

## Invoice.java

---

```
1 import com.cete.dynamicpdf.*;
2 import com.cete.dynamicpdf.pageelements.*;
3 import com.cete.dynamicpdf.pageelements.barcoding.Interleaved25;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6 import java.math.BigDecimal;
7 import java.sql.*;
8 import java.text.DecimalFormat;
9 import java.text.SimpleDateFormat;
10 import java.util.Enumeration;
11 import java.util.Locale;
12 import java.util.Vector;
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 public class Invoice extends HttpServlet {
21
22     ServletOutputStream sOut;
23     Connection connection;
24
25     public void init(ServletConfig servletConfig) throws ServletException {
26         super.init(servletConfig);
27     }
28     public void doGet(HttpServletRequest req, HttpServletResponse res)
29     throws IOException,ServletException {
30         CeTeConnection ceTe = (CeTeConnection)getServletContext().getAttribute("cetecon");
31         connection = ceTe.getConnection();
32         sOut = res.getOutputStream();
33         // Create a document and set it's properties
34         Document objDocument = new Document();
35         objDocument.setCreator("Invoice.html");
36         objDocument.setAuthor("ceTe Software");
37         objDocument.setTitle("Invoice");
38
39
40
41         String[] invoiceNo = req.getParameterValues("invoiceno");
42         if (invoiceNo != null) {
43             // Add Invoices to the document
44             for (int i = 0; i < invoiceNo.length; i++) {
45                 MyInvoice objInvoice = new MyInvoice();
46                 //Add the template to the document
47                 objDocument.setTemplate(objInvoice.getTemplate());
48                 objInvoice.draw(connection, objDocument,
49                     Integer.parseInt(invoiceNo[i]));
```

```
50     }
51 }
52
53
54 // Outputs the Invoices to the current web page
55 objDocument.drawToWeb(req, res, sOut, "Invoice.pdf");
56 ceTe.close();
57 sOut.close();
58
59 }
60
61 private class MyInvoice {
62
63     private BigDecimal subTotal = new BigDecimal(0.0);
64     private BigDecimal freight = new BigDecimal(0.0);
65     private float yOffset = 0;
66     Enumeration e1 = null;
67     private Template template = new Template();
68     private boolean pageTemplateImagesSet = false;
69     private RgbColor objBGColor = new WebColor("#E0E0FF");
70     private WebColor objTotalBGColor = new WebColor("#FFC0C0");
71     private WebColor objThankYouText = new WebColor("#000080");
72     private CeteDAO ceteDAO = null;
73
74     public MyInvoice() {
75         // Top part of Invoice
76         template.getElements().add(new Label("Northwind Traders\n1234 " +
77             "International Drive\nAnywhere, Earth ABC123", 56, 0, 200, 44,
78             Font.getHelvetica()));
79         template.getElements().add(new Label("Invoice", 0, 0, 540, 18,
80             Font.getHelveticaBold(), 18, TextAlign.RIGHT));
81
82         template.getElements().add(new PageNumberingLabel("Page %%SP%% of %%ST%% ",
83             450, 253, 90, 20, Font.getHelveticaBold(), 12, TextAlign.CENTER));
84
85         // Add Invoice Details Template
86         template.getElements().add(getDetailsGroup());
87
88         // Add BillTo Template
89         template.getElements().add(getBillToGroup());
90
91         // Add ShipTo Template
92         template.getElements().add(getShipToGroup());
93
94         // Add Line Item Template
95         template.getElements().add(getLineItemGroup());
96
97         // Sets the image to the page template
98         setPageTemplateImage();
```

```
99     }
100
101
102     public Template getTemplate() {
103         return template;
104     }
105
106     private Group getDetailsGroup() {
107         // Returns a group containing the details template
108         Group objGroup = new Group();
109
110         objGroup.add(new Rectangle(340, 24, 200, 72, Grayscale.getBlack(),
111             objBGColor, 0.5f));
112         objGroup.add(new Label("Order ID:", 343, 25, 85, 12,
113             Font.getHelveticaBold(), 12, TextAlign.RIGHT));
114         objGroup.add(new Label("Order Date:", 343, 39, 85, 12,
115             Font.getHelveticaBold(), 12, TextAlign.RIGHT));
116         objGroup.add(new Label("Customer ID:", 343, 53, 85, 12,
117             Font.getHelveticaBold(), 12, TextAlign.RIGHT));
118         objGroup.add(new Label("Shipped Date:", 343, 67, 85, 12,
119             Font.getHelveticaBold(), 12, TextAlign.RIGHT));
120         objGroup.add(new Label("Order ID:", 343, 25, 85, 12,
121             Font.getHelveticaBold(), 12, TextAlign.RIGHT));
122         objGroup.add(new Label("Shipped Via:", 343, 81, 85, 12,
123             Font.getHelveticaBold(), 12, TextAlign.RIGHT));
124
125         return objGroup;
126     }
127
128     private Group getBillToGroup() {
129         // Returns a group containing the bill to template
130         Group objGroup = new Group();
131
132         objGroup.add(new Rectangle(25, 120, 200, 90, 0.5f));
133         objGroup.add(new Line(25, 136, 225, 136, 0.5f));
134         objGroup.add(new Label("Bill To:", 28, 121, 200, 12,
135             Font.getHelveticaBold(), 12));
136
137         return objGroup;
138     }
139
140     private Group getShipToGroup() {
141         // Returns a group containing the ship to template
142         Group objGroup = new Group();
143
144         objGroup.add(new Rectangle(315, 120, 200, 90, 0.5f));
145         objGroup.add(new Line(315, 136, 515, 136, 0.5f));
146         objGroup.add(new Label("Ship To:", 318, 121, 200, 12,
147             Font.getHelveticaBold(), 12));
```

```
148
149     return objGroup;
150 }
151
152 private Group getLineItemGroup() {
153
154     // Returns a group containing the line items template
155     Group objGroup = new Group();
156
157     for (int i = 0; i < 10; i++) {
158         objGroup.add(new Rectangle(0, 306 + i * 36, 540, 18, objBGColor,
159             objBGColor));
160     }
161     objGroup.add(new Rectangle(450, 250, 90, 20, 0.5f));
162     objGroup.add(new Rectangle(450, 702, 90, 18, objTotalBGColor,
163         objTotalBGColor));
164     objGroup.add(new Rectangle(0, 270, 540, 450, 0.5f));
165     objGroup.add(new Line(0, 288, 540, 288, 0.5f));
166     objGroup.add(new Line(0, 666, 540, 666, 0.5f));
167     objGroup.add(new Line(60, 270, 60, 666, 0.5f));
168     objGroup.add(new Line(360, 270, 360, 720, 0.5f));
169     objGroup.add(new Line(450, 270, 450, 720, 0.5f));
170     objGroup.add(new Line(450, 702, 540, 702, 0.5f));
171     objGroup.add(new Label("Quantity", 0, 272, 60, 12,
172         Font.getHelveticaBold(), 12, TextAlign.CENTER));
173     objGroup.add(new Label("Description", 60, 272, 300, 12,
174         Font.getHelveticaBold(), 12, TextAlign.CENTER));
175     objGroup.add(new Label("Unit Price", 360, 272, 90, 12,
176         Font.getHelveticaBold(), 12, TextAlign.CENTER));
177     objGroup.add(new Label("Price", 450, 272, 90, 12,
178         Font.getHelveticaBold(), 12, TextAlign.CENTER));
179
180     objGroup.add(new Label("Thank you for your purchase.\nWe " +
181         "appreciate your business.", 5, 672, 350, 54,
182         Font.getHelveticaBold(), 18, objThankYouText));
183     objGroup.add(new Label("Sub Total", 364, 668, 82, 12,
184         Font.getHelveticaBold(), 12, TextAlign.RIGHT));
185     objGroup.add(new Label("Freight", 364, 686, 82, 12,
186         Font.getHelveticaBold(), 12, TextAlign.RIGHT));
187     objGroup.add(new Label("Total", 364, 704, 82, 12,
188         Font.getHelveticaBold(), 12, TextAlign.RIGHT));
189     return objGroup;
190 }
191
192 private void setPageTemplateImage() {
193     // Adds the image to page template if it is not already added
194     if (!pageTemplateImagesSet) {
195         try {
196             template.getElements().add(new Image(getServletContext()
```

```
197         .getRealPath("/images/DPDFLogo.png"),
198         0, 0, 0.16f));
199     } catch (FileNotFoundException ex) {
200         ex.printStackTrace(System.err);
201     }
202     pageTemplateImagesSet = true;
203 }
204 }
205
206 public void draw(Connection connection, Document document,
207     int invoiceNumber) {
208     // Each Invoice should begin a new section
209     document.getSections().begin();
210     // Gets the Invoice data
211     Vector<CeteDAO> v1 = getInvoiceData(connection, invoiceNumber);
212     e1 = v1.elements();
213
214     // Adds the invoice if there is data
215     if (e1 != null) {
216         // Draws the invoice data, returns a page object if it is
217         // the last page
218         Page objLastPage = drawInvoiceData(document);
219         // Draws additional pages if necessary
220         while (objLastPage == null) {
221             objLastPage = drawInvoiceData(document);
222         }
223         // Draws the totals to the bottom of the last page of the
224         // Invoice
225         drawTotals(ceteDAO, objLastPage);
226     }
227
228 }
229
230 private Page drawInvoiceData(Document document) {
231     // Tracks if the invoice is finished
232     boolean invoiceFinished = true;
233     // Tracks the y position on the page
234
235     if (e1.hasMoreElements()) {
236         ceteDAO = (CeteDAO)e1.nextElement();
237         // Sets the Freight amount
238         freight = ceteDAO.getFreight();
239     }
240
241     yOffset = 288;
242     // Create a page for the Invoice
243     Page objPage = new Page(PageSize.LETTER,
244         PageOrientation.PORTRAIT, 36.0f);
245     // Add Details to the Invoice
```

```
246 drawInvoiceDetails(ceteDAO, objPage);
247 // Add bill to address
248 drawBillTo(ceteDAO, objPage);
249 // Add ship to address
250 drawShipTo(ceteDAO, objPage);
251
252 // Add line items to the Invoice
253 drawLineItem(ceteDAO, objPage);
254
255 while (el.hasMoreElements()) {
256     // Break if at the bottom of the page
257     if (yOffset >= 666) {
258         invoiceFinished = false;
259         break;
260     }
261     ceteDAO = (CeteDAO)el.nextElement();
262     // Add line items to the Invoice
263     drawLineItem(ceteDAO, objPage);
264 }
265
266 // Add the page to the document
267 document.getPages().add(objPage);
268
269 // If Invoice is finished return the page else return null so
270 // another page will be added
271 if (invoiceFinished) {
272     return objPage;
273 } else {
274     objPage.getElements().add(new Label("Continued...", 454, 704, 82,
275         12, Font.getHelvetica(), 12, TextAlign.RIGHT));
276     return null;
277 }
278
279 private void drawInvoiceDetails(CeteDAO ceteDAO, Page page) {
280
281     // Adds Invoice details to the page
282     page.getElements().add(new Label(String.valueOf(ceteDAO
283         .getOrderID()), 437, 25, 100, 12, Font.getHelvetica(),
284         12));
285     SimpleDateFormat dateFormat = new SimpleDateFormat("d MMM yyyy");
286     page.getElements().add(new Label(dateFormat.format(ceteDAO
287         .getOrderDate()), 437, 39, 100, 12,
288         Font.getHelvetica(), 12));
289     page.getElements().add(new Label(String.valueOf(ceteDAO
290         .getCustomerID()), 437, 53, 100, 12,
291         Font.getHelvetica(), 12));
292     Date date = ceteDAO.getShippedDate();
293     if (date != null) {
294
```

```
295         page.getElements().add(new Label(dateFormat.format(date),
296             437, 67, 100, 12, Font.getHelvetica(), 12));
297     }
298     page.getElements().add(new Label(ceteDAO.getShipperName(),
299         437, 81, 100, 12, Font.getHelvetica(), 12));
300     page.getElements().add(new Interleaved25(String.valueOf(ceteDAO
301         .getOrderID()), 380, 4, 18, false));
302
303 }
304
305 private void drawBillTo(CeteDAO ceteDAO, Page page) {
306
307     // Adds bill to address
308     String billToAddress = ceteDAO.getBillName() + "\n" +
309         ceteDAO.getBillAddress() + "\n" +
310         ceteDAO.getBillCity() + ", " +
311         ceteDAO.getBillPostalCode() + "\n" +
312         ceteDAO.getBillCountry();
313     page.getElements().add(new TextArea(billToAddress, 28, 139,
314         194, 49, Font.getHelvetica(), 12));
315
316 }
317
318 private void drawShipTo(CeteDAO ceteDAO, Page page) {
319
320     // Adds ship to address
321     String shipToAddress = ceteDAO.getShipName() + "\n" +
322         ceteDAO.getShipAddress() + "\n" +
323         ceteDAO.getShipCity() + ", " +
324         ceteDAO.getShipPostalCode() + "\n" +
325         ceteDAO.getShipCountry();
326     page.getElements().add(new TextArea(shipToAddress, 318, 139,
327         194, 49, Font.getHelvetica(), 12));
328
329 }
330
331 private void drawLineItem(CeteDAO ceteDAO, Page page)    {
332     // Adds a line item to the invoice
333     BigDecimal quantity = new BigDecimal(0.0);
334     BigDecimal unitPrice = new BigDecimal(0.0);
335     quantity = new BigDecimal(ceteDAO.getQuantity());
336     unitPrice = ceteDAO.getUnitPrice();
337
338     BigDecimal lineTotal = unitPrice.multiply(quantity);
339     subTotal = subTotal.add(lineTotal);
340
341     page.getElements().add(new Label(quantity.toString(), 4,
342         3 + yOffset, 52, 12, Font.getHelvetica(), 12,
343         TextAlign.RIGHT));
```

```
344         page.getElements().add(new Label(ceteDAO.getProductName(),
345             64, 3 + yOffset, 292, 12, Font.getHelvetica(), 12));
346
347
348         DecimalFormat df = new DecimalFormat("#,##0.00");
349         page.getElements().add(new Label(df.format(unitPrice
350             .doubleValue()), 364, 3 + yOffset, 82, 12,
351             Font.getHelvetica(), 12, TextAlign.RIGHT));
352         page.getElements().add(new Label(df.format(lineTotal
353             .doubleValue()), 454, 3 + yOffset, 82, 12,
354             Font.getHelvetica(), 12, TextAlign.RIGHT));
355
356         yOffset += 18;
357     }
358
359     private void drawTotals(CeteDAO ceteDAO, Page page) {
360         // Add totals to the bottom of the Invoice
361         BigDecimal grandTotal = subTotal.add(freight);
362         DecimalFormat df = new DecimalFormat("#,##0.00");
363         page.getElements().add(new Label(df.format(subTotal
364             .doubleValue()), 454, 667, 82, 12, Font.getHelvetica(), 12,
365             TextAlign.RIGHT));
366         page.getElements().add(new Label(df.format(freight
367             .doubleValue()), 454, 685, 82, 12, Font.getHelvetica(), 12,
368             TextAlign.RIGHT));
369         page.getElements().add(new Label(df.format(grandTotal
370             .doubleValue()), 454, 703, 82, 12, Font.getHelvetica(), 12,
371             TextAlign.RIGHT));
372     }
373
374     private Vector<CeteDAO> getInvoiceData(Connection connection,
375         int invoiceNumber) {
376         Vector<CeteDAO> v1 = null;
377         // Creates a ResultSet for the Invoice
378         try {
379             PreparedStatement ps = connection.prepareStatement(
380                 "SELECT o.OrderID, o.CustomerID, o.OrderDate, o.ShippedDate, Freight, " +
381                 "o.ShipName, o.ShipAddress, o.ShipCity, o.ShipPostalCode, o.ShipCountry, " +
382                 "c.CompanyName as BillName, c.Address as BillAddress, " +
383                 "c.City as BillCity, c.PostalCode as BillPostalCode, " +
384                 "c.Country as BillCountry, s.CompanyName as ShipperName, " +
385                 "p.ProductName, od.UnitPrice, od.Quantity " +
386                 "FROM Orders o " +
387                 "JOIN Customers c ON " +
388                 "o.CustomerID = c.CustomerID " +
389                 "JOIN Shippers s ON " +
390                 "o.ShipVia = s.ShipperID " +
391                 "JOIN [Order Details] od ON " +
392                 "o.OrderID = od.OrderID " +
```



```
393         "JOIN Products p ON " +
394         "od.ProductID = p.ProductID " +
395         "WHERE o.OrderID = ? ";
396
397         ps.setInt(1, invoiceNumber);
398         ResultSet rs = ps.executeQuery();
399         v1 = new Vector<CeteDAO>(1,1);
400         while (rs.next()) {
401             CeteDAO ceteDAO = new CeteDAO(rs.getInt(1), rs.getString(2),
402                 rs.getDate(3), rs.getDate(4),
403                 rs.getBigDecimal(5), rs.getString(6),
404                 rs.getString(7), rs.getString(8),
405                 rs.getString(9), rs.getString(10),
406                 rs.getString(11), rs.getString(12),
407                 rs.getString(13), rs.getString(14),
408                 rs.getString(15), rs.getString(16),
409                 rs.getString(17), rs.getBigDecimal(18),
410                 rs.getShort(19));
411             v1.add(ceteDAO);
412         }
413     } catch (SQLException ex1) {
414         ex1.printStackTrace(System.err);
415     }
416     return v1;
417 }
418 }
419
420 class CeteDAO {
421
422     private int orderID;
423     private String customerID;
424     private Date orderDate;
425     private Date shippedDate;
426     private BigDecimal freight;
427     private String shipName;
428     private String shipAddress;
429     private String shipCity;
430     private String shipPostalCode;
431     private String shipCountry;
432     private String billName;
433     private String billAddress;
434     private String billCity;
435     private String billPostalCode;
436     private String billCountry;
437     private String shipperName;
438     private String productName;
439     private BigDecimal unitPrice;
440     private short quantity;
441 }
```

```
442     public CeteDAO(int orderID, String customerID, Date orderDate,
443                   Date shippedDate, BigDecimal freight, String shipName,
444                   String shipAddress, String shipCity, String shipPostalCode,
445                   String shipCountry, String billName, String billAddress,
446                   String billCity, String billPostalCode, String billCountry,
447                   String shipperName, String productName, BigDecimal unitPrice,
448                   short quantity) {
449         this.orderID = orderID;
450         this.customerID = customerID;
451         this.orderDate = orderDate;
452         this.shippedDate = shippedDate;
453         this.freight = freight;
454         this.shipName = shipName;
455         this.shipAddress = shipAddress;
456         this.shipCity = shipCity;
457         this.shipPostalCode = shipPostalCode;
458         this.shipCountry = shipCountry;
459         this.billName = billName;
460         this.billAddress = billAddress;
461         this.billCity = billCity;
462         this.billPostalCode = billPostalCode;
463         this.billCountry = billCountry;
464         this.shipperName = shipperName;
465         this.productName = productName;
466         this.unitPrice = unitPrice;
467         this.quantity = quantity;
468     }
469
470     public int getOrderID() {
471         return orderID;
472     }
473
474     public String getCustomerID() {
475         return customerID;
476     }
477
478     public Date getOrderDate() {
479         return orderDate;
480     }
481
482     public Date getShippedDate() {
483         return shippedDate;
484     }
485
486     public BigDecimal getFreight() {
487         return freight;
488     }
489
490     public String getShipName() {
```

```
491         return shipName;
492     }
493
494     public String getShipAddress() {
495         return shipAddress;
496     }
497
498     public String getShipCity() {
499         return shipCity;
500     }
501
502     public String getShipPostalCode() {
503         return shipPostalCode;
504     }
505
506     public String getShipCountry() {
507         return shipCountry;
508     }
509
510     public String getBillName() {
511         return billName;
512     }
513
514     public String getBillAddress() {
515         return billAddress;
516     }
517
518     public String getBillCity() {
519         return billCity;
520     }
521
522     public String getBillPostalCode() {
523         return billPostalCode;
524     }
525
526     public String getBillCountry() {
527         return billCountry;
528     }
529
530     public String getShipperName() {
531         return shipperName;
532     }
533
534     public String getProductName() {
535         return productName;
536     }
537
538     public BigDecimal getUnitPrice() {
539         return unitPrice;
```

```
540     }
541
542     public short getQuantity() {
543         return quantity;
544     }
545
546
547
548
549     }
550
551
552
553
554 }
```