

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!-- =====
4:   Buildfile for Project 4, version 6.0
5:   Computer Science E-259
6:
7:   This buildfile "shipped" in the root of the following hierarchy.
8:
9:   project4-6.0/
10:    conf/
11:    src/
12:    com/
13:      amazon/
14:        soap/
15:          AWSECommerceService/
16:            cscie259/
17:              project4/
18:                scamazon/
19:                  warehouse/
20: temp/
21: webapps/
22:   ROOT/
23:     docs/
24:       com/
25:         amazon/
26:           soap/
27:             AWSECommerceService/
28:               cscie259/
29:                 project4/
30:                   scamazon/
31:                     warehouse/
32: WEB-INF/
33: scamazon/
34:   dtd/
35:   fo/
36:   images/
37: WEB-INF/
38:   lib/
39:   xml/
40:   cache/
41:   xsd/
42:   xsl/
43: warehouse/
44:   dtd/
45:   images/
46: WEB-INF/
47:   xml/
48:   xsd/
49:   xsl/
50:
51: To compile Scamazon.com and its warehouse, execute 'ant compile' or
52: 'ant' from within project4-6.0/.
53:
54: To generate Javadoc for your code (in project4-6.0/webapps/ROOT/docs/),
55: execute 'ant javadoc' from within project4-6.0/.
56:
57: To publish Javadoc for your code at
58: http://www.people.fas.harvard.edu/~username/cscie259/javadoc/project4-6.0/,
59: where username is your FAS username, execute 'ant publish-javadoc'
60: from within project4-6.0/.
61:
62: To delete your bytecodes as well as Tomcat's logs and runtime
```

```
63:   directories (plus Axis's runtime directories, if any),
64:   execute 'ant clean' from within project4-6.0/.
65: ===== -->
66:
67: <project name="project4" default="compile" basedir=". ">
68:
69:   <description>Project 4</description>
70:
71:   <!-- set global properties for this build -->
72:   <property name="attachments"
73:     location="webapps/warehouse/WEB-INF/attachments" />
74:   <property name="conf" location="conf" />
75:   <property name="docs" location="webapps/ROOT/docs" />
76:   <property name="jwsClasses"
77:     location="webapps/warehouse/WEB-INF/jwsClasses" />
78:   <property name="logs" location="logs" />
79:   <property name="servlets" location="webapps/scamazon/WEB-INF/classes" />
80:   <property name="services" location="webapps/warehouse/WEB-INF/classes" />
81:   <property name="src" location="src" />
82:   <property name="temp" location="temp" />
83:   <property name="work" location="work" />
84:
85:   <!-- init -->
86:   <target name="init">
87:
88:     <!-- set the standard DSTAMP, TSTAMP, and TODAY properties -->
89:     <!-- according to the default formats -->
90:     <tstamp/>
91:
92:     <!-- Create the build directories -->
93:     <mkdir dir="${servlets}" />
94:     <mkdir dir="${services}" />
95:
96:   </target>
97:
98:   <!-- compile -->
99:   <target name="compile"
100:     depends="init"
101:     description="compile Project 4">
102:     <javac srcdir="${src}"
103:       destdir="${servlets}"
104:       debug="true"
105:       fork="true"
106:       includes="com/amazon/soap/AWSECommerceService/**/*.java" />
107:     <javac srcdir="${src}"
108:       destdir="${servlets}"
109:       debug="true"
110:       fork="true"
111:       includes="cscie259/project4/scamazon/**/*.java" />
112:     <javac srcdir="${src}"
113:       destdir="${services}"
114:       debug="true"
115:       fork="true"
116:       includes="cscie259/project4/warehouse/**/*.java" />
117:   </target>
118:
119:   <!-- javadoc -->
120:   <target name="javadoc"
121:     description="generate Javadoc">
122:     <javadoc packagenames="com.amazon.soap.AWSECommerceService.*,cscie259.proj
ect4.*"
123:       sourcepath="${src}"
```

```
124:         destdir="${docs}"
125:         author="true"
126:         version="true"
127:         private="true"
128:         nodeprecated="true"
129:         windowtitle="Project 4"
130:         header="Project 4"/>
131:     </target>
132:
133:     <!-- publish javadoc -->
134:     <target name="publish-javadoc"
135:         depends="javadoc"
136:         description="publish Javadoc">
137:         <copy todir="${user.home}/public_html/cscie259/javadoc/project4-6.0">
138:             <fileset dir="${docs}" />
139:         </copy>
140:         <chmod parallel="false" perm="a+rX" dir="${user.home}/public_html"
141:             includes="**/*" type="both" />
142:     </target>
143:
144:     <!-- clean -->
145:     <target name="clean"
146:         description="remove class, log, runtime, temp, and work files">
147:         <delete dir="${attachments}" />
148:         <delete dir="${conf}/Catalina" />
149:         <delete dir="${jwsClasses}" />
150:         <delete dir="${logs}" />
151:         <delete dir="${services}" />
152:         <delete dir="${servlets}" />
153:         <delete includeemptydirs="true">
154:             <fileset dir="${temp}" includes="**/*" />
155:         </delete>
156:         <delete dir="${work}" />
157:     </target>
158:
159: </project>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!-- =====
4: Configuration file for Project 4, version 6.0
5: Computer Science E-259
6:
7: Be sure to set the value of the Server element's port
8: attribute to an integer between 1024 and 65535, inclusive.
9: Of course, that port cannot already be in use. Nor can it
10: be the same value you choose for the Connector element's port.
11:
12: Also be sure to set the value of the Connector element's port
13: attribute to an integer between 1024 and 65535, inclusive.
14: Of course, that port cannot already be in use. Nor can it
15: be the same value you choose for the Server element's port.
16: ===== -->
17:
18:
19: <!-- A "Server" is a singleton element that represents the entire JVM,
20: which may contain one or more "Service" instances. The Server
21: listens for a shutdown command on the indicated port.
22:
23: Note: A "Server" is not itself a "Container", so you may not
24: define subcomponents such as "Valves" or "Loggers" at this level.
25: -->
26:
27: <Server port="" shutdown="SHUTDOWN">
28:
29: <!-- Global JNDI resources -->
30: <GlobalNamingResources>
31:
32: <!-- Editable user database that can also be used by
33:      UserDatabaseRealm to authenticate users -->
34: <Resource name="UserDatabase" auth="Container"
35:      type="org.apache.catalina.UserDatabase"
36:      description="User database that can be updated and saved"
37:      factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
38:      pathname="conf/tomcat-users.xml" />
39:
40: </GlobalNamingResources>
41:
42: <!-- A "Service" is a collection of one or more "Connectors" that share
43:      a single "Container" (and therefore the web applications visible
44:      within that Container). Normally, that Container is an "Engine",
45:      but this is not required. -->
46:
47: <!-- Define the Tomcat Stand-Alone Service -->
48: <Service name="Catalina">
49:
50: <!-- A "Connector" represents an endpoint by which requests are received
51:      and responses are returned. Each Connector passes requests on to the
52:      associated "Container" (normally an Engine) for processing. -->
53:
54: <!-- Define a non-SSL HTTP/1.1 Connector -->
55: <Connector port=""
56:      maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
57:      enableLookups="false" acceptCount="100"
58:      connectionTimeout="20000" disableUploadTimeout="true" />
59:
60: <!-- An Engine represents the entry point (within Catalina) that processes
61:      every request. The Engine implementation for Tomcat stand alone
62:      analyzes the HTTP headers included with the request, and passes them
```

```
63:      on to the appropriate Host (virtual host). -->
64:
65: <!-- Define the top level container in our container hierarchy -->
66: <Engine name="Catalina" defaultHost="localhost">
67:
68: <!-- Because this Realm is here, an instance will be shared globally -->
69:
70: <!-- This Realm uses the UserDatabase configured in the global JNDI
71:      resources under the key "UserDatabase". Any edits
72:      that are performed against this UserDatabase are immediately
73:      available for use by the Realm. -->
74: <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
75:      resourceName="UserDatabase"/>
76:
77: <!-- Define the default virtual host -->
78: <Host name="localhost" appBase="webapps"
79:      unpackWARs="true" autoDeploy="true"
80:      xmlValidation="true" xmlNamespaceAware="true">
81:
82: <!-- Root Context -->
83: <Context path="" docBase="ROOT" debug="0" reloadable="true">
84: <!-- disable persistent sessions -->
85: <Manager pathname="" />
86: </Context>
87:
88: <!-- Scamazon.com Context -->
89: <Context path="/scamazon"
90:      docBase="scamazon" debug="0" reloadable="true">
91: <!-- disable persistent sessions -->
92: <Manager pathname="" />
93: </Context>
94:
95: <!-- Warehouse Context -->
96: <Context path="/warehouse"
97:      docBase="warehouse" debug="0" reloadable="true">
98: <!-- disable persistent sessions -->
99: <Manager pathname="" />
100: </Context>
101:
102:
103: </Host>
104:
105: </Engine>
106:
107: </Service>
108:
109: </Server>
```

```
1: package cscie259.project4.scamazon;
2:
3: import java.io.IOException;
4:
5: import javax.servlet.ServletException;
6:
7: import javax.servlet.http.HttpServletRequest;
8: import javax.servlet.http.HttpServletResponse;
9:
10:
11: /**
12:  * This servlet implements Scamazon.com's shopping cart.
13:  *
14:  * Any references to the local file system are resolved
15:  * with respect to project4-6.0/webapps/scamazon/, per
16:  * ScamazonServlet's implementation of javax.xml.transform.URIResolver.
17:  * <p/>
18:  * You MUST modify this file.
19:  *
20:  * @author Computer Science E-259
21:  * @version 6.0
22:  *
23:  * @author YOUR NAME GOES HERE
24:  */
25:
26: public class Cart extends ScamazonServlet
27: {
28:     /**
29:      * Handles operations involving the shopping cart.
30:      *
31:      * @param request HTTP request object
32:      * @param response HTTP response object
33:      *
34:      * @throws IOException if an IO-related error occurs
35:      * @throws ServletException if a servlet-related error occurs
36:      */
37:
38:     protected void doWork(HttpServletRequest request,
39:                             HttpServletResponse response)
40:         throws IOException, ServletException
41:     {
42:         // TODO
43:     }
44: }
```

```

1: package cscie259.project4.scamazon;
2:
3: import java.io.IOException;
4: import java.io.PrintWriter;
5:
6: import java.util.Enumeration;
7:
8: import javax.servlet.ServletException;
9:
10: import javax.servlet.http.HttpServletRequest;
11: import javax.servlet.http.HttpServletResponse;
12:
13: import javax.xml.transform.ErrorListener;
14: import javax.xml.transform.Source;
15: import javax.xml.transform.Transformer;
16: import javax.xml.transform.TransformerException;
17: import javax.xml.transform.TransformerFactory;
18: import javax.xml.transform.URIResolver;
19:
20: import javax.xml.transform.sax.SAXSource;
21:
22: import javax.xml.transform.stream.StreamResult;
23: import javax.xml.transform.stream.StreamSource;
24:
25: import org.xml.sax.InputSource;
26: import org.xml.sax.XMLReader;
27:
28: import org.xml.sax.helpers.XMLReaderFactory;
29:
30:
31: /**
32:  * This servlet implements Scamazon.com's catalog.
33:  * <p/>
34:  * Any references to the local file system are resolved
35:  * with respect to project4-6.0/webapps/scamazon/, per
36:  * ScamazonServlet's implementation of javax.xml.transform.URIResolver.
37:  * <p/>
38:  * You MAY modify this file.
39:  *
40:  * @author Computer Science E-259
41:  * @version 6.0
42:  *
43:  * @author YOUR NAME GOES HERE
44:  */
45:
46: public class Catalog extends ScamazonServlet
47: {
48:     /**
49:      * Handles operations involving the catalog.
50:      *
51:      * @param request HTTP request object
52:      * @param response HTTP response object
53:      *
54:      * @throws IOException if an IO-related error occurs
55:      * @throws ServletException if a servlet-related error occurs
56:      */
57:
58:     public void doWork(HttpServletRequest request,
59:                       HttpServletResponse response)
60:     throws IOException, ServletException
61:     {
62:         // output text/html MIME type

```

```

63:         response.setContentType("text/html");
64:
65:         // retrieve PrintWriter
66:         PrintWriter out = response.getWriter();
67:
68:         // attempt to process catalog operation, passing any parameters
69:         // to stylesheet
70:         try
71:         {
72:             // instantiate a transformer factory
73:             TransformerFactory tFactory = TransformerFactory.newInstance();
74:
75:             // instantiate AnErrorListener
76:             AnErrorListener errorListener = new AnErrorListener();
77:
78:             // handle errors during processing of transformation instructions
79:             tFactory.setErrorListener(errorListener);
80:
81:             // configure URI resolving
82:             tFactory.setURIResolver(new myURIResolver());
83:
84:             // load stylesheet
85:             StreamSource xslSource = new StreamSource(getServletContext().
86:                                                         getResourceAsStream("/xsl/catalog.xsl"));
87:
88:             // instantiate transformer
89:             Transformer transformer = tFactory.newTransformer(xslSource);
90:
91:             // handle errors during the transformation
92:             transformer.setErrorListener(errorListener);
93:
94:             // pass HTTP parameters to stylesheet
95:             Enumeration names = request.getParameterNames();
96:             while (names.hasMoreElements())
97:             {
98:                 String name = (String)names.nextElement();
99:                 String value = request.getParameter(name);
100:                 transformer.setParameter(name, value);
101:             }
102:
103:             // create an XMLReader
104:             XMLReader reader
105:                 = XMLReaderFactory.createXMLReader();
106:
107:             // enable validation as appropriate
108:             String validation = System.getProperty("enable.validation");
109:             if (validation != null && validation.equals("dtd"))
110:             {
111:                 reader.setFeature("http://xml.org/sax/features/validation",
112:                                     true);
113:             }
114:             else if (validation != null && validation.equals("xsd"))
115:             {
116:                 reader.setFeature("http://xml.org/sax/features/validation",
117:                                     true);
118:                 reader.setFeature("http://apache.org/xml/features/validation/schem
119:                                     a",
120:                                     true);
121:             }
122:             else
123:             {
124:                 reader.setFeature("http://xml.org/sax/features/validation",

```

```
124:             false);
125:         }
126:
127:         // load XML
128:         SAXSource saxSource = new SAXSource(reader,
129:             new InputSource(getServletContext().
130:                 getResourceAsStream("/xml/catalog.xml")));
131:
132:         // transform!
133:         transformer.transform(saxSource, new StreamResult(out));
134:
135:     }
136:     catch (Exception e)
137:     {
138:         e.printStackTrace();
139:         throw new ServletException("Can't process XSLT: " + e.getMessage());
140:     }
141: }
142: }
```



```
1: package cscie259.project4.scamazon;
2:
3: import java.io.IOException;
4:
5: import javax.servlet.RequestDispatcher;
6: import javax.servlet.ServletException;
7:
8: import javax.servlet.http.HttpServletRequest;
9: import javax.servlet.http.HttpServletResponse;
10:
11: import javax.xml.transform.ErrorListener;
12: import javax.xml.transform.TransformerException;
13:
14: import org.apache.axis.client.Call;
15: import org.apache.axis.client.Service;
16:
17:
18: /**
19:  * Helper methods for Scamazon.com.
20:  * <p/>
21:  * You MAY modify this file.
22:  *
23:  * @author Computer Science E-259
24:  * @version 6.0
25:  *
26:  * @author YOUR NAME GOES HERE
27:  */
28:
29: public final class Proxy
30: {
31:     /**
32:      * Default constructor is private so that this utility
33:      * class cannot be instantiated.
34:      */
35:     private Proxy() {}
36:
37:
38:     /**
39:      * A proxy for the web service. Takes as input a PO element in the
40:      * form of a String (as well as the warehouse's current host and port);
41:      * returns as a String a PO-ACK element.
42:      *
43:      * @param poXmlString PO element
44:      * @param host warehouse's host (e.g., "icel.fas.harvard.edu")
45:      * @param port warehouse's (i.e., servlet container's) current port
46:      *
47:      * @throws ServletException if a servlet-related error occurs
48:      *
49:      * @return PO-ACK element
50:      */
51:     public static String processPO(String poXmlString, String host, int port)
52:     throws ServletException
53:     {
54:         // attempt to invoke web service
55:         try
56:         {
57:             // warehouse's location
58:             String endpoint = "http://" + host + ":" + port +
59:                 "/warehouse/services/Purchasing";
60:
61:             // prepare to invoke the web service
62:             Service service = new Service();
```

```
63:             Call call = (Call) service.createCall();
64:             call.setTargetEndpointAddress(new java.net.URL(endpoint));
65:             call.setOperationName("processPO");
66:
67:             // invoke the service and return the (PO-ACK) response
68:             return (String) call.invoke(new Object [] { poXmlString });
69:         }
70:         catch (Exception e)
71:         {
72:             throw new ServletException("Error running SOAP: " +
73:                 e.getMessage());
74:         }
75:     }
76: }
```



```

1: package cscie259.project4.scamazon;
2:
3: import java.io.*;
4:
5: import javax.servlet.*;
6: import javax.servlet.http.*;
7:
8: import javax.xml.transform.Transformer;
9: import javax.xml.transform.TransformerFactory;
10: import javax.xml.transform.TransformerException;
11: import javax.xml.transform.Source;
12: import javax.xml.transform.Result;
13: import javax.xml.transform.URIResolver;
14:
15: import javax.xml.transform.sax.SAXResult;
16:
17: import javax.xml.transform.stream.StreamSource;
18:
19: import org.xml.sax.InputSource;
20: import org.xml.sax.XMLReader;
21:
22: import org.apache.fop.apps.Driver;
23: import org.apache.fop.apps.Version;
24: import org.apache.fop.apps.XSLTInputHandler;
25: import org.apache.fop.messaging.MessageHandler;
26:
27: import org.apache.avalon.framework.logger.ConsoleLogger;
28: import org.apache.avalon.framework.logger.Logger;
29:
30:
31: /**
32:  * This servlet produces Scamazon.com's receipts as PDFs.
33:  * <p>
34:  * Any references to the local file system in the servlet's context
35:  * are resolved with respect to project4-6.0/webapps/scamazon/.
36:  * <p>
37:  * Based on version 1.1.2.1 of Apache's FopServlet.java,
38:  * per the below copyright notice; original source from
39:  * FOP 0.20.5rc2's distribution.
40:  * <p>
41:  * $Id: FopServlet.java,v 1.1.2.1 2003/01/13 16:27:14 jeremias Exp $
42:  * Copyright (C) 2001 The Apache Software Foundation. All rights reserved.
43:  * For details on use and redistribution please refer to the
44:  * LICENSE file included with these sources.
45:  * <p>
46:  * Given an XSL-FO file or, alternatively, XML and XSLT which together
47:  * yield XSL-FO, produces PDF, which is returned directly to the
48:  * browser.
49:  * <p>
50:  * Current usage:
51:  * <pre>
52:  * http://$HOSTNAME.fas.harvard.edu:n/scamazon/servlet/receipt?fo=foo.fo&ext=.pdf
53:  * http://$HOSTNAME.fas.harvard.edu:n/scamazon/servlet/receipt?xml=foo.xml&xsl=f
54:  * oo.xsl&ext=.pdf
55:  * </pre>
56:  * Per http://xml.apache.org/fop/embedding.html, "Some browsers have problems
57:  * handling the PDF result sent back to the browser. IE is particularly bad
58:  * and different versions behave differently. Having a '.pdf' on the end of
59:  * the URL may help."
60:  * <p>
61:  * Hence our inclusion of the unused "ext" parameter in the current usage.
62:  * <p>
63:  *
64:  * @author The Apache Software Foundation
65:  * @author Computer Science E-259
66:  * @version 6.0
67:  *
68:  * @author YOUR NAME GOES HERE
69:  */
70:
71: public class Receipt extends ScamazonServlet
72: {
73:     /**
74:      * HTTP parameter referencing an XSL-FO file
75:      */
76:     public static final String FO_REQUEST_PARAM = "fo";
77:
78:
79:     /**
80:      * HTTP parameter referencing an XML file
81:      */
82:     public static final String XML_REQUEST_PARAM = "xml";
83:
84:
85:     /**
86:      * HTTP parameter referencing an XSLT file
87:      */
88:     public static final String XSLT_REQUEST_PARAM = "xsl";
89:
90:
91:     /* reference to a Logger */
92:     private Logger log = null;
93:
94:
95:     /**
96:      * Responds to GETs in the same manner as POSTs.
97:      *
98:      * @param request HTTP request object
99:      * @param response HTTP response object
100:      *
101:      * @throws IOException if an IO-related error occurs
102:      * @throws ServletException if a servlet-related error occurs
103:      */
104:
105:     public void doGet(HttpServletRequest request,
106:                       HttpServletResponse response)
107:     throws IOException, ServletException
108:     {
109:         doWork(request, response);
110:     }
111:
112:
113:     /**
114:      * Respond to POSTs in the same manner as GETs.
115:      *
116:      * @param request HTTP request object
117:      * @param response HTTP response object
118:      *
119:      * @throws IOException if an IO-related error occurs
120:      * @throws ServletException if a servlet-related error occurs
121:      */
122:     public void doPost(HttpServletRequest request,

```

```
123:         HttpServletResponse response)
124:     throws IOException, ServletException
125:     {
126:         doWork(request, response);
127:     }
128:
129:
130: /**
131:  * Handles operations involving receipts.
132:  *
133:  * @param request HTTP request object
134:  * @param response HTTP response object
135:  *
136:  * @throws ServletException if a servlet-related error occurs
137:  */
138: public void doWork(HttpServletRequest request,
139:                   HttpServletResponse response)
140:     throws ServletException
141:     {
142:         // instantiate Logger as necessary
143:         if (log == null)
144:         {
145:             log = new ConsoleLogger(ConsoleLogger.LEVEL_WARN);
146:             MessageHandler.setScreenLogger(log);
147:         }
148:
149:         // attempt to process request
150:         try
151:         {
152:             // retrieve HTTP parameters
153:             String foParam = request.getParameter(FO_REQUEST_PARAM);
154:             String xmlParam = request.getParameter(XML_REQUEST_PARAM);
155:             String xslParam = request.getParameter(XSLT_REQUEST_PARAM);
156:
157:             // process XSL-FO request; assume (arbitrarily) that file is in
158:             // project4-6.0/webapps/scamazon/fo/
159:             if (foParam != null)
160:             {
161:                 InputSource inputSource
162:                     = new InputSource(getServletContext().
163:                                     getResourceAsStream("/fo/" + foParam));
164:                 renderFO(inputSource, response);
165:             }
166:
167:             // process XML/XSLT request
168:             else if ((xmlParam != null) && (xslParam != null))
169:             {
170:                 processXSL(xmlParam, xslParam, response);
171:             }
172:
173:             // err
174:             else
175:             {
176:                 // output text/plain MIME type
177:                 response.setContentType("text/plain");
178:
179:                 // retrieve PrintWriter
180:                 PrintWriter out = response.getWriter();
181:
182:                 // report error
183:                 out.println("Error: improper usage.");
184:             }
185:         }
```

```
185:     }
186:     catch (ServletException e)
187:     {
188:         throw e;
189:     }
190:     catch (Exception e)
191:     {
192:         throw new ServletException(e);
193:     }
194: }
195:
196:
197: /**
198:  * Renders an FO input source into a PDF file which is written
199:  * directly to the response object's OutputStream.
200:  *
201:  * @param foFile XSL-FO file
202:  * @param response HTTP response object
203:  *
204:  * @throws ServletException if a servlet-related error occurs
205:  */
206: public void renderFO(InputSource foFile, HttpServletResponse response)
207:     throws ServletException
208:     {
209:         // attempt to generate receipt
210:         try
211:         {
212:             // instantiate new output stream
213:             ByteArrayOutputStream out = new ByteArrayOutputStream();
214:
215:             // output application/pdf MIME type
216:             response.setContentType("application/pdf");
217:
218:             // process XSL-FO, storing results in output stream
219:             Driver driver = new Driver(foFile, out);
220:             driver.setLogger(log);
221:             driver.setRenderer(Driver.RENDER_PDF);
222:             driver.run();
223:
224:             // return output
225:             byte [] content = out.toByteArray();
226:             response.setContentLength(content.length);
227:             response.getOutputStream().write(content);
228:             response.getOutputStream().flush();
229:         }
230:         catch (Exception e)
231:         {
232:             throw new ServletException(e);
233:         }
234:     }
235:
236:
237: /**
238:  * Renders an XML file into a PDF file by applying a stylesheet
239:  * that converts the XML to XSL-FO. The PDF is written
240:  * directly to the response object's OutputStream.
241:  *
242:  * @param xml name of XML input file
243:  * @param xsl name of XSLT input file
244:  * @param response HTTP response object
245:  *
246:  * @throws ServletException if a servlet-related error occurs
```

```
247:    */
248:    public void processXSL(String xml, String xsl,
249:        HttpServletResponse response)
250:        throws ServletException
251:    {
252:        // instantiate driver
253:        Driver driver = new Driver();
254:
255:        // instantiate and configure Logger
256:        Logger logger = new ConsoleLogger(ConsoleLogger.LEVEL_INFO);
257:        driver.setLogger(logger);
258:        MessageHandler.setScreenLogger(logger);
259:
260:        // configure renderer
261:        driver.setRenderer(Driver.RENDER_PDF);
262:
263:        // instantiate and configure output stream
264:        ByteArrayOutputStream out = new ByteArrayOutputStream();
265:        driver.setOutputStream(out);
266:
267:        // output application/pdf MIME type
268:        response.setContentType("application/pdf");
269:
270:        // attempt to generate receipt
271:        try
272:        {
273:            // instantiate a transformer factory
274:            TransformerFactory tFactory = TransformerFactory.newInstance();
275:
276:            // configure URI resolving
277:            tFactory.setURIResolver(new myURIResolver());
278:
279:            // load stylesheet; assume (arbitrarily) that file is in
280:            // project4-6.0/webapps/scamazon/xsl/
281:            StreamSource xslSource
282:            = new StreamSource(getServletContext().
283:                getResourceAsStream("/xsl/" + xsl));
284:
285:            // instantiate transformer
286:            Transformer transformer = tFactory.newTransformer(xslSource);
287:
288:            // resulting SAX events (the generated XSL-FO)
289:            // must be piped through to FOP
290:            Result res = new SAXResult(driver.getContentHandler());
291:
292:            // load XML; assume (arbitrarily) that file is in
293:            // project4-6.0/webapps/scamazon/xml/
294:            Source xmlSource
295:            = new StreamSource(getServletContext().
296:                getResourceAsStream("/xml/" + xml));
297:
298:            // start XSLT transformation and FOP processing
299:            transformer.transform(xmlSource, res);
300:
301:            // return output
302:            byte [] content = out.toByteArray();
303:            response.setContentLength(content.length);
304:            response.getOutputStream().write(content);
305:            response.getOutputStream().flush();
306:        }
307:        catch (Exception e)
308:        {
```

```
309:            throw new ServletException(e);
310:        }
311:    }
312:
313:
314:    /**
315:     * Resolves references to files.
316:     */
317:    private class myURIResolver implements URIResolver
318:    {
319:        /**
320:         * Resolves references to files with respect to
321:         * project4-6.0/webapps/scamazon/.
322:         *
323:         * @param href path to file
324:         * @param base current base
325:         *
326:         * @throws TransformerException if a transformation-related error occurs
327:         */
328:        public Source resolve(String href, String base)
329:            throws TransformerException
330:        {
331:            return (new StreamSource(getServletContext().
332:                getResourceAsStream("/" + href)));
333:        }
334:    }
335: }
```

```

1: package cscie259.project4.scamazon;
2:
3: import java.io.IOException;
4:
5: import javax.servlet.RequestDispatcher;
6: import javax.servlet.ServletException;
7:
8: import javax.servlet.http.HttpServlet;
9: import javax.servlet.http.HttpServletRequest;
10: import javax.servlet.http.HttpServletResponse;
11:
12: import javax.xml.transform.ErrorListener;
13: import javax.xml.transform.Source;
14: import javax.xml.transform.Transformer;
15: import javax.xml.transform.TransformerException;
16: import javax.xml.transform.URIResolver;
17:
18: import javax.xml.transform.stream.StreamSource;
19:
20:
21: /**
22:  * Base class for Scamazon's servlets; defines as an inner class an ErrorListener.
23:  * You MAY modify this file.
24:  *
25:  * @author Computer Science E-259
26:  * @version 6.0
27:  *
28:  * @author YOUR NAME GOES HERE
29:  */
30:
31:
32: public abstract class ScamazonServlet extends HttpServlet
33: {
34:     /**
35:      * Responds to GETs in the same manner as POSTs.
36:      *
37:      * @param request HTTP request object
38:      * @param response HTTP response object
39:      *
40:      * @throws IOException on network failure
41:      * @throws ServletException on generic failure
42:      */
43:     public void doGet(HttpServletRequest request,
44:                       HttpServletResponse response)
45:         throws IOException, ServletException
46:     {
47:         doWork(request, response);
48:     }
49:
50:     /**
51:      * Responds to POSTs in the same manner as GETs.
52:      *
53:      * @param request HTTP request object
54:      * @param response HTTP response object
55:      *
56:      * @throws IOException if an IO-related error occurs
57:      * @throws ServletException if a servlet-related error occurs
58:      */
59:     public void doPost(HttpServletRequest request,
60:                       HttpServletResponse response)
61:         throws IOException, ServletException
62:     {
63:         doWork(request, response);
64:     }
65:
66:
67:     /**
68:      * To be implemented by descendants, handles GETs and POSTs identically.
69:      *
70:      * @param request HTTP request object
71:      * @param response HTTP response object
72:      *
73:      * @throws IOException if an IO-related error occurs
74:      * @throws ServletException if a servlet-related error occurs
75:      */
76:     protected abstract void doWork(HttpServletRequest request,
77:                                     HttpServletResponse response)
78:         throws IOException, ServletException;
79:
80:
81:
82:     /**
83:      * Used to forward the request and response objects to another servlet
84:      * (whose path is of the form "/servlet/foo") for processing.
85:      *
86:      * @param request HTTP request object
87:      * @param response HTTP response object
88:      * @param path the path to the servlet being forwarded to
89:      *
90:      * @throws IOException if an IO-related error occurs
91:      * @throws ServletException if a servlet-related error occurs
92:      */
93:     public void forward(HttpServletRequest request,
94:                       HttpServletResponse response,
95:                       String path)
96:         throws IOException, ServletException
97:     {
98:         RequestDispatcher dispatch =
99:             request.getRequestDispatcher(path);
100:         dispatch.forward(request, response);
101:     }
102:
103:
104:     /**
105:      * Used to redirect the user to another url (or servlet in the
106:      * same container), without preserving the current
107:      * HttpServletRequest object.
108:      *
109:      * @param response HTTP response object
110:      * @param url the url to which the user's being redirected
111:      *
112:      * @throws IOException if an IO-related error occurs
113:      */
114:     public void redirect(HttpServletResponse response, String url)
115:         throws IOException
116:     {
117:         response.sendRedirect(url);
118:     }
119:
120:
121:     /**
122:      * Handles processing errors.
123:      */
124: }

```

```
125:     protected static class AnErrorListener implements ErrorListener
126:     {
127:         /**
128:          * Receive notification of a parser warning.
129:          *
130:          * @param e the exception thrown
131:          *
132:          * @throws TransformerException if a transformation-related error occurs
133:          */
134:         public void warning(TransformerException e)
135:         throws TransformerException
136:         {
137:             System.out.println("Parsing warning:  " + e.getMessage());
138:         }
139:
140:
141:         /**
142:          * Receive notification of a recoverable parser error.
143:          *
144:          * @param e the exception thrown
145:          *
146:          * @throws TransformerException if a transformation-related error occurs
147:          */
148:         public void error(TransformerException e)
149:         throws TransformerException
150:         {
151:             System.out.println("Parsing error:  " + e.getMessage());
152:         }
153:
154:
155:         /**
156:          * Report a fatal XML parsing error.
157:          *
158:          * @param e the exception thrown
159:          *
160:          * @throws TransformerException if a transformation-related error occurs
161:          */
162:         public void fatalError(TransformerException e)
163:         throws TransformerException
164:         {
165:             System.out.println("Fatal parsing error:  " + e.getMessage());
166:         }
167:     }
168:
169:
170:     /**
171:     * Resolves references to files.
172:     */
173:     protected class myURIResolver implements URIResolver
174:     {
175:         /**
176:          * Resolves references to files with respect to
177:          * project4-6.0/webapps/scamazon/.
178:          *
179:          * @param href path to file
180:          * @param base current base
181:          *
182:          * @throws TransformerException if a transformation-related error occurs
183:          *
184:          * @return Source for resolution of references to files
185:          */
186:         public Source resolve(String href, String base)
```

```
187:         throws TransformerException
188:         {
189:             return (new StreamSource(getServletContext().
190:                                     getResourceAsStream("/" + href)));
191:         }
192:     }
193: }
```

```

1: package cscie259.project4.warehouse;
2:
3: import java.io.ByteArrayInputStream;
4: import java.io.StringWriter;
5: import javax.xml.transform.ErrorListener;
6: import javax.xml.transform.Source;
7: import javax.xml.transform.Transformer;
8: import javax.xml.transform.TransformerException;
9: import javax.xml.transform.TransformerFactory;
10: import javax.xml.transform.sax.SAXSource;
11: import javax.xml.transform.stream.StreamResult;
12: import javax.xml.transform.stream.StreamSource;
13: import org.xml.sax.InputSource;
14: import org.xml.sax.XMLReader;
15: import org.xml.sax.helpers.XMLReaderFactory;
16:
17:
18: /**
19:  * This class is treated by Tomcat (or, more specifically,
20:  * by Axis) as a web service; it processes purchase orders.
21:  * <p/>
22:  * Any references to the local file system are resolved
23:  * with respect to project4-6.0/.
24:  * <p/>
25:  * The class does not utilize a javax.xml.transform.URIResolver,
26:  * since it's not technically a servlet (i.e., it doesn't extend
27:  * javax.servlet.http.HttpServlet).
28:  * <p/>
29:  * You MAY modify this file.
30:  *
31:  * @author Computer Science E-259
32:  * @version 6.0
33:  *
34:  * @author YOUR NAME GOES HERE
35:  */
36:
37: public class Purchasing
38: {
39:     /* path to the warehouse's base directory */
40:     private static final String warehouseDir = "webapps/warehouse";
41:
42:
43:     /**
44:      * Processes the given PO element, provided in the form of a String.
45:      *
46:      * @param poXmlString String containing PO element
47:      *
48:      * @return String containing PO-ACK element
49:      */
50:
51:     public String processPO(String poXmlString)
52:     {
53:         // storage for PO-ACK output
54:         StringWriter writer = new StringWriter();
55:
56:         // attempt to process PO
57:         try
58:         {
59:             // instantiate a transformer factory
60:             TransformerFactory tFactory = TransformerFactory.newInstance();
61:
62:             // instantiate ErrorListener

```

```

63:             AnErrorListener errorListener = new AnErrorListener();
64:
65:             // handle errors during processing of transformation instructions
66:             tFactory.setErrorListener(errorListener);
67:
68:             // load stylesheet
69:             StreamSource xslSource =
70:                 new StreamSource(warehouseDir + "/xsl/po-ack.xsl");
71:
72:             // instantiate transformer
73:             Transformer transformer = tFactory.newTransformer(xslSource);
74:
75:             // handle errors during the transformation
76:             transformer.setErrorListener(errorListener);
77:
78:             // create an XMLReader
79:             XMLReader reader = XMLReaderFactory.createXMLReader();
80:
81:             // enable validation as appropriate
82:             String validation = System.getProperty("enable.validation");
83:             if (validation != null && validation.equals("dtd"))
84:                 reader.setFeature("http://xml.org/sax/features/validation", true);
85:             else if (validation != null && validation.equals("xsd"))
86:             {
87:                 reader.setFeature("http://xml.org/sax/features/validation", true);
88:                 reader.setFeature("http://apache.org/xml/features/validation/schem
a", true);
89:             }
90:             else
91:                 reader.setFeature("http://xml.org/sax/features/validation", false);
92:
93:             // load XML
94:             SAXSource saxSource = new SAXSource(reader,
95:                 new InputSource(new ByteArrayInputStream(poXmlString.getBytes())));
96:
97:             // transform!
98:             transformer.transform(saxSource, new StreamResult(writer));
99:         }
100:         catch (Exception e)
101:         {
102:             e.printStackTrace();
103:             throw new RuntimeException("Can't process XSLT: " + e.getMessage());
104:         }
105:
106:         // return PO-ACK
107:         return writer.toString();
108:     }
109:
110:
111:     /**
112:      * Handle processing errors.
113:      */
114:
115:     private class AnErrorListener implements ErrorListener
116:     {
117:         /**
118:          * Receive notification of a parser warning.
119:          *
120:          * @param e the exception thrown
121:          */

```

```
122:         * @throws TransformerException
123:         */
124:     public void warning(TransformerException e)
125:     throws TransformerException
126:     {
127:         System.out.println("Parsing warning:  " + e.getMessage());
128:     }
129:
130:
131:     /**
132:     * Receive notification of a recoverable parser error.
133:     *
134:     * @param e    the exception thrown
135:     *
136:     * @throws TransformerException
137:     */
138:     public void error(TransformerException e)
139:     throws TransformerException
140:     {
141:         System.out.println("Parsing error:  " + e.getMessage());
142:     }
143:
144:
145:     /**
146:     * Report a fatal XML parsing error.
147:     *
148:     * @param e    the exception thrown
149:     *
150:     * @throws TransformerException
151:     */
152:     public void fatalError(TransformerException e)
153:     throws TransformerException
154:     {
155:         System.out.println("Fatal parsing error:  " + e.getMessage());
156:     }
157: }
158: }
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-transitional.dtd">
4:
5: <%-- direct Xalan's environment to browser instead of console --%>
6: <% java.io.ByteArrayOutputStream os = new java.io.ByteArrayOutputStream(); %>
7: <% java.io.PrintWriter pw = new java.io.PrintWriter(os); %>
8: <% (new org.apache.xalan.xslt.EnvironmentCheck()).checkEnvironment(pw); %>
9:
10: <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
11:   <head>
12:     <title>Environment Happiness Page</title>
13:   </head>
14:   <body>
15:     <pre>
16: <% out.print(os); %>
17:   </pre>
18:   </body>
19: </html>
```



```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!DOCTYPE web-app
4:     PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
5:     "http://java.sun.com/dtd/web-app_2_3.dtd">
6:
7: <!-- =====
8:     Configuration file for Project 4, version 6.0
9:     Computer Science E-259
10: ===== -->
11:
12: <web-app/>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-transitional.dtd">
4:
5: <%-- =====
6:     Demonstration of Amazon's ECS for Project 4, version 6.0
7:     Computer Science E-259
8: ===== --%>
9:
10: <%@ page language="java" %>
11: <%@ taglib uri="http://java.sun.com/jstl/core" prefix="c" %>
12:
13: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
14:   <head>
15:     <title>Books</title>
16:   </head>
17:   <body>
18:     <c:forEach var="item" items='${requestScope.items}'>
19:       <c:out value='${item.itemAttributes.title}' />
20:       <br />
21:       <c:out value='${item.itemAttributes.author[0]}' />
22:       <br />
23:       <c:out value='${item.itemAttributes.publisher}' />,
24:       <c:out value='${item.itemAttributes.publicationDate}' />
25:       <br />
26:       <c:out value='${item.itemAttributes.ISBN}' />
27:       <br />
28:       <br />
29:       Reviews
30:       <br />
31:       <ul>
32:         <c:forEach var="review" items='${item.customerReviews.review}'>
33:           <li>
34:             (<c:out value='${review.rating}' />)
35:             <c:out value='${review.summary}' />
36:           </li>
37:         </c:forEach>
38:       </ul>
39:       <br />
40:       <hr />
41:       <br />
42:     </c:forEach>
43:   </body>
44: </html>
```

1: <!-- TODO (unless you create an XML schema instead) -->

1: <!-- TODO -->

2:

1: <!-- TODO (unless you create an XML schema instead) -->

1: <!-- TODO (unless you create an XML schema instead) -->

```
1: <%-- redirect user to Catalog servlet by default --%>
2:
3: <% response.sendRedirect("/scamazon/servlet/catalog"); %>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!DOCTYPE web-app
4:   PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
5:   "http://java.sun.com/dtd/web-app_2_3.dtd">
6:
7: <!-- =====
8:   Configuration file for Project 4, version 6.0
9:   Computer Science E-259
10: ===== -->
11:
12: <web-app>
13:
14:   <servlet>
15:     <servlet-name>cart</servlet-name>
16:     <servlet-class>cscie259.project4.scamazon.Cart</servlet-class>
17:   </servlet>
18:
19:   <servlet>
20:     <servlet-name>catalog</servlet-name>
21:     <servlet-class>cscie259.project4.scamazon.Catalog</servlet-class>
22:   </servlet>
23:
24:   <servlet>
25:     <servlet-name>ecs</servlet-name>
26:     <servlet-class>cscie259.project4.scamazon.ECS</servlet-class>
27:   </servlet>
28:
29:   <servlet>
30:     <servlet-name>receipt</servlet-name>
31:     <servlet-class>cscie259.project4.scamazon.Receipt</servlet-class>
32:   </servlet>
33:
34: </web-app>
```



```
1: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2:
3: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4:
5:     <!-- TODO (unless you create a DTD instead) -->
6:
7: </xsd:schema>
```

```
1: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2:
3: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4:
5:     <!-- TODO -->
6:
7: </xsd:schema>
```

```
1: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2:
3: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4:
5:     <!-- TODO (unless you create a DTD instead) -->
6:
7: </xsd:schema>
```

```
1: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2:
3: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4:
5:     <!-- TODO (unless you create a DTD instead) -->
6:
7: </xsd:schema>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!-- Any references to the local filesystem in XSLT instructions are resolved
4:      with respect to project4-6.0/webapps/scamazon/.
5:
6:      For instance, to access the contents of
7:      project4-6.0/webapps/scamazon/xml/foo.xml, you
8:      could use:
9:
10:         xsl:variable name="root" select="document('xml/foo.xml') "
11:
12:      To include project4-6.0/webapps/scamazon/xsl/bar.xsl, you
13:      could use:
14:
15:         xsl:include href="xsl/bar.xsl"
16:
17:      To import project4-6.0/webapps/scamazon/xsl/baz.xsl, you
18:      could use:
19:
20:         xsl:import href="xsl/baz.xsl"
21:
22:      However, any references to the local filesystem in XHTML are resolved
23:      with respect to project4-6.0/webapps/scamazon/xsl/.
24:
25:      For instance, to reference project4-6.0/webapps/scamazon/images/qux.gif,
26:      you could use:
27:
28:         img src="../../../images/qux.gif"
29: -->
30:
31: <!-- XSLT stylesheet for displaying contents of user's shopping cart -->
32: <xsl:stylesheet version="1.0" exclude-result-prefixes="xalan" xmlns:xalan="http://
xml.apache.org/xslt" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
33:
34:   <!-- output pretty-printed results as XHTML 1.0 -->
35:   <xsl:output doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-syst
em="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" encoding="UTF-8" indent="yes
" method="xml" xalan:indent-amount="4"/>
36:
37:   <xsl:template match="/">
38:     <html xml:lang="en" lang="en">
39:       <head>
40:         <title>Scamazon.com--Earth's Smallest Selection</title>
41:       </head>
42:       <body>
43:         <div align="center">
44:
45:           <!-- Scamazon.com's logo -->
46:           
47:           <br /><br />
48:
49:           <!-- TODO -->
50:
51:         </div>
52:       </body>
53:     </html>
54:   </xsl:template>
55:
56: </xsl:stylesheet>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!-- XSLT stylesheet for displaying contents of Scamazon.com's catalog -->
4:
5: <!-- Any references to the local filesystem in XSLT instructions are resolved
6:      with respect to project4-6.0/webapps/scamazon/.
7:
8:      For instance, to access the contents of
9:      project4-6.0/webapps/scamazon/xml/foo.xml, you
10:     could use:
11:
12:         xsl:variable name="root" select="document('xml/foo.xml')"
13:
14:     To include project4-6.0/webapps/scamazon/xsl/bar.xsl, you
15:     could use:
16:
17:         xsl:include href="xsl/bar.xsl"
18:
19:     To import project4-6.0/webapps/scamazon/xsl/baz.xsl, you
20:     could use:
21:
22:         xsl:import href="xsl/baz.xsl"
23:
24:     However, any references to the local filesystem in XHTML are resolved
25:     with respect to project4-6.0/webapps/scamazon/xsl/.
26:
27:     For instance, to reference project4-6.0/webapps/scamazon/images/qux.gif,
28:     you could use:
29:
30:         img src="../../images/qux.gif"
31: -->
32:
33: <xsl:stylesheet version="1.0" exclude-result-prefixes="xalan" xmlns:xalan="http://
xml.apache.org/xslt" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
34:
35:     <!-- output pretty-printed results as XHTML 1.0 -->
36:     <xsl:output doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-syst
em="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" encoding="UTF-8" indent="yes
" method="xml" xalan:indent-amount="4"/>
37:
38:     <xsl:template match="/">
39:         <html xml:lang="en" lang="en">
40:             <head>
41:                 <title>Scamazon.com--Earth's Smallest Selection</title>
42:             </head>
43:             <body>
44:                 <div align="center">
45:
46:                     <!-- Scamazon.com's logo -->
47:                     
48:                     <br /><br />
49:
50:                     <!-- TODO -->
51:
52:                 </div>
53:             </body>
54:         </html>
55:     </xsl:template>
56:
57: </xsl:stylesheet>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!-- XSLT stylesheet for confirming an order -->
4:
5: <!-- Any references to the local filesystem in XSLT instructions are resolved
6:      with respect to project4-6.0/webapps/scamazon/.
7:
8:      For instance, to access the contents of
9:      project4-6.0/webapps/scamazon/xml/foo.xml, you
10:     could use:
11:
12:         xsl:variable name="root" select="document('xml/foo.xml')"
13:
14:     To include project4-6.0/webapps/scamazon/xsl/bar.xsl, you
15:     could use:
16:
17:         xsl:include href="xsl/bar.xsl"
18:
19:     To import project4-6.0/webapps/scamazon/xsl/baz.xsl, you
20:     could use:
21:
22:         xsl:import href="xsl/baz.xsl"
23:
24:     However, any references to the local filesystem in XHTML are resolved
25:     with respect to project4-6.0/webapps/scamazon/xsl/.
26:
27:     For instance, to reference project4-6.0/webapps/scamazon/images/qux.gif,
28:     you could use:
29:
30:         img src="../../../images/qux.gif"
31: -->
32:
33: <xsl:stylesheet version="1.0" exclude-result-prefixes="xalan" xmlns:xalan="http://
xml.apache.org/xslt" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
34:
35:     <!-- output pretty-printed results as XHTML 1.0 -->
36:     <xsl:output doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-syst
em="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" encoding="UTF-8" indent="yes
" method="xml" xalan:indent-amount="4"/>
37:
38:     <xsl:template match="/">
39:         <html xml:lang="en" lang="en">
40:             <head>
41:                 <title>Scamazon.com--Earth's Smallest Selection</title>
42:             </head>
43:             <body>
44:                 <div align="center">
45:
46:                     <!-- Scamazon.com's logo -->
47:                     
48:                     <br /><br />
49:
50:                     <!-- TODO -->
51:
52:                 </div>
53:             </body>
54:         </html>
55:     </xsl:template>
56:
57: </xsl:stylesheet>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!-- Any references to the local filesystem in XSLT instructions are resolved
4:      with respect to project4-6.0/webapps/scamazon/.
5:
6:      For instance, to access the contents of
7:      project4-6.0/webapps/scamazon/xml/foo.xml, you
8:      could use:
9:
10:         xsl:variable name="root" select="document('xml/foo.xml')"
11:
12:      To include project4-6.0/webapps/scamazon/xsl/bar.xsl, you
13:      could use:
14:
15:         xsl:include href="xsl/bar.xsl"
16:
17:      To import project4-6.0/webapps/scamazon/xsl/baz.xsl, you
18:      could use:
19:
20:         xsl:import href="xsl/baz.xsl"
21:
22:      However, any references to the local filesystem in XHTML are resolved
23:      with respect to project4-6.0/webapps/scamazon/xsl/.
24:
25:      For instance, to reference project4-6.0/webapps/scamazon/images/qux.gif,
26:      you could use:
27:
28:         img src="../../images/qux.gif"
29: -->
30:
31: <!-- XSLT stylesheet for producing a receipt in the form of XSL-FO 1.0 -->
32: <xsl:stylesheet version="1.0" exclude-result-prefixes="xalan" xmlns:xalan="http://
xml.apache.org/xslt" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fo="http://ww
w.w3.org/1999/XSL/Format">
33:
34:   <!-- output pretty-printed XSL-FO 1.0 -->
35:   <xsl:output encoding="UTF-8" indent="yes" method="xml" xalan:indent-amount="4"/>
36:
37:   <xsl:template match="/">
38:
39:     <!-- TODO -->
40:
41:   </xsl:template>
42:
43: </xsl:stylesheet>
```



1: <!-- TODO -->

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-transitional.dtd">
4:
5: <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
6:   <head>
7:     <title>Warehouse</title>
8:   </head>
9:   <body>
10:     <center>
11:       <!-- from http://www.gatewaynmra.org/images/rws-no-warehouse.jpg -->
12:       
13:     </center>
14:   </body>
15: </html>
```

```
1: <?xml version="1.0" encoding="ISO-8859-1"?>
2:
3: <!DOCTYPE web-app
4:   PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
5:   "http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">
6:
7: <!-- =====
8:   Configuration file for Project 4, version 6.0
9:   Computer Science E-259
10:  ===== -->
11:
12: <web-app>
13:
14:   <display-name>Apache-Axis</display-name>
15:
16:   <servlet>
17:     <servlet-name>AxisServlet</servlet-name>
18:     <display-name>Apache-Axis Servlet</display-name>
19:     <servlet-class>
20:       org.apache.axis.transport.http.AxisServlet
21:     </servlet-class>
22:   </servlet>
23:
24:   <servlet>
25:     <servlet-name>SOAPMonitorService</servlet-name>
26:     <display-name>SOAPMonitorService</display-name>
27:     <servlet-class>
28:       org.apache.axis.monitor.SOAPMonitorService
29:     </servlet-class>
30:     <init-param>
31:       <param-name>SOAPMonitorPort</param-name>
32:       <param-value>5001</param-value>
33:     </init-param>
34:     <load-on-startup>100</load-on-startup>
35:   </servlet>
36:
37:   <servlet-mapping>
38:     <servlet-name>AxisServlet</servlet-name>
39:     <url-pattern>/servlet/AxisServlet</url-pattern>
40:   </servlet-mapping>
41:
42:   <servlet-mapping>
43:     <servlet-name>AxisServlet</servlet-name>
44:     <url-pattern>*.jws</url-pattern>
45:   </servlet-mapping>
46:
47:   <servlet-mapping>
48:     <servlet-name>AxisServlet</servlet-name>
49:     <url-pattern>/services/*</url-pattern>
50:   </servlet-mapping>
51:
52:   <servlet-mapping>
53:     <servlet-name>SOAPMonitorService</servlet-name>
54:     <url-pattern>/SOAPMonitor</url-pattern>
55:   </servlet-mapping>
56:
57:   <!-- currently the W3C havent settled on a media type for WSDL;
58:   http://www.w3.org/TR/2003/WD-wsdl12-20030303/#ietf-draft
59:   for now we go with the basic 'it's XML' response -->
60:   <mime-mapping>
61:     <extension>wsdl</extension>
62:     <mime-type>text/xml</mime-type>
63:   </mime-mapping>
64:
65:   <mime-mapping>
66:     <extension>xsd</extension>
67:     <mime-type>text/xml</mime-type>
68:   </mime-mapping>
69:
70:   <welcome-file-list id="WelcomeFileList">
71:     <welcome-file>index.html</welcome-file>
72:     <welcome-file>index.jsp</welcome-file>
73:     <welcome-file>index.jws</welcome-file>
74:   </welcome-file-list>
75:
76: </web-app>
77:
```

```
1: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2:
3: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4:
5:     <!-- TODO -->
6:
7: </xsd:schema>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!-- XSLT stylesheet for acknowledging an order -->
4:
5: <!-- Any references to the local filesystem in XSLT instructions are resolved
6:      with respect to project4-6.0/webapps/warehouse/xsl/.
7:
8:      For instance, to access the contents of
9:      project4-6.0/webapps/warehouse/xml/foo.xml, you
10:     could use:
11:
12:         xsl:variable name="root" select="document('../xml/foo.xml')"
13:
14:     To include project4-6.0/webapps/warehouse/xsl/bar.xsl, you
15:     could use:
16:
17:         xsl:include href="bar.xsl"
18:
19:     To import project4-6.0/webapps/warehouse/xsl/baz.xsl, you
20:     could use:
21:
22:         xsl:import href="baz.xsl"
23:
24: -->
25:
26: <xsl:stylesheet version="1.0" exclude-result-prefixes="xalan" xmlns:xalan="http://
xml.apache.org/xslt" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
27:
28:   <!-- output pretty-printed XML 1.0 -->
29:   <xsl:output encoding="UTF-8" indent="yes" method="xml" omit-xml-declaration="yes
" xalan:indent-amount="4"/>
30:
31:   <xsl:template match="/">
32:     <PO-ACK>
33:
34:       <!-- TODO -->
35:
36:     </PO-ACK>
37:
38:   </xsl:template>
39:
40: </xsl:stylesheet>
41:
```