**Parsing XML in Java**

This document provides instructions of parsing XML in Java. XML stands for eXtensible Markup Language, which is a common format of web service output.

In this example, we will invoke a movie service by sending a HTTP request: <http://www.omdbapi.com/?s=Spider%20Man&r=XML>. The service returns its results as an XML document.

**Step 1: Import built-in SAX Parser:**

**import** javax.xml.parsers.DocumentBuilderFactory;

**import** javax.xml.parsers.DocumentBuilder;

**import** javax.xml.parsers.ParserConfigurationException;

**Step 2: Use a Document Object to Parse the XML result:**

* Create a DocumentBuilderFactory Object and a Document Builder object.

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.*newInstance*();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.parse("http://www.omdbapi.com/?s=Titanic&r=XML");

doc.getDocumentElement().normalize();

* Now doc is the XML document node. An XML document is parsed as a tree. In order to retrieve the nodes (element nodes and text nodes) in the tree, we create a NodeList.

NodeList nList = doc.getElementsByTagName("Movie");

nList contains a list of element nodes where the tag is “Movie”.

* We then retrieve each node in nList and print out the corresponding information.

for (int temp = 0; temp < nList.getLength(); temp++) {

 Node nNode = nList.item(temp);

 if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

 Element eElement = (Element) nNode;

 System.out.println(eElement.getAttribute("Title")+" "+eElement.getAttribute("Year"));

 }

}

Output result:

Spider-Man 2002

The Amazing Spider-Man 2012

Spider-Man 2 2004

Spider-Man 3 2007

The Amazing Spider-Man 2 2014

Spider-Man 1994–1998

The Spectacular Spider-Man 2008–2009

Ultimate Spider-Man 2012–

Spider-Man 2 2004

Spider-Man 2003–