

```
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.IOException;
8 import java.sql.*;
9 import java.text.DecimalFormat;
10 import java.text.SimpleDateFormat;
11 import java.util.Date;
12 import java.util.GregorianCalendar;
13 import javax.servlet.ServletConfig;
14 import javax.servlet.ServletException;
15 import javax.servlet.ServletOutputStream;
16 import javax.servlet.http.HttpServlet;
17 import javax.servlet.http.HttpServletRequest;
18 import javax.servlet.http.HttpServletResponse;
19
20
21 public class SimpleReport extends HttpServlet {
22
23     // Template for document elements
24     private Template template;
25
26     // Page Dimensions of pages
27     private PageDimensions pageDimensions;
28     // Current page that elements are being added to
29     private Page currentPage;
30     // Top Y coordinate for the body of the report
31     private float bodyTop;
32     // Bottom Y coordinate for the body of the report
33     private float bodyBottom;
34     // Current Y coordinate where elements are being added
35     private float currentY;
36     // Used to control the alternating background
37     private boolean alternateBG;
38
39     ServletOutputStream sOut;
40
41     Connection connection;
42
43     public void init(ServletConfig servletConfig) throws ServletException {
44         super.init(servletConfig);
45         pageDimensions = new PageDimensions(PageSize.LETTER,
46                                             PageOrientation.PORTRAIT, 54.0f);
47         bodyTop = 38;
48         bodyBottom = pageDimensions.getBody().getBottom() - pageDimensions.getBody().getTop();
49     }
```

```
50
51 }
52
53 public void doGet(HttpServletRequest req, HttpServletResponse res)
54     throws IOException, ServletException {
55     CeTeConnection ceTe = (CeTeConnection)getContext().getAttribute("cetecon");
56     connection = ceTe.getConnection();
57     sOut = res.getOutputStream();
58     alternateBG = false;
59     currentY = 0;
60     template = new Template();
61
62     // Create a document and set it's properties
63     Document objDocument = new Document();
64     objDocument.setCreator("SimpleReport.java");
65     objDocument.setAuthor("ceTe Software");
66     objDocument.setTitle("Simple Report");
67
68     // Adds elements to the header template
69     setTemplate();
70     objDocument.setTemplate(template);
71
72     ResultSet data = getContactListData();
73     // Builds the report
74     buildDocument(objDocument, data);
75
76     // Outputs the SimpleReport to the current web page
77     objDocument.drawToWeb(req, res, sOut, "SimpleReport.pdf");
78     ceTe.close();
79     sOut.close();
80
81 }
82
83 private void setTemplate() {
84     GregorianCalendar gc = new GregorianCalendar();
85     Date date = gc.getTime();
86     SimpleDateFormat sdl = new SimpleDateFormat("dd MMM yyyy, H:mm:ss E");
87     // Adds elements to the header template
88     template.getElements().add(new Label(sdl.format(date), 0, 0, 504, 12,
89         Font.getHelveticaBold(), 12));
90     template.getElements().add(new Label("Northwind Product List", 0, 0,
91         504, 12, Font.getHelveticaBold(), 12,
92         TextAlign.CENTER));
93     PageNumberingLabel pageNumLabel = new PageNumberingLabel("Page %%CP%%" +
94         " of %%TP%%", 0, 0, 504, 12, Font.getHelveticaBold(), 12,
95         TextAlign.RIGHT);
96     template.getElements().add(pageNumLabel);
97     template.getElements().add(new Label("Product", 2, 23, 236, 11,
98         Font.getTimesBold(), 11));
```

```
99     template.getElements().add(new Label("Qty Per Unit", 242, 23, 156, 11,
100         Font.getTimesBold(), 11));
101     template.getElements().add(new Label("Unit Price", 402, 23, 100, 11,
102         Font.getTimesBold(), 11, TextAlign.RIGHT));
103     template.getElements().add(new Line(0, 36, 504, 36));
104
105 }
106
107 private void buildDocument(Document document, ResultSet data) {
108     // Builds the PDF document with data from the ResultSet
109     addNewPage(document);
110     try {
111         while (data.next()) {
112             // Add current record to the document
113             addRecord(document, data);
114         }
115     } catch (SQLException ex) {
116         System.err.println("cannot access the ResultSet :"+ex.getMessage());
117     }
118 }
119
120 private void addRecord(Document document, ResultSet data) {
121     // Adds a new page to the document if needed
122     if (currentY > bodyBottom) {
123         addNewPage(document);
124     }
125
126     // Adds alternating background to document if needed
127     if (alternateBG) {
128         currentPage.getElements().add(new Rectangle(0, currentY, 504, 18,
129             new WebColor("E0E0FF"), new WebColor("E0E0FF")));
130     }
131     try {
132         // Adds Labels to the document with data from current record
133         currentPage.getElements().add(new Label(data.getString(1), 2,
134             currentY + 3, 236, 11, Font.getTimesRoman(),
135             11));
136         currentPage.getElements().add(new Label(data.getString(2), 242,
137             currentY + 3, 156, 11,
138             Font.getTimesRoman(), 11));
139         DecimalFormat d1 = new DecimalFormat("0.00");
140         currentPage.getElements().add(new Label(d1.format(data.getBigDecimal
141             (3).doubleValue()), 402, currentY + 3, 100, 11,
142             Font.getTimesRoman(), 11, TextAlign.RIGHT));
143     } catch (SQLException ex) {
144         System.err.println("cannot retrieve data from ResultSet :"+
145             ex.getMessage());
146     }
147 }
```

```
148 // Toggles alternating background
149 alternateBG = !alternateBG;
150 // Increments the current Y position on the page
151 currentY += 18;
152 }
153
154 private void addNewPage(Document document) {
155 // Adds a new page to the document
156 currentPage = new Page(pageDimensions);
157
158 currentY = bodyTop;
159 alternateBG = false;
160
161 document.getPages().add(currentPage);
162 }
163
164 private ResultSet getContactListData() {
165
166     ResultSet data = null;
167     try {
168 // Creates a ResultSet for the report
169 PreparedStatement ps = connection.prepareStatement("SELECT " +
170                                                     "ProductName, QuantityPerUnit, UnitPrice " +
171                                                     "FROM Products");
172
173     data = ps.executeQuery();
174     } catch (SQLException ex) {
175         System.err.println("cannot execute the query :"+ex.getMessage());
176     }
177
178     return data;
179 }
180 }
```