



Generating XML

Crash course on generating XML



What is XML?

XML is a markup language using tags (entities surrounded in < and >).

XML stands for eXtensible Markup Language.

Goals: simplicity, generality and usability

Pros: Machine and human readable, easy for machines to validate and check

Read: <http://www.w3schools.com/xml/> <https://en.wikipedia.org/wiki/XML>

XML data integrity

XML can be checked for validity and wellformedness

Wellformed: Adheres to the XML syntax

Validity: Adheres to some schema (DTD)

Syntax (overview)

Declaration:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Followed by elements:

```
<STUDENT> ← start tag
```

```
<NAME>Donald Duck</NAME> ← child element
```

```
</STUDENT> ← end tag
```

Donald Duck is content, while the tags are markup!

Tags may have attributes

```
<STUDENT studentID="2">
```

```
  <NAME>Beata Bengtsson</NAME>
```

```
</STUDENT>
```

studentID is an attribute whose value is "2"

The top-level tag is sometimes called root element

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<STUDENTS> ← root element!
```

```
  <STUDENT id="1">
```

```
    <NAME>Anders Andersson</NAME>
```

```
  </STUDENT>
```

```
  <STUDENT id="2">
```

```
    <NAME>Beata Bengtsson</NAME>
```

```
  </STUDENT>
```

```
  <STUDENT id="3">
```

```
    <NAME>Charlie Christensen</NAME>
```

```
  </STUDENT>
```

```
  <STUDENT id="4">
```

```
    <NAME>Dick Dale</NAME>
```

```
  </STUDENT>
```

```
  <STUDENT id="5">
```

```
    <NAME>Edward Eriksson</NAME>
```

```
  </STUDENT>
```

```
</STUDENTS>
```

Creating an XML document from Java (finally!)

Sh*tload of imports

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.OutputKeys;

import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.DOMException;
```

Creating the Document

```
DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();  
DocumentBuilder docBuilder      = docFactory.newDocumentBuilder();  
Document doc                    = docBuilder.newDocument();
```


Creating the root element

```
Element rootElement = doc.createElement("STUDENTS");  
doc.appendChild(rootElement);
```

Will generate (when we create it):

```
<?xml version="1.0" encoding="UTF-8"?>  
<STUDENTS>  
</STUDENTS>
```

Adding child elements

```
// A treemap with <ID,NAME>
for(String id : map.keySet()){
    Element student = doc.createElement("STUDENT");
    student.setAttribute("id", id);
    Element name     = doc.createElement("NAME");
    name.appendChild(doc.createTextNode(map.get(id)));
    student.appendChild(name);
    rootElement.appendChild(student);
}
```

Transformers - robots in disguise

```
TransformerFactory transformerFactory = TransformerFactory.  
newInstance();  
Transformer transformer = transformerFactory.newTransformer();  
transformer.setOutputProperty(OutputKeys.INDENT, "yes");  
Transformer  
    .setOutputProperty("{http://xml.apache.org/xslt}indent-amount",  
        "2");  
DOMSource source = new DOMSource(doc);  
StreamResult result = new StreamResult(new File("Student.xml"));  
transformer.transform(source, result);
```

The resulting file

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<STUDENTS>
  <STUDENT id="1">
    <NAME>Anders Andersson</NAME>
  </STUDENT>
  <STUDENT id="2">
    <NAME>Beata Bengtsson</NAME>
  </STUDENT>
  <STUDENT id="3">
    <NAME>Charlie Christensen</NAME>
  </STUDENT>
  <STUDENT id="4">
    <NAME>Dick Dale</NAME>
  </STUDENT>
  <STUDENT id="5">
    <NAME>Edward Eriksson</NAME>
  </STUDENT>
</STUDENTS>
```

For reference - The TreeMap with the students

```
Map<String,String> students = new TreeMap<String,String>();  
students.put("1", "Anders Andersson");  
students.put("2", "Beata Bengtsson");  
students.put("3", "Charlie Christensen");  
students.put("4", "Dick Dale");  
students.put("5", "Edward Eriksson");
```