

MergerInvoice.java

```
1 import com.cete.dynamicpdf.Document;
2 import com.cete.dynamicpdf.Font;
3 import com.cete.dynamicpdf.Page;
4 import com.cete.dynamicpdf.TextAlign;
5 import com.cete.dynamicpdf.merger.ImportedPage;
6 import com.cete.dynamicpdf.pageelements.Label;
7 import com.cete.dynamicpdf.pageelements.PageNumberingLabel;
8 import com.cete.dynamicpdf.pageelements.Rectangle;
9 import com.cete.dynamicpdf.pageelements.TextArea;
10 import com.cete.dynamicpdf.pageelements.barcoding.Interleaved25;
11 import java.io.IOException;
12 import java.math.BigDecimal;
13 import java.sql.*;
14 import java.text.DecimalFormat;
15 import java.text.SimpleDateFormat;
16 import java.util.Enumeration;
17 import java.util.Vector;
18 import javax.servlet.ServletConfig;
19 import javax.servlet.ServletException;
20 import javax.servlet.ServletOutputStream;
21 import javax.servlet.http.HttpServlet;
22 import javax.servlet.http.HttpServletRequest;
23 import javax.servlet.http.HttpServletResponse;
24
25 public class MergerInvoice extends HttpServlet {
26
27     ServletOutputStream sOut;
28     Connection connection;
29
30     public void init(ServletConfig servletConfig) throws ServletException {
31         super.init(servletConfig);
32     }
33     public void doGet(HttpServletRequest req, HttpServletResponse res)
34     throws IOException, ServletException {
35         CeTeConnection ceTe = (CeTeConnection)getServletContext()
36             .getAttribute("cetecon");
37         connection = ceTe.getConnection();
38         sOut = res.getOutputStream();
39         // Create a document and set it's properties
40         Document document = new Document();
41         document.setCreator("Invoice.html");
42         document.setAuthor("ceTe Software");
43         document.setTitle("Invoice");
44
45
46
47         String[] invoiceNo = req.getParameterValues("invoiceno");
48         if (invoiceNo != null) {
49             // Add Invoices to the document
```

```
50         for (int i = 0; i < invoiceNo.length; i++) {
51             MyInvoice invoice = new MyInvoice();
52             invoice.draw(connection, document,
53                 Integer.parseInt(invoiceNo[i]));
54         }
55     }
56
57
58     // Outputs the Invoices to the current web page
59     document.drawToWeb(req, res, sOut, "Invoice.pdf");
60     ceTe.close();
61     sOut.close();
62
63 }
64
65 private class MyInvoice {
66
67     private String templatePath;
68     private BigDecimal subTotal = new BigDecimal(0.0);
69     private BigDecimal freight = new BigDecimal(0.0);
70     private float yOffset = 0;
71     Enumeration e1 = null;
72     private boolean templatePathSet = false;
73     private CeteDAO ceteDAO = null;
74
75     public MyInvoice() {
76         // Sets the template files path
77         setTemplatePath();
78     }
79
80     private void setTemplatePath() {
81         // Sets the template path
82         if (!templatePathSet) {
83             templatePath = getServletContext()
84                 .getRealPath("/pdfs/Invoice.pdf");
85             templatePathSet = true;
86         }
87     }
88
89     public void draw(Connection connection, Document document,
90         int invoiceNumber) {
91         // Each Invoice should begin a new section
92         document.getSections().begin();
93         // Gets the Invoice data
94         Vector<CeteDAO> v1 = getInvoiceData(connection, invoiceNumber);
95         e1 = v1.elements();
96
97         // Adds the invoice if there is data
98         if (e1 != null) {
```

```
99         // Draws the invoice data, returns a page object if it is
100         // the last page
101         Page objLastPage = drawInvoiceData(document);
102         // Draws additional pages if necessary
103         while (objLastPage == null) {
104             objLastPage = drawInvoiceData(document);
105         }
106         // Draws the totals to the bottom of the last page of the
107         //Invoice
108         drawTotals(ceteDAO, objLastPage);
109     }
110
111 }
112
113 private Page drawInvoiceData(Document document) {
114     // Tracks if the invoice is finished
115     boolean invoiceFinished = true;
116     // Tracks the y position on the page
117
118     if (e1.hasMoreElements()) {
119         ceteDAO = (CeteDAO)e1.nextElement();
120         // Sets the Freight amount
121         freight = ceteDAO.getFreight();
122     }
123
124     yOffset = 288;
125
126     // Create a page using "Invoice.pdf" as a template
127     Page objPage = new ImportedPage(templatePath, 1);
128     // Add Details to the Invoice
129     drawInvoiceDetails(ceteDAO, objPage);
130     // Add bill to address
131     drawBillTo(ceteDAO, objPage);
132     // Add ship to address
133     drawShipTo(ceteDAO, objPage);
134
135     // Add line items to the Invoice
136     drawLineItem(ceteDAO, objPage);
137
138     while (e1.hasMoreElements()) {
139         // Break if at the bottom of the page
140         if (yOffset >= 666) {
141             invoiceFinished = false;
142             break;
143         }
144         ceteDAO = (CeteDAO)e1.nextElement();
145         // Add line items to the Invoice
146         drawLineItem(ceteDAO, objPage);
147     }
```

```
148
149     // Add the page to the document
150     document.getPages().add(objPage);
151
152     // If Invoice is finished return the page else return null so
153     // another page will be added
154     if (invoiceFinished) {
155         return objPage;
156     } else {
157         objPage.getElements().add(new Label("Continued...", 454, 704, 82,
158             12, Font.getHelvetica(), 12, TextAlign.RIGHT));
159         return null;
160     }
161 }
162
163 private void drawInvoiceDetails(CeteDAO ceteDAO, Page page) {
164
165     // Adds Invoice details to the page
166     page.getElements().add(new Label(String.valueOf(ceteDAO
167         .getOrderID()), 437, 25, 100, 12, Font.getHelvetica(),
168         12));
169     SimpleDateFormat dateFormat = new SimpleDateFormat("d MMM yyyy");
170     page.getElements().add(new Label(dateFormat.format(ceteDAO
171         .getOrderDate()), 437, 39, 100, 12,
172         Font.getHelvetica(), 12));
173     page.getElements().add(new Label(String.valueOf(ceteDAO
174         .getCustomerID()), 437, 53, 100, 12,
175         Font.getHelvetica(), 12));
176     Date date = ceteDAO.getShippedDate();
177     if (date != null) {
178         page.getElements().add(new Label(dateFormat.format(date),
179             437, 67, 100, 12, Font.getHelvetica(), 12));
180     }
181     page.getElements().add(new Label(ceteDAO.getShipperName(),
182         437, 81, 100, 12, Font.getHelvetica(), 12));
183     page.getElements().add(new Interleaved25(String.valueOf(ceteDAO
184         .getOrderID()), 380, 4, 18, false));
185     page.getElements().add(new Rectangle(450, 250, 90, 20, 0.5f));
186     page.getElements().add(new PageNumberingLabel("Page %%SP%% of %%ST%%",
187         450, 253, 90, 20, Font.getHelveticaBold(), 12,
188         TextAlign.CENTER));
189
190 }
191
192 private void drawBillTo(CeteDAO ceteDAO, Page page) {
193
194     // Adds bill to address
195     String billToAddress = ceteDAO.getBillName() + "\n" +
196         ceteDAO.getBillAddress() + "\n" +
```

```
197         ceteDAO.getBillCity() + ", " +
198         ceteDAO.getBillPostalCode() + "\n" +
199         ceteDAO.getBillCountry();
200     page.getElements().add(new TextArea(billToAddress, 28, 139,
201         194, 49, Font.getHelvetica(), 12));
202
203 }
204
205 private void drawShipTo(CeteDAO ceteDAO, Page page) {
206
207     // Adds ship to address
208     String shipToAddress = ceteDAO.getShipName() + "\n" +
209         ceteDAO.getShipAddress() + "\n" +
210         ceteDAO.getShipCity() + ", " +
211         ceteDAO.getShipPostalCode() + "\n" +
212         ceteDAO.getShipCountry();
213
214     page.getElements().add(new TextArea(shipToAddress, 318, 139,
215         194, 49, Font.getHelvetica(), 12));
216
217 }
218
219 private void drawLineItem(CeteDAO ceteDAO, Page page) {
220     // Adds a line item to the invoice
221     BigDecimal quantity = new BigDecimal(0.0);
222     BigDecimal unitPrice = new BigDecimal(0.0);
223     quantity = new BigDecimal(ceteDAO.getQuantity());
224     unitPrice = ceteDAO.getUnitPrice();
225
226     BigDecimal lineTotal = unitPrice.multiply(quantity);
227     subTotal = subTotal.add(lineTotal);
228
229     page.getElements().add(new Label(quantity.toString(), 4,
230         3 + yOffset, 52, 12, Font.getHelvetica(), 12,
231         TextAlign.RIGHT));
232
233     page.getElements().add(new Label(ceteDAO.getProductName(),
234         64, 3 + yOffset, 292, 12, Font.getHelvetica(), 12));
235
236     DecimalFormat df = new DecimalFormat("#,##0.00");
237     page.getElements().add(new Label(df.format(unitPrice
238         .doubleValue()), 364, 3 + yOffset, 82, 12,
239         Font.getHelvetica(), 12, TextAlign.RIGHT));
240     page.getElements().add(new Label(df.format(lineTotal
241         .doubleValue()), 454, 3 + yOffset, 82, 12,
242         Font.getHelvetica(), 12, TextAlign.RIGHT));
243
244     yOffset += 18;
245 }
```

```

246
247     private void drawTotals(CeteDAO ceteDAO, Page page) {
248         // Add totals to the bottom of the Invoice
249         BigDecimal grandTotal = subTotal.add(freight);
250         DecimalFormat df = new DecimalFormat("#,##0.00");
251         page.getElements().add(new Label(df.format(subTotal
252             .doubleValue()), 454, 667, 82, 12, Font.getHelvetica(), 12,
253             TextAlign.RIGHT));
254         page.getElements().add(new Label(df.format(freight
255             .doubleValue()), 454, 685, 82, 12, Font.getHelvetica(), 12,
256             TextAlign.RIGHT));
257         page.getElements().add(new Label(df.format(grandTotal
258             .doubleValue()), 454, 703, 82, 12, Font.getHelvetica(), 12,
259             TextAlign.RIGHT));
260     }
261
262     private Vector<CeteDAO> getInvoiceData(Connection connection,
263         int invoiceNumber) {
264         Vector<CeteDAO> v1 = null;
265         try {
266             PreparedStatement ps = connection.prepareStatement(
267                 "SELECT o.OrderID, o.CustomerID, o.OrderDate, o.ShippedDate, Freight, " +
268                 "o.ShipName, o.ShipAddress, o.ShipCity, o.ShipPostalCode, o.ShipCountry, " +
269                 "c.CompanyName as BillName, c.Address as BillAddress, " +
270                 "c.City as BillCity, c.PostalCode as BillPostalCode, " +
271                 "c.Country as BillCountry, s.CompanyName as ShipperName, " +
272                 "p.ProductName, od.UnitPrice, od.Quantity " +
273                 "FROM Orders o " +
274                 "JOIN Customers c ON " +
275                 "o.CustomerID = c.CustomerID " +
276                 "JOIN Shippers s ON " +
277                 "o.ShipVia = s.ShipperID " +
278                 "JOIN [Order Details] od ON " +
279                 "o.OrderID = od.OrderID " +
280                 "JOIN Products p ON " +
281                 "od.ProductID = p.ProductID " +
282                 "WHERE o.OrderID = ? ");
283
284             ps.setInt(1, invoiceNumber);
285             ResultSet rs = ps.executeQuery();
286             v1 = new Vector<CeteDAO>(1,1);
287             while (rs.next()) {
288                 CeteDAO ceteDAO = new CeteDAO(rs.getInt(1), rs.getString(2),
289                     rs.getDate(3), rs.getDate(4),
290                     rs.getBigDecimal(5), rs.getString(6),
291                     rs.getString(7), rs.getString(8),
292                     rs.getString(9), rs.getString(10),
293                     rs.getString(11), rs.getString(12),
294                     rs.getString(13), rs.getString(14),

```

```
295         rs.getString(15), rs.getString(16),
296         rs.getString(17), rs.getBigDecimal(18),
297         rs.getShort(19));
298         vl.add(ceteDAO);
299     }
300     } catch (SQLException ex1) {
301         ex1.printStackTrace(System.err);
302     }
303     return vl;
304 }
305 }
306
307 class CeteDAO {
308
309     private int orderID;
310     private String customerID;
311     private Date orderDate;
312     private Date shippedDate;
313     private BigDecimal freight;
314     private String shipName;
315     private String shipAddress;
316     private String shipCity;
317     private String shipPostalCode;
318     private String shipCountry;
319     private String billName;
320     private String billAddress;
321     private String billCity;
322     private String billPostalCode;
323     private String billCountry;
324     private String shipperName;
325     private String productName;
326     private BigDecimal unitPrice;
327     private short quantity;
328
329     public CeteDAO(int orderID, String customerID, Date orderDate,
330         Date shippedDate, BigDecimal freight, String shipName,
331         String shipAddress, String shipCity, String shipPostalCode,
332         String shipCountry, String billName, String billAddress,
333         String billCity, String billPostalCode, String billCountry,
334         String shipperName, String productName, BigDecimal unitPrice,
335         short quantity) {
336         this.orderID = orderID;
337         this.customerID = customerID;
338         this.orderDate = orderDate;
339         this.shippedDate = shippedDate;
340         this.freight = freight;
341         this.shipName = shipName;
342         this.shipAddress = shipAddress;
343         this.shipCity = shipCity;
```

```
344         this.shipPostalCode = shipPostalCode;
345         this.shipCountry = shipCountry;
346         this.billName = billName;
347         this.billAddress = billAddress;
348         this.billCity = billCity;
349         this.billPostalCode = billPostalCode;
350         this.billCountry = billCountry;
351         this.shipperName = shipperName;
352         this.productName = productName;
353         this.unitPrice = unitPrice;
354         this.quantity = quantity;
355     }
356
357     public int getOrderID() {
358         return orderID;
359     }
360
361     public String getCustomerID() {
362         return customerID;
363     }
364
365     public Date getOrderDate() {
366         return orderDate;
367     }
368
369     public Date getShippedDate() {
370         return shippedDate;
371     }
372
373     public BigDecimal getFreight() {
374         return freight;
375     }
376
377     public String getShipName() {
378         return shipName;
379     }
380
381     public String getShipAddress() {
382         return shipAddress;
383     }
384
385     public String getShipCity() {
386         return shipCity;
387     }
388
389     public String getShipPostalCode() {
390         return shipPostalCode;
391     }
392
```



```
393     public String getShipCountry() {
394         return shipCountry;
395     }
396
397     public String getBillName() {
398         return billName;
399     }
400
401     public String getBillAddress() {
402         return billAddress;
403     }
404
405     public String getBillCity() {
406         return billCity;
407     }
408
409     public String getBillPostalCode() {
410         return billPostalCode;
411     }
412
413     public String getBillCountry() {
414         return billCountry;
415     }
416
417     public String getShipperName() {
418         return shipperName;
419     }
420
421     public String getProductName() {
422         return productName;
423     }
424
425     public BigDecimal getUnitPrice() {
426         return unitPrice;
427     }
428
429     public short getQuantity() {
430         return quantity;
431     }
432
433
434
435 }
436
437
438
439
440
441 }
```

