

PurchaseOrders.java

```
1 package ctrl;
2
3 import java.io.File;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 /**
26  * Servlet implementation class PurchaseOrders
27  */
28 @WebServlet("/PurchaseOrders")
29 public class PurchaseOrders extends HttpServlet {
30     private static final long serialVersionUID = 1L;
31
32     /**
33      * @see HttpServlet#HttpServlet()
34      */
35     public PurchaseOrders() {
36         super();
37         // TODO Auto-generated constructor stub
38     }
39
40     /**
41      * @see HttpServlet#doGet(HttpServletRequest request,
42      * HttpServletResponse response)
43      */
44     protected void doGet(HttpServletRequest request,
45     HttpServletResponse response) throws ServletException, IOException
46     {
47         response.setContentType("text/xml");
48
49         String poPath = getServletContext().getRealPath("/POs");
50         String xslFilePath = getServletContext().getRealPath("res/
51         xml/PO.xsl");
52
53         HttpSession s = request.getSession();
54         Writer out = response.getWriter();
55         StringWriter sw = null;
56
57         String orderNum = request.getParameter("orderNum");
58         Account user = (Account) s.getAttribute("AUTH");
59
60         OrderDAO orderDao = new OrderDAO(new File(poPath));
61         File[] orders = orderDao.getPOs(user.getUsername());
62         File orderFile = null;
63
64         String[] userOrderLinks = new String[orders.length + 1];
```

PurchaseOrders.java

```
61      Map<String, String> userOrderLinks = new HashMap<>();
62
63      try {
64          if (orders.length > 0) {
65              int i=1;
66              for (File f: orders) {
67                  String fileName =
68 f.getAbsolutePath().replace(poPath, "");
69                  fileName = fileName.replaceAll("/", "");
70                  fileName = fileName.replaceAll(".xml", "");
71                  fileName.replace(orderDao.getOrderFileNameFirstPart(user.getUserName()), "");
72                  userOrderLinks.put(fileName + "",
73 request.getContextPath() + "/PurchaseOrders?orderNum=" +
74 fileName);
75
76                  if (orderNum != null &&
77 fileName.equals(orderNum)) {
78                      orderFile = f;
79                  }
80
81                  i++;
82              }
83
84              if (orderFile != null) {
85                  sw = new StringWriter();
86                  Source xml = new StreamSource(orderFile);
87                  Source xslt = new StreamSource(xsltFilePath);
88
89                  TransformerFactory tFactory =
90 TransformerFactory.newInstance();
91                  Transformer transform =
92 tFactory.newTransformer(xslt);
93                  transform.transform(xml, new StreamResult(sw));
94              } else {
95                  if (orderNum == null) {
96                      s.setAttribute("ordersLinks",
97 userOrderLinks);
98                      request.setAttribute("ordersLinks",
99 userOrderLinks);
100                  } else {
101                      request.setAttribute("error", "No Purchase
102 Order found!");
103                  }
104              }
105          }
106      } catch (Exception e) {
107          // Handle exception
108      }
109  }
```

PurchaseOrders.java

```
94         }
95     }
96     } else {
97         request.setAttribute("error", "No Purchase Order
for current user!");
98     }
99
100     } catch (Exception e) {
101         request.setAttribute("error", e.getMessage());
102     }
103
104     if (sw != null) {
105         out.write(sw.toString());
106     } else {
107         request.getRequestDispatcher("/WEB-INF/pages/
PurchaseOrders.jspx").forward(request, response);
108     }
109 }
110
111 /**
112  * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
113  */
114 protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
{
115     // TODO Auto-generated method stub
116     doGet(request, response);
117 }
118
119 }
120
```