

# DOM for creating XML

Unit - II

# XML creation using Java DOM API

- DOM provide classes to create XML file easily
- Steps:
- Create a Document with **DocumentBuilder** class
- Define all the XML content – node, attribute with **Element** class
- Use **Transformer** class to output the entire XML content to stream output, typically a File

# Example

- To create the below XML:

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

```
<company>
```

```
<staff id="1">
```

```
  <firstname>ABC</firstname>
```

```
<lastname>DEF</lastname>
```

```
  <nickname>mkyong</nickname>
```

```
  <salary>100000</salary>
```

```
</staff>
```

```
</company>
```

# Example Contd...

```
public static void main(String argv[]) {  
    try { DocumentBuilderFactory docFactory =  
        DocumentBuilderFactory.newInstance();  
    DocumentBuilder docBuilder = docFactory.newDocumentBuilder();  
  
    // root elements Document doc = docBuilder.newDocument(); Element  
    rootElement = doc.createElement("company");  
    doc.appendChild(rootElement);  
  
    // staff elements Element staff = doc.createElement("Staff");  
    rootElement.appendChild(staff); // set attribute to staff element  
  
    Attr attr = doc.createAttribute("id");  
    attr.setValue("1");  
    staff.setAttributeNode(attr);
```

# Example Contd...

- `// firstname elements`  
`Element firstname =`  
`doc.createElement("firstname");`  
`firstname.appendChild(doc.createTextNode("ABC"));`  
`staff.appendChild(firstname);`  
`..... (repeat for other elements)`
- `// write the content into xml file`  
`TransformerFactory transformerFactory =`  
`TransformerFactory.newInstance();`  
`Transformer transformer = transformerFactory.newTransformer();`

## Contd...

```
DOMSource source = new DOMSource(doc);  
StreamResult result = new StreamResult(new  
    File("C:\\file.xml"));
```

```
// Output to console
```

```
StreamResult result = new StreamResult(System.out);  
transformer.transform(source, result);  
System.out.println("File saved!"); }
```

# XSL Formatting Objects (XSL-FO)

- XSL-FO designed to assist in printing and displaying of XML data
- The main emphasis is on the document layout and structure
- This includes dimensions of the output document (page headers, footers, and margins)
- XSL-FO also allows the developer to define the formatting rules for the content, such as font, style, color, and positioning
- XSL-FO is a sophisticated version of Cascading Style Sheets (CSS)
- XSL-FO borrows lot of the terminology and elements from CSS
- Ex. For XSL FO engine – Apache XSL FOP, XEP, iText, Unicorn
- Mostly used in generating PDF