

InvoiceMerger.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 InvoiceMerger 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)getContext()
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     unitPrice = unitPrice.setScale(2, BigDecimal.ROUND_HALF_EVEN);
237     lineTotal = lineTotal.setScale(2, BigDecimal.ROUND_HALF_EVEN);
238     DecimalFormat df = new DecimalFormat("#,##0.00");
239     page.getElements().add(new Label(df.format(unitPrice
240         .doubleValue()), 364, 3 + yOffset, 82, 12,
241         Font.getHelvetica(), 12, TextAlign.RIGHT));
242     page.getElements().add(new Label(df.format(lineTotal
243         .doubleValue()), 454, 3 + yOffset, 82, 12,
244         Font.getHelvetica(), 12, TextAlign.RIGHT));
245 }
```

```
246     yOffset += 18;
247 }
248
249 private void drawTotals(CeteDAO ceteDAO, Page page) {
250     // Add totals to the bottom of the Invoice
251     BigDecimal grandTotal = subTotal.add(freight);
252     subTotal = subTotal.setScale(2, BigDecimal.ROUND_HALF_EVEN);
253     freight = freight.setScale(2, BigDecimal.ROUND_HALF_EVEN);
254     grandTotal = grandTotal.setScale(2, BigDecimal.ROUND_HALF_EVEN);
255     DecimalFormat df = new DecimalFormat("#,##0.00");
256     page.getElements().add(new Label(df.format(subTotal
257         .doubleValue()), 454, 667, 82, 12, Font.getHelvetica(), 12,
258         TextAlign.RIGHT));
259     page.getElements().add(new Label(df.format(freight
260         .doubleValue()), 454, 685, 82, 12, Font.getHelvetica(), 12,
261         TextAlign.RIGHT));
262     page.getElements().add(new Label(df.format(grandTotal
263         .doubleValue()), 454, 703, 82, 12, Font.getHelvetica(), 12,
264         TextAlign.RIGHT));
265 }
266
267 private Vector<CeteDAO> getInvoiceData(Connection connection,
268     int invoiceNumber) {
269     Vector<CeteDAO> v1 = null;
270     try {
271         PreparedStatement ps = connection.prepareStatement(
272             "SELECT o.OrderID, o.CustomerID, o.OrderDate, o.ShippedDate, Freight, " +
273             "o.ShipName, o.ShipAddress, o.ShipCity, o.ShipPostalCode, o.ShipCountry, " +
274             "c.CompanyName as BillName, c.Address as BillAddress, " +
275             "c.City as BillCity, c.PostalCode as BillPostalCode, " +
276             "c.Country as BillCountry, s.CompanyName as ShipperName, " +
277             "p.ProductName, od.UnitPrice, od.Quantity " +
278             "FROM Orders o " +
279             "JOIN Customers c ON " +
280             "o.CustomerID = c.CustomerID " +
281             "JOIN Shippers s ON " +
282             "o.ShipVia = s.ShipperID " +
283             "JOIN [Order Details] od ON " +
284             " o.OrderID = od.OrderID " +
285             "JOIN Products p ON " +
286             "od.ProductID = p.ProductID " +
287             "WHERE o.OrderID = ? ");
288
289         ps.setInt(1, invoiceNumber);
290         ResultSet rs = ps.executeQuery();
291         v1 = new Vector<CeteDAO>(1,1);
292         while (rs.next()) {
293             CeteDAO ceteDAO = new CeteDAO(rs.getInt(1), rs.getString(2),
294                 rs.getDate(3), rs.getDate(4),
```

```
295         rs.getBigDecimal(5), rs.getString(6),
296         rs.getString(7), rs.getString(8),
297         rs.getString(9), rs.getString(10),
298         rs.getString(11), rs.getString(12),
299         rs.getString(13), rs.getString(14),
300         rs.getString(15), rs.getString(16),
301         rs.getString(17), rs.getBigDecimal(18),
302         rs.getShort(19));
303     vl.add(ceteDAO);
304     }
305     } catch (SQLException ex1) {
306         ex1.printStackTrace(System.err);
307     }
308     return vl;
309 }
310 }
311
312 class CeteDAO {
313
314     private int orderID;
315     private String customerID;
316     private Date orderDate;
317     private Date shippedDate;
318     private BigDecimal freight;
319     private String shipName;
320     private String shipAddress;
321     private String shipCity;
322     private String shipPostalCode;
323     private String shipCountry;
324     private String billName;
325     private String billAddress;
326     private String billCity;
327     private String billPostalCode;
328     private String billCountry;
329     private String shipperName;
330     private String productName;
331     private BigDecimal unitPrice;
332     private short quantity;
333
334     public CeteDAO(int orderID, String customerID, Date orderDate,
335         Date shippedDate, BigDecimal freight, String shipName,
336         String shipAddress, String shipCity, String shipPostalCode,
337         String shipCountry, String billName, String billAddress,
338         String billCity, String billPostalCode, String billCountry,
339         String shipperName, String productName, BigDecimal unitPrice,
340         short quantity) {
341         this.orderID = orderID;
342         this.customerID = customerID;
343         this.orderDate = orderDate;
```

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



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

```
442  
443  
444  
445  
446 }  
447
```