

## ContactList.java

---

```
1
2 import com.cete.dynamicpdf.*;
3 import com.cete.dynamicpdf.pageelements.*;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6 import java.sql.*;
7 import java.text.SimpleDateFormat;
8 import java.util.Date;
9 import java.util.GregorianCalendar;
10 import javax.servlet.ServletConfig;
11 import javax.servlet.ServletException;
12 import javax.servlet.ServletOutputStream;
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16
17
18
19 public class ContactList extends HttpServlet {
20
21     // Top, bottom, left and right margins of report
22     private float margin;
23     // Height of the header
24     private float headerHeight;
25     // Height of the footer
26     private float footerHeight;
27     // Size of paper to use
28     private PageDimensions pageSize;
29     // Bottom Y coordinate for the body of the report
30     private float bodyBottom;
31
32     // Current page that elements are being added to
33     private Page currentPage;
34     // Template for header and footer elements
35     private Template template;
36     // Current Y coordinate where elements are being added
37     private float currentY;
38     // Used to control the alternating background
39     private boolean alternateBG;
40     // Used to test for grouping
41     private String tmpChar = "";
42
43     ServletOutputStream sOut;
44     Connection connection;
45
46     public void init(ServletConfig servletConfig) throws ServletException {
47         super.init(servletConfig);
48         margin = 36;
49         headerHeight = 74;
```

## ContactList.java

---

```
50     footerHeight = 14;
51     pageSize = new PageDimensions(PageSize.LETTER,
52                                   PageOrientation.LANDSCAPE, margin);
53     bodyBottom = pageSize.getHeight() - (margin * 2) - footerHeight;
54
55
56 }
57 public void doGet(HttpServletRequest req, HttpServletResponse res)
58     throws IOException,ServletException {
59     CeTeConnection ceTe = (CeTeConnection)getServletContext().getAttribute("cetecon");
60     connection = ceTe.getConnection();
61     sOut = res.getOutputStream();
62     currentY = 0;
63     alternateBG = false;
64     template = new Template();
65
66     // Create a document and set it's properties
67     Document objDocument = new Document();
68     objDocument.setCreator("ContactList.java");
69     objDocument.setAuthor("ceTe Software");
70     objDocument.setTitle("Contact List");
71     objDocument.setTemplate(template);
72
73     // Adds elements to the template
74     setHeaderTemplate();
75     setFooterTemplate();
76     ResultSet data = getContactListData();
77
78     // Builds the report
79     buildDocument(objDocument, data);
80
81     // Outputs the ContactList to the current web page
82     objDocument.drawToWeb(req, res, sOut, "ContactList.pdf");
83     ceTe.close();
84     sOut.close();
85 }
86
87 private void setHeaderTemplate() {
88     // Adds header elements to the template
89     try {
90         template.getElements().add(new Image(getServletContext()
91                                             .getRealPath("images/DPDFLogo.png"), 0, 0, 0.21f));
92     } catch (FileNotFoundException ex) {
93         ex.printStackTrace(System.err);
94     }
95     template.getElements().add(new Label("Northwind Traders", 0, 0, 720, 18,
96                                         Font.getHelveticaBold(), 18, TextAlign.CENTER));
97     template.getElements().add(new Label("Contact List", 0, 21, 720, 12,
98                                         Font.getHelvetica(), 12, TextAlign.CENTER));
```

```
99     GregorianCalendar gc = new GregorianCalendar();
100     Date date = gc.getTime();
101     SimpleDateFormat sd1 = new SimpleDateFormat("dd MMM yyyy, H:mm:ss E");
102     template.getElements().add(new Label(sd1.format(date), 0, 36, 720, 12,
103         Font.getHelvetica(), 12, TextAlign.CENTER));
104     template.getElements().add(new Rectangle(0, 56, 720, 16,
105         new WebColor("0000A0"), new WebColor("0000A0")));
106     template.getElements().add(new Label("Cust ID", 2, 57, 58, 12,
107         Font.getHelveticaBold(), 12, TextAlign.LEFT,
108         Grayscale.getWhite()));
109     template.getElements().add(new Label("Company", 62, 57, 156, 12,
110         Font.getHelveticaBold(), 12, TextAlign.LEFT,
111         Grayscale.getWhite()));
112     template.getElements().add(new Label("Contact", 222, 57, 156, 12,
113         Font.getHelveticaBold(), 12, TextAlign.LEFT,
114         Grayscale.getWhite()));
115     template.getElements().add(new Label("Title", 362, 57, 156, 12,
116         Font.getHelveticaBold(), 12, TextAlign.LEFT,
117         Grayscale.getWhite()));
118     template.getElements().add(new Label("Phone", 522, 57, 86, 12,
119         Font.getHelveticaBold(), 12, TextAlign.LEFT,
120         Grayscale.getWhite()));
121     template.getElements().add(new Label("Fax", 622, 57, 86, 12,
122         Font.getHelveticaBold(), 12, TextAlign.LEFT,
123         Grayscale.getWhite()));
124 }
125
126 private void setFooterTemplate() {
127     // Adds footer elements to the template
128     PageNumberingLabel pageNumLabel = new PageNumberingLabel("Page " +
129         "%CP(i)% of %TP(i)%", 0, bodyBottom + 5, 720, 10,
130         Font.getHelvetica(), 10, TextAlign.CENTER);
131     template.getElements().add(pageNumLabel);
132 }
133
134 private void buildDocument(Document document, ResultSet data) {
135     // Builds the PDF document with data from the ResultSet
136     addNewPage(document);
137     try {
138         while (data.next()) {
139             // Add current record to the document
140             addRecord(document, data);
141         }
142     } catch (SQLException ex) {
143         ex.printStackTrace(System.err);
144     }
145 }
146
147 private void addRecord(Document document, ResultSet data) {
```

```
148     // Creates TextAreas that are expandable
149     try {
150
151         String tmpCustomerId = data.getString(1);
152         String tmpStr = data.getString(2);
153
154         TextArea companyName = new TextArea(data.getString(3), 62,
155                                             currentY + 3, 156, 11,
156                                             Font.getTimesRoman(), 11);
157         TextArea contactName = new TextArea(data.getString(4), 222,
158                                             currentY + 3, 136, 11,
159                                             Font.getTimesRoman(), 11);
160         TextArea contactTitle = new TextArea(data.getString(5), 362,
161                                             currentY + 3, 156, 11,
162                                             Font.getTimesRoman(), 11);
163
164
165
166
167         float requiredHeight = setExpandableRecords(document,
168                                                     companyName,contactName,
169                                                     contactTitle,tmpStr);
170
171
172         // Creates non expandable Labels
173         Label customerID = new Label(tmpCustomerId, 2, currentY + 3,
174                                     58, 11,Font.getTimesRoman(), 11);
175
176
177         Label phone = new Label(data.getString(6), 522, currentY + 3, 96, 11,
178                                 Font.getTimesRoman(), 11);
179         String str = data.getString(7);
180         Label fax = new Label(data.isNull()?":str", 622, currentY + 3, 96,
181                               11, Font.getTimesRoman(), 11);
182
183         // Adds alternating background if required
184         if (alternateBG) {
185             currentPage.getElements().add(new Rectangle(0, currentY, 720,
186                                                         requiredHeight + 6, new WebColor("E0E0FF"),
187                                                         new WebColor("E0E0FF")));
188         }
189         // Toggles alternating background
190         alternateBG = !alternateBG;
191
192         // Adds elements to the current page
193         currentPage.getElements().add(customerID);
194         currentPage.getElements().add(companyName);
195         currentPage.getElements().add(contactName);
196         currentPage.getElements().add(contactTitle);
```

```
197     currentPage.getElements().add(phone);
198     currentPage.getElements().add(fax);
199
200     // increments the current Y position on the page
201     currentY += requiredHeight + 6;
202     } catch (SQLException ex) {
203         ex.printStackTrace(System.err);
204     }
205 }
206
207 private float setExpandableRecords(Document document,
208     TextArea companyName, TextArea contactName,
209     TextArea contactTitle, String tmpStr) {
210     // Gets the maximum height required of the three TextAreas
211     float requiredHeight = getMaxRecordHeight(companyName, contactName,
212         contactTitle);
213
214     // Add space for the section header if required
215     float sectionHeaderHeight = 0;
216     if (!tmpChar.equals(tmpStr.substring(0,1))) {
217         sectionHeaderHeight = 26;
218     }
219     // Add a new page if needed
220     if (bodyBottom < currentY + requiredHeight + sectionHeaderHeight + 4) {
221         addNewPage(document);
222         if (sectionHeaderHeight == 0) {
223             // Update Y coordinate of TextArea when placed on the new page
224             companyName.setY(currentY + 1);
225             contactName.setY(currentY + 1);
226             contactTitle.setY(currentY + 1);
227         }
228     }
229
230     if (sectionHeaderHeight > 0) {
231         addSectionHeader(tmpStr);
232         companyName.setY(currentY + 3);
233         contactName.setY(currentY + 3);
234         contactTitle.setY(currentY + 3);
235     }
236
237     return requiredHeight;
238 }
239
240 private void addSectionHeader(String tmpStr) {
241     tmpChar = tmpStr.substring(0, 1);
242     currentPage.getElements().add(new Label("- " + tmpChar + " -", 0,
243         currentY + 6, 720, 18, Font.getHelveticaBold(),
244         18, TextAlign.CENTER));
245 }
```

## ContactList.java

---

```
246     currentY += 26;
247     alternateBG = false;
248 }
249
250 private float getMaxRecordHeight(TextArea companyName, TextArea contactName,
251                                 TextArea contactTitle) {
252     // Returns the maximum required height of the three TextAreas
253     float requiredHeight = 11;
254     float requiredHeightB = 0;
255
256     requiredHeight = companyName.getRequiredHeight();
257     requiredHeightB = contactName.getRequiredHeight();
258     if (requiredHeightB > requiredHeight) {
259         requiredHeight = requiredHeightB;
260     }
261     requiredHeightB = contactTitle.getRequiredHeight();
262     if (requiredHeightB > requiredHeight) {
263         requiredHeight = requiredHeightB;
264     }
265
266     if (requiredHeight > 11) {
267         companyName.setHeight(requiredHeight);
268         contactName.setHeight(requiredHeight);
269         contactTitle.setHeight(requiredHeight);
270     }
271
272     return requiredHeight;
273 }
274
275
276 private void addNewPage(Document document) {
277     // Adds a new page to the document
278     currentPage = new Page(pageSize);
279
280     currentY = headerHeight;
281     alternateBG = false;
282
283     document.getPages().add(currentPage);
284 }
285
286 private ResultSet getContactListData() {
287     ResultSet data = null;
288     // Creates a ResultSet for the report
289     try {
290         PreparedStatement ps = connection.prepareStatement("SELECT Customer"+
291                                                         "ID, FirstL = Left(CompanyName,1), Company"+
292                                                         "Name, ContactName, ContactTitle, Phone, "+
293                                                         "Fax FROM Customers ORDER BY CompanyName");
294     }
```

## ContactList.java

---

```
295     data = ps.executeQuery();
296     } catch (SQLException ex) {
297         ex.printStackTrace(System.err);
298     }
299     return data;
300 }
301
302
303
304 }
```