

MailingLabels.java

```
1 import com.cete.dynamicpdf.*;
2 import com.cete.dynamicpdf.pageelements.TextArea;
3 import java.io.IOException;
4 import java.sql.Connection;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import javax.servlet.ServletConfig;
9 import javax.servlet.ServletException;
10 import javax.servlet.ServletOutputStream;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14
15
16 public class MailingLabels extends HttpServlet {
17     // Set page dimensions
18     int topMargin;
19     int bottomMargin;
20     float rightMargin;
21     float leftMargin;
22     // Set the number of labels per page
23     int maximumColumns;
24     int maximumRows;
25     // Set the spacing between the labels
26     int horizontalSpace;
27     int verticalSpace;
28     // These margins are on the labels themselves
29     int labelTopBottomMargin;
30     int labelLeftRightMargin;
31
32     Document document;
33     Page page;
34     float currentColumn, currentRow, labelWidth, labelHeight;
35     String companyName;
36     String contactName;
37     String phone;
38     String fax;
39     ServletOutputStream sOut;
40     Connection connection;
41
42     public void init(ServletConfig servletConfig) throws ServletException {
43         super.init(servletConfig);
44         topMargin = 36;
45         bottomMargin = 36;
46         rightMargin = 13.5f;
47         leftMargin = 13.5f;
48         maximumColumns = 3;
49         maximumRows = 10;
```

MailingLabels.java

```
50     horizontalSpace = 9;
51     verticalSpace = 0;
52     labelTopBottomMargin = 5;
53     labelLeftRightMargin = 15;
54 }
55 public void doGet(HttpServletRequest req, HttpServletResponse res)
56     throws IOException, ServletException {
57     CeTeConnection ceTe = (CeTeConnection)getContext().getAttribute("cetecon");
58     connection = ceTe.getConnection();
59     sOut = res.getOutputStream();
60
61     // Create a document and set it's properties
62     document = new Document();
63     page = new Page(PageSize.LETTER, PageOrientation.PORTRAIT);
64     document.setCreator("MailingLabels.java");
65     document.setAuthor("ceTe Software");
66     document.setTitle("Mailing Labels");
67
68     // Entrypoint for the labels
69     currentRow = 1;
70     currentColumn = 1;
71
72     ResultSet data = null;
73     // Creates a ResultSet for the report
74     try {
75         PreparedStatement ps = connection.prepareStatement("Select CompanyName, " +
76             "ContactName, Phone, Fax FROM Customers");
77         data = ps.executeQuery();
78     } catch (SQLException ex1) {
79         ex1.printStackTrace(System.err);
80     }
81
82     // Loop over the ResultSet and add each label
83     try {
84         while (data.next()) {
85             companyName = safeDBNull(data.getString("CompanyName"), data);
86             contactName = safeDBNull(data.getString("ContactName"), data);
87             phone = safeDBNull(data.getString("Phone"), data);
88             fax = safeDBNull(data.getString("Fax"), data);
89             addLabel();
90         }
91     } catch (SQLException ex2) {
92         ex2.printStackTrace(System.err);
93     }
94     if (page.getElements().size() > 0) {
95         document.getPages().add(page);
96     }
97     // Outputs the MailingLabels to the current web page
98     document.drawToWeb(req, res, sOut, "MailingLabels.pdf");
```

```
99     ceTe.close();
100     sOut.close();
101 }
102
103 private float findLabelHeight() {
104     return (page.getDimensions().getHeight() - (page.getDimensions().getTopMargin()
105     + page.getDimensions().getBottomMargin()) - ((maximumRows - 1)
106     * verticalSpace)) / maximumRows;
107 }
108
109 private float findLabelWidth() {
110     return (page.getDimensions().getWidth() - (page.getDimensions().getRightMargin()
111     + page.getDimensions().getLeftMargin()) - ((maximumColumns - 1)
112     * horizontalSpace)) / maximumColumns;
113 }
114
115 private void addLabel() {
116     // Add a new page if you are beyond the maximum Rows
117     if (currentRow == maximumRows + 1) {
118         document.getPages().add(page);
119         currentRow = 1;
120     }
121     // Determines if the the label belongs in the first row or first column of the page
122     if (currentColumn > 1 & currentRow > 1) {
123         addToPage();
124     } else if (currentColumn > 1 & currentRow == 1) {
125         addToFirstRow();
126     } else if (currentColumn == 1 & currentRow > 1) {
127         addToFirstColumn();
128     } else {
129         page = new Page(PageSize.LETTER, PageOrientation.PORTRAIT);
130
131         page.getDimensions().setTopMargin(topMargin);
132         page.getDimensions().setBottomMargin(bottomMargin);
133         page.getDimensions().setRightMargin(rightMargin);
134         page.getDimensions().setLeftMargin(leftMargin);
135         labelWidth = findLabelWidth();
136         labelHeight = findLabelHeight();
137         addToFirstRowColumn();
138     }
139
140     // Increment your row if you are beyond the maximum columns
141     if (currentColumn == maximumColumns + 1) {
142         currentRow = currentRow + 1;
143         currentColumn = 1;
144     }
145 }
146
147 // Adds the label on at least row 2 column 2 of the page
```

MailingLabels.java

```
148     private void addToPage() {
149         float x;
150         float y;
151         y = (currentRow - 1) * (labelHeight + verticalSpace);
152         x = (currentColumn - 1) * (labelWidth + horizontalSpace);
153         addLabelInfo(x, y);
154         currentColumn = currentColumn + 1;
155     }
156
157     // Adds the label on the first row of labels
158     private void addToFirstRow() {
159         float x;
160         float y;
161         y = 0;
162         x = (currentColumn - 1) * (labelWidth + horizontalSpace);
163         addLabelInfo(x, y);
164         currentColumn = currentColumn + 1;
165     }
166
167     // Adds the label to the First column of labels
168     private void addToFirstColumn() {
169         float x;
170         float y;
171         y = (currentRow - 1) * (labelHeight + verticalSpace);
172         x = 0;
173         addLabelInfo(x, y);
174         currentColumn = currentColumn + 1;
175     }
176
177     // Adds only the first label of every page (row 1 column 1)
178     private void addToFirstRowColumn() {
179         float x;
180         float y;
181         y = 0;
182         x = 0;
183         addLabelInfo(x, y);
184         currentColumn = currentColumn + 1;
185     }
186
187     // This is where you format the look of each label
188     private void addLabelInfo(float x, float y) {
189         TextArea txt1 = new TextArea(companyName, x + labelLeftRightMargin,
190             y + labelTopBottomMargin, labelWidth
191             - (labelLeftRightMargin * 2), 11,
192             Font.getTimesRoman(), 11);
193         TextArea txt2 = new TextArea(contactName, x + labelLeftRightMargin,
194             y + labelTopBottomMargin + 12, labelWidth
195             - (labelLeftRightMargin * 2), 11,
196             Font.getTimesRoman(), 11);
```

MailingLabels.java

```
197     TextArea txt3 = new TextArea(phone, x + labelLeftRightMargin,
198         y + labelTopBottomMargin + 24, labelWidth
199         - (labelLeftRightMargin * 2), 11,
200         Font.getTimesRoman(), 11);
201     TextArea txt4 = new TextArea(fax, x + labelLeftRightMargin,
202         y + labelTopBottomMargin + 36, labelWidth
203         - (labelLeftRightMargin * 2), 11,
204         Font.getTimesRoman(), 11);
205
206     page.getElements().add(txt1);
207     page.getElements().add(txt2);
208     page.getElements().add(txt3);
209     page.getElements().add(txt4);
210 }
211
212 private String safeDBNull(String value, ResultSet data) {
213     try {
214         if (data.isNull()) {
215             return "";
216         } else {
217             return value;
218         }
219     } catch (SQLException ex3) {
220         ex3.printStackTrace(System.err);
221     }
222     return "";
223 }
224 }
```