name of a particular zip code. We need to navigate to `yweather:location` node and then to access the *city* and *region* attributes. To do that we will use XPATHs (<http://www.w3schools.com/xpath/>). You can access the *city* and *region* attributes using the following XPATHs respectively: `/rss/channel/yweather:location/@city` and `/rss/channel/yweather:location/@region`

Make sure that you are comfortable with XPATHs above before gearing up.

⑥ Now that, right click on your project and create a new java file, MyParser, from the New → Class menu. Check *public static void main(...)* option from the dialog. This should automatically generate the main method. Now copy and paste the following code segment into your main method and examine it.

```
String addr = "http://weather.yahooapis.com/forecastrss?p=14221";
URL myURL = new URL(addr);
SAXReader xmlReader = new SAXReader();
Document feed = xmlReader.read(myURL);
String city = feed.valueOf("/rss/channel/yweather:location/@city");
System.out.println(city);
```

⑦ This code first creates an instance of *java.net.URL* class to access Yahoo! Weather and then feeds it into *xmlReader.read* method so as to retrieve the XML response from Yahoo and return a *Document* instance. Then the *feed.valueOf* method above yields the value of a particular attribute that is described with an XPATH parameter - in which case *city*.

⑧ Save your java file and click on *Run*. You need to see *Buffalo* printed out on the console. Now revisit the Yahoo! Weather Developer Network and exercise on the other attributes of interest. Keep in mind that *Document.valueOf* method returns String type. So you might need to do some type-conversion as for integer and date typed attributes.