

SimpleXMLReport.java

```
1
2 import com.cete.dynamicpdf.*;
3 import com.cete.dynamicpdf.pageelements.Label;
4 import com.cete.dynamicpdf.pageelements.Line;
5 import com.cete.dynamicpdf.pageelements.PageNumberingLabel;
6 import com.cete.dynamicpdf.pageelements.Rectangle;
7 import java.io.File;
8 import java.io.IOException;
9 import java.text.SimpleDateFormat;
10 import java.util.Date;
11 import java.util.GregorianCalendar;
12 import javax.servlet.ServletConfig;
13 import javax.servlet.ServletException;
14 import javax.servlet.ServletOutputStream;
15 import javax.servlet.http.HttpServlet;
16 import javax.servlet.http.HttpServletRequest;
17 import javax.servlet.http.HttpServletResponse;
18 import javax.xml.parsers.DocumentBuilder;
19 import javax.xml.parsers.DocumentBuilderFactory;
20 import javax.xml.parsers.ParserConfigurationException;
21 import org.w3c.dom.Document;
22 import org.w3c.dom.Element;
23 import org.w3c.dom.Node;
24 import org.w3c.dom.NodeList;
25 import org.xml.sax.SAXException;
26
27
28
29 public class SimpleXMLReport extends HttpServlet {
30
31     // Template for document elements
32     private Template template;
33
34     // Page Dimensions of pages
35     private PageDimensions pageDimensions;
36     // Current page that elements are being added to
37     private Page currentPage;
38     // Top Y coordinate for the body of the report
39     private float bodyTop;
40     // Bottom Y coordinate for the body of the report
41     private float bodyBottom;
42     // Current Y coordinate where elements are being added
43     private float currentY;
44     // Used to control the alternating background
45     private boolean alternateBG;
46
47     ServletOutputStream sOut;
48
49     public void init(ServletConfig config) throws ServletException {
```

SimpleXMLReport.java

```
50     super.init(config);
51     pageDimensions = new PageDimensions(PageSize
52                                     .LETTER, PageOrientation.PORTRAIT, 54.0f);
53     bodyTop = 38;
54     bodyBottom = pageDimensions.getBody().getBottom() - pageDimensions.getBody().getTop();
55 }
56
57 public void doGet(HttpServletRequest req, HttpServletResponse res)
58     throws IOException,ServletException {
59
60     sOut = res.getOutputStream();
61     currentY = 0;
62     alternateBG = false;
63     template = new Template();
64     // Create a document and set it's properties
65     com.cete.dynamicpdf.Document objDocument = new com.cete.dynamicpdf
66                                     .Document();
67     objDocument.setCreator("SimpleXMLReport.java");
68     objDocument.setAuthor("ceTe Software");
69     objDocument.setTitle("Simple XML Report");
70
71     DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
72     DocumentBuilder docBuilder = null;
73     try {
74         docBuilder = factory.newDocumentBuilder();
75     } catch(ParserConfigurationException ex1) {
76         System.err.println("ERROR OCCURED WHILE BUILDING DOCUMENT " +ex1
77                             .getMessage());
78     }
79     Document doc = null;
80     try {
81         doc = docBuilder.parse(new File(getServletContext()
82                                     .getRealPath("/data/ProductList.xml")));
83     } catch (SAXException ex2) {
84         System.err.println("ERROR OCCURED WHILE PARSING "+ex2.getMessage());
85     } catch (IOException ex3) {
86         System.err.println("IOException occured "+ex3.getMessage());
87     }
88
89     // Adds elements to the header template
90     setTemplate();
91     // Add the template to the document
92     objDocument.setTemplate(template);
93
94     // Builds the report
95     buildDocument(objDocument, doc);
96     // Outputs the SimpleXMLReport to the current web page
97     objDocument.drawToWeb(req, res, sOut, "SimpleXMLReport.pdf");
98     sOut.close();
```

```
99
100 }
101
102 private void setTemplate() {
103     // Adds elements to the header template
104     GregorianCalendar gc = new GregorianCalendar();
105     Date date = gc.getTime();
106     SimpleDateFormat sdl = new SimpleDateFormat("dd MMM yyyy, H:mm:ss E");
107     template.getElements().add(new Label(sdl.format(date), 0, 0, 504, 12,
108         Font.getHelveticaBold(), 12));
109     template.getElements().add(new Label("Northwind Product List", 0, 0,
110         504, 12, Font.getHelveticaBold(), 12,
111         TextAlign.CENTER));
112     PageNumberingLabel pageNumLabel = new PageNumberingLabel("Page %%CP%%" +
113         " of %%TP%%", 0, 0, 504, 12, Font.getHelveticaBold(), 12,
114         TextAlign.RIGHT);
115     template.getElements().add(pageNumLabel);
116     template.getElements().add(new Label("Product", 2, 23, 236, 11,
117         Font.getTimesBold(), 11));
118     template.getElements().add(new Label("Qty Per Unit", 242, 23, 156, 11,
119         Font.getTimesBold(), 11));
120     template.getElements().add(new Label("Unit Price", 402, 23, 100, 11,
121         Font.getTimesBold(), 11, TextAlign.RIGHT));
122     template.getElements().add(new Line(0, 36, 504, 36));
123 }
124
125 private void buildDocument(com.cete.dynamicpdf.Document document, Document doc) {
126     // Builds the PDF document with data from the XML Data
127     addNewPage(document);
128     //doc.getDocumentElement().normalize();
129
130     NodeList list = doc.getElementsByTagName("Products");
131
132     int length = list.getLength();
133     for (int i = 0; i < length; i++) {
134         Element element = (Element)list.item(i);
135         if (element.getNodeType() == Node.ELEMENT_NODE
136             && element.getNodeName().equals("Products")) {
137             addRecord(document, element);
138         }
139     }
140 }
141
142 private void addRecord(com.cete.dynamicpdf.Document document,
143     Element element) {
144     // Adds a new page to the document if needed
145     if (currentY > bodyBottom) {
146         addNewPage(document);
147     }
148 }
```

```
148
149 // Adds alternating background to document if needed
150 if (alternateBG) {
151     currentPage.getElements().add(new Rectangle(0, currentY, 504, 18,
152         new WebColor("E0E0FF"), new WebColor("E0E0FF")));
153 }
154
155 // Adds Labels to the document with data from the current node
156 currentPage.getElements().add(new Label(element
157     .getAttribute("ProductName"), 2, currentY + 3,
158     236, 11, Font.getTimesRoman(), 11));
159 currentPage.getElements().add(new Label(element
160     .getAttribute("QuantityPerUnit"), 242,
161     currentY + 3, 156, 11, Font.getTimesRoman(),
162     11));
163 currentPage.getElements().add(new Label(element
164     .getAttribute("UnitPrice"), 402, currentY + 3,
165     100, 11, Font.getTimesRoman(), 11,
166     TextAlign.RIGHT));
167
168 // Toggles alternating background
169 alternateBG = !alternateBG;
170 // Increments the current Y position on the page
171 currentY += 18;
172 }
173
174 private void addNewPage(com.cete.dynamicpdf.Document document) {
175     // Adds a new page to the document
176     currentPage = new Page(pageDimensions);
177
178     currentY = bodyTop;
179     alternateBG = false;
180
181     document.getPages().add(currentPage );
182 }
183
184
185 }
```