

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!-- =====
4: Buildfile for Project 4, version 7.0
5: Computer Science E-259
6:
7: This buildfile "shipped" in the root of the following hierarchy.
8:
9: project4-7.0/
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:   images/
36:   WEB-INF/
37:   lib/
38:   xml/
39:   cache/
40:   xsd/
41:   xsl/
42:   warehouse/
43:   dtd/
44:   images/
45:   WEB-INF/
46:   xml/
47:   xsd/
48:   xsl/
49:
50: To compile Scamazon.com and its warehouse, execute 'ant compile' or
51: 'ant' from within project4-7.0/.
52:
53: To generate Javadoc for your code (in project4-7.0/webapps/ROOT/docs/),
54: execute 'ant javadoc' from within project4-7.0/.
55:
56: To publish Javadoc for your code at
57: http://www.people.fas.harvard.edu/~username/cscie259/javadoc/project4-7.0/,
58: where username is your FAS username, execute 'ant publish-javadoc'
59: from within project4-7.0/.
60:
61: To delete your bytecodes as well as Tomcat's logs and runtime
62: directories (plus Axis's runtime directories, if any),
63: execute 'ant clean' from within project4-7.0/.
64: ===== -->
65:
66: <project name="project4" default="compile" basedir=".">
67:
```

```
68:   <description>Project 4</description>
69:
70:   <!-- set global properties for this build -->
71:   <property name="attachments"
72:     location="webapps/warehouse/WEB-INF/attachments" />
73:   <property name="conf" location="conf" />
74:   <property name="docs" location="webapps/ROOT/docs" />
75:   <property name="jwsClasses"
76:     location="webapps/warehouse/WEB-INF/jwsClasses" />
77:   <property name="logs" location="logs" />
78:   <property name="servlets" location="webapps/scamazon/WEB-INF/classes" />
79:   <property name="services" location="webapps/warehouse/WEB-INF/classes" />
80:   <property name="src" location="src" />
81:   <property name="temp" location="temp" />
82:   <property name="work" location="work" />
83:
84:   <!-- init -->
85:   <target name="init">
86:
87:     <!-- set the standard DSTAMP, TSTAMP, and TODAY properties -->
88:     <!-- according to the default formats -->
89:     <tstamp/>
90:
91:     <!-- Create the build directories -->
92:     <mkdir dir="${servlets}" />
93:     <mkdir dir="${services}" />
94:
95:   </target>
96:
97:   <!-- compile -->
98:   <target name="compile"
99:     depends="init"
100:     description="compile Project 4">
101:     <javac srcdir="${src}"
102:       destdir="${servlets}"
103:       debug="true"
104:       fork="true"
105:       includes="com/amazon/soap/AWSECommerceService/**/*.java" />
106:     <javac srcdir="${src}"
107:       destdir="${servlets}"
108:       debug="true"
109:       fork="true"
110:       includes="cscie259/project4/scamazon/**/*.java" />
111:     <javac srcdir="${src}"
112:       destdir="${services}"
113:       debug="true"
114:       fork="true"
115:       includes="cscie259/project4/warehouse/**/*.java" />
116:   </target>
117:
118:   <!-- javadoc -->
119:   <target name="javadoc"
120:     description="generate Javadoc">
121:     <delete includeemptydirs="true">
122:       <fileset dir="${docs}" includes="**/*" />
123:     </delete>
124:     <javadoc packagenames="com.amazon.soap.AWSECommerceService.*,cscie259.proj
ect4.*"
125:       sourcepath="${src}"
126:       destdir="${docs}"
127:       author="true"
128:       version="true"
129:       private="true"
130:       nodeprecated="true"
131:       windowtitle="Project 4"
132:       header="Project 4" />
133:   </target>
```

```
134:
135:     <!-- publish javadoc -->
136:     <target name="publish-javadoc"
137:           depends="javadoc"
138:           description="publish Javadoc">
139:       <copy todir="${user.home}/public_html/cscie259/javadoc/project4-7.0">
140:         <fileset dir="${docs}" />
141:       </copy>
142:       <chmod parallel="false" perm="a+rX" dir="${user.home}/public_html"
143:         includes="**/*" type="both" />
144:     </target>
145:
146:     <!-- clean -->
147:     <target name="clean"
148:           description="remove class, log, runtime, temp, and work files">
149:       <delete dir="${attachments}" />
150:       <delete dir="${conf}/Catalina" />
151:       <delete dir="${jwsClasses}" />
152:       <delete dir="${logs}" />
153:       <delete dir="${services}" />
154:       <delete dir="${servlets}" />
155:       <delete includeemptydirs="true">
156:         <fileset dir="${temp}" includes="**/*" />
157:       </delete>
158:       <delete dir="${work}" />
159:     </target>
160:
161: </project>
```

```
1: <!-- =====
2:      Configuration file for Project 4, version 7.0
3:      Computer Science E-259
4:
5:      Be sure to set the value of the Server element's port
6:      attribute to an integer between 1024 and 65535, inclusive.
7:      Of course, that port cannot already be in use. Nor can it
8:      be the same value you choose for the Connector element's port.
9:
10:     Also be sure to set the value of the Connector element's port
11:     attribute to an integer between 1024 and 65535, inclusive.
12:     Of course, that port cannot already be in use. Nor can it
13:     be the same value you choose for the Server element's port.
14: ===== -->
15:
16: <Server port="" shutdown="SHUTDOWN">
17:
18:   <Listener className="org.apache.catalina.core.JasperListener" />
19:
20:   <GlobalNamingResources>
21:     <Resource name="UserDatabase" auth="Container"
22:       type="org.apache.catalina.UserDatabase"
23:       description="User database that can be updated and saved"
24:       factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
25:       pathname="conf/tomcat-users.xml" />
26:   </GlobalNamingResources>
27:
28:   <Service name="Catalina">
29:     <Connector port="" protocol="HTTP/1.1"
30:       maxThreads="150" connectionTimeout="20000" />
31:     <Engine name="Catalina" defaultHost="localhost">
32:       <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
33:         resourceName="UserDatabase"/>
34:       <Host name="localhost" appBase="webapps"
35:         unpackWARs="true" autoDeploy="true"
36:         xmlValidation="false" xmlNamespaceAware="false">
37:         <Context path=""
38:           docBase="ROOT" debug="0" reloadable="true"
39:           privileged="true">
40:           <Manager pathname="" />
41:         </Context>
42:         <Context path="/scamazon"
43:           docBase="scamazon" debug="0" reloadable="true"
44:           privileged="true">
45:           <Manager pathname="" />
46:         </Context>
47:         <Context path="/warehouse"
48:           docBase="warehouse" debug="0" reloadable="true"
49:           privileged="true">
50:           <Manager pathname="" />
51:         </Context>
52:       </Host>
53:     </Engine>
54:   </Service>
55: </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 project4-7.0/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 7.0
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 project4-7.0/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 7.0
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",
125:                                     false);
126:             }
127:
128:             // load XML
129:             SAXSource saxSource = new SAXSource(reader,
130:                                                  new InputSource(getServletContext().
131:                                                                  getResourceAsStream("/xml/catalog.xml")));
132:
133:             // transform!
134:             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: import java.util.regex.Pattern;
5:
6: import javax.servlet.ServletException;
7: import javax.servlet.http.HttpServletRequest;
8: import javax.servlet.http.HttpServletResponse;
9:
10: import com.amazon.soap.AWSECommerceService.*;
11:
12:
13: /**
14:  * This servlet implements a demonstration of Amazon's
15:  * E-Commerce Service (ECS).
16:  *
17:  * Be sure to assign ECS.ACCESS_KEY_ID the value of your
18:  * own Access Key ID.
19:  *
20:  * You MAY modify this file.
21:  *
22:  * @author Computer Science E-259
23:  * @version 7.0
24:  */
25:
26: public class ECS extends ScamazonServlet
27: {
28:     /* your Access Key ID */
29:     private static final String ACCESS_KEY_ID = "";
30:
31:     /* ISBNs of course's recommended books */
32:     private static final String ISBNs
33:         = "0130092290 | 0201740958 | 0735611831 | 0764543814";
34:
35:     /* response groups to request of ECS */
36:     private static final String [] RESPONSE_GROUPS = new String[]{"Large"};
37:
38:
39:     /**
40:      * Demonstrates ECS.
41:      *
42:      * @throws IOException if an IO-related error occurs
43:      * @throws ServletException if a servlet-related error occurs
44:      */
45:     protected void doWork(HttpServletRequest request,
46:                             HttpServletResponse response)
47:     throws ServletException, IOException
48:     {
49:         // check for Access Key ID
50:         if (ACCESS_KEY_ID.equals(""))
51:             throw new ServletException("Access Key ID was not provided.");
52:
53:         // prepare connection to ECS
54:         AWSECommerceService service = new AWSECommerceServiceLocator();
55:         AWSECommerceServicePortType port = null;
56:         try
57:         {
58:             port = service.getAWSECommerceServicePort();
59:         }
60:         catch (javax.xml.rpc.ServiceException e)
61:         {
62:             throw new ServletException("AWS Service failure", e);
63:         }
64:
65:         // prepare ItemSearchRequest for ECS
66:         ItemSearchRequest itemSearchRequest = new ItemSearchRequest();
67:         itemSearchRequest.setResponseGroup(RESPONSE_GROUPS);

```

[illegible]

```
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 7.0
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.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:  *
24:  * You MAY modify this file.
25:  *
26:  * @author Computer Science E-259
27:  * @version 7.0
28:  *
29:  * @author YOUR NAME GOES HERE
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:     /**
52:      * Responds to POSTs in the same manner as GETs.
53:      *
54:      * @param request HTTP request object
55:      * @param response HTTP response object
56:      *
57:      * @throws IOException if an IO-related error occurs
58:      * @throws ServletException if a servlet-related error occurs
59:      */
60:     public void doPost(HttpServletRequest request,
61:                        HttpServletResponse response)
62:     throws IOException, ServletException
63:     {
64:         doWork(request, response);
65:     }
66:
67:
```

```
68:     /**
69:      * To be implemented by descendants, handles GETs and POSTs identically.
70:      *
71:      * @param request HTTP request object
72:      * @param response HTTP response object
73:      *
74:      * @throws IOException if an IO-related error occurs
75:      * @throws ServletException if a servlet-related error occurs
76:      */
77:     protected abstract void doWork(HttpServletRequest request,
78:                                     HttpServletResponse response)
79:     throws IOException, ServletException;
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:      * project4-7.0/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 project4-7.0/.
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 7.0
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:     public String processPO(String poXmlString)
51:     {
52:         // storage for PO-ACK output
53:         StringWriter writer = new StringWriter();
54:
55:         // attempt to process PO
56:         try
57:         {
58:             // instantiate a transformer factory
59:             TransformerFactory tFactory = TransformerFactory.newInstance();
60:
61:             // instantiate ErrorListener
62:             AnErrorListener errorListener = new AnErrorListener();
63:
64:             // handle errors during processing of transformation instructions
65:             tFactory.setErrorListener(errorListener);
66:
67:             // load stylesheet

```

```

68:             StreamSource xslSource =
69:                 new StreamSource(warehouseDir + "/xsl/po-ack.xsl");
70:
71:             // instantiate transformer
72:             Transformer transformer = tFactory.newTransformer(xslSource);
73:
74:             // handle errors during the transformation
75:             transformer.setErrorListener(errorListener);
76:
77:             // create an XMLReader
78:             XMLReader reader = XMLReaderFactory.createXMLReader();
79:
80:             // enable validation as appropriate
81:             String validation = System.getProperty("enable.validation");
82:             if (validation != null && validation.equals("dtd"))
83:                 reader.setFeature("http://xml.org/sax/features/validation", true);
84:             else if (validation != null && validation.equals("xsd"))
85:             {
86:                 reader.setFeature("http://xml.org/sax/features/validation", true);
87:                 reader.setFeature("http://apache.org/xml/features/validation/schem
a", true);
88:             }
89:             else
90:                 reader.setFeature("http://xml.org/sax/features/validation", false);
91:
92:             // load XML
93:             SAXSource saxSource = new SAXSource(reader,
94:                 new InputSource(new ByteArrayInputStream(poXmlString.getBytes())));
95:
96:             // transform!
97:             transformer.transform(saxSource, new StreamResult(writer));
98:         }
99:         catch (Exception e)
100:         {
101:             e.printStackTrace();
102:             throw new RuntimeException("Can't process XSLT: " + e.getMessage()
);
103:         }
104:
105:         // return PO-ACK
106:         return writer.toString();
107:     }
108:
109:
110:     /**
111:      * Handle processing errors.
112:      */
113:     private class AnErrorListener implements ErrorListener
114:     {
115:         /**
116:          * Receive notification of a parser warning.
117:          *
118:          * @param e the exception thrown
119:          *
120:          * @throws TransformerException
121:          */
122:         public void warning(TransformerException e)
123:             throws TransformerException
124:         {
125:             System.out.println("Parsing warning: " + e.getMessage());
126:         }
127:
128:
129:         /**
130:          * Receive notification of a recoverable parser error.
131:          *

```

```
132:         * @param e the exception thrown
133:         *
134:         * @throws TransformerException
135:         */
136:     public void error(TransformerException e)
137:     throws TransformerException
138:     {
139:         System.out.println("Parsing error: " + e.getMessage());
140:     }
141:
142:
143:     /**
144:     * Report a fatal XML parsing error.
145:     *
146:     * @param e the exception thrown
147:     *
148:     * @throws TransformerException
149:     */
150:     public void fatalError(TransformerException e)
151:     throws TransformerException
152:     {
153:         System.out.println("Fatal parsing error: " + e.getMessage());
154:     }
155: }
156: }
```

```
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="ISO-8859-1"?>
2:
3: <!-- =====
4:      Configuration file for Project 4, version 7.0
5:      Computer Science E-259
6: ===== -->
7:
8: <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
9:      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
10:      xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns
11: /j2ee/web-app_2_5.xsd"
12:      version="2.5"/>
```

```
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 7.0
7:     Computer Science E-259
8:     ===== --%>
9:
10: <%@ page language="java" %>
11: <%@ taglib uri="http://java.sun.com/jstl/core_rt" 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="ISO-8859-1"?>
2:
3: <!-- =====
4:      Configuration file for Project 4, version 7.0
5:      Computer Science E-259
6: ===== -->
7:
8: <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
9:      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
10:      xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns
11: /j2ee/web-app_2_5.xsd"
12:      version="2.5">
13:
14:   <!-- servlets -->
15:   <servlet>
16:     <servlet-name>cart</servlet-name>
17:     <servlet-class>cscie259.project4.scamazon.Cart</servlet-class>
18:   </servlet>
19:   <servlet>
20:     <servlet-name>catalog</servlet-name>
21:     <servlet-class>cscie259.project4.scamazon.Catalog</servlet-class>
22:   </servlet>
23:   <servlet>
24:     <servlet-name>ecs</servlet-name>
25:     <servlet-class>cscie259.project4.scamazon.ECS</servlet-class>
26:   </servlet>
27: </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 project4-7.0/webapps/scamazon/.
5:
6:      For instance, to access the contents of
7:      project4-7.0/webapps/scamazon/xml/foo.xml, you
8:      could use:
9:
10:         xsl:variable name="root" select="document('xml/foo.xml')"
11:
12:      To include project4-7.0/webapps/scamazon/xsl/bar.xsl, you
13:      could use:
14:
15:         xsl:include href="xsl/bar.xsl"
16:
17:      To import project4-7.0/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 project4-7.0/webapps/scamazon/xsl/.
24:
25:      For instance, to reference project4-7.0/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 project4-7.0/webapps/scamazon/.
7:
8:      For instance, to access the contents of
9:      project4-7.0/webapps/scamazon/xml/foo.xml, you
10:     could use:
11:
12:         xsl:variable name="root" select="document('xml/foo.xml')"
13:
14:     To include project4-7.0/webapps/scamazon/xsl/bar.xsl, you
15:     could use:
16:
17:         xsl:include href="xsl/bar.xsl"
18:
19:     To import project4-7.0/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 project4-7.0/webapps/scamazon/xsl/.
26:
27:     For instance, to reference project4-7.0/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 project4-7.0/webapps/scamazon/.
7:
8:      For instance, to access the contents of
9:      project4-7.0/webapps/scamazon/xml/foo.xml, you
10:     could use:
11:
12:         xsl:variable name="root" select="document('xml/foo.xml')"
13:
14:     To include project4-7.0/webapps/scamazon/xsl/bar.xsl, you
15:     could use:
16:
17:         xsl:include href="xsl/bar.xsl"
18:
19:     To import project4-7.0/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 project4-7.0/webapps/scamazon/xsl/.
26:
27:     For instance, to reference project4-7.0/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: <!-- 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: <!-- =====
4:     Configuration file for Project 4, version 7.0
5:     Computer Science E-259
6: ===== -->
7:
8: <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
9:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
10:     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns
11: /j2ee/web-app_2_5.xsd"
12:     version="2.5">
13:     <display-name>Apache-Axis</display-name>
14:
15:     <listener>
16:         <listener-class>org.apache.axis.transport.http.AxisHTTPSessionListener</li
stener-class>
17:     </listener>
18:
19:     <servlet>
20:         <servlet-name>AxisServlet</servlet-name>
21:         <display-name>Apache-Axis Servlet</display-name>
22:         <servlet-class>
23:             org.apache.axis.transport.http.AxisServlet
24:         </servlet-class>
25:     </servlet>
26:
27:     <servlet>
28:         <servlet-name>AdminServlet</servlet-name>
29:         <display-name>Axis Admin Servlet</display-name>
30:         <servlet-class>
31:             org.apache.axis.transport.http.AdminServlet
32:         </servlet-class>
33:         <load-on-startup>100</load-on-startup>
34:     </servlet>
35:
36:     <servlet>
37:         <servlet-name>SOAPMonitorService</servlet-name>
38:         <display-name>SOAPMonitorService</display-name>
39:         <servlet-class>
40:             org.apache.axis.monitor.SOAPMonitorService
41:         </servlet-class>
42:         <init-param>
43:             <param-name>SOAPMonitorPort</param-name>
44:             <param-value>5001</param-value>
45:         </init-param>
46:         <load-on-startup>100</load-on-startup>
47:     </servlet>
48:
49:     <servlet-mapping>
50:         <servlet-name>AxisServlet</servlet-name>
51:         <url-pattern>/servlet/AxisServlet</url-pattern>
52:     </servlet-mapping>
53:
54:     <servlet-mapping>
55:         <servlet-name>AxisServlet</servlet-name>
56:         <url-pattern>*.jws</url-pattern>
57:     </servlet-mapping>
58:
59:     <servlet-mapping>
60:         <servlet-name>AxisServlet</servlet-name>
61:         <url-pattern>/services/*</url-pattern>
62:     </servlet-mapping>
63:
64:     <servlet-mapping>
65:         <servlet-name>SOAPMonitorService</servlet-name>
66:         <url-pattern>/SOAPMonitor</url-pattern>
67:     </servlet-mapping>
68:
69:     <!-- uncomment this if you want the admin servlet -->
70:     <!--
71:     <servlet-mapping>
72:         <servlet-name>AdminServlet</servlet-name>
73:         <url-pattern>/servlet/AdminServlet</url-pattern>
74:     </servlet-mapping>
75: -->
76:
77:     <session-config>
78:         <!-- Default to 5 minute session timeouts -->
79:         <session-timeout>5</session-timeout>
80:     </session-config>
81:
82:     <!-- currently the W3C havent settled on a media type for WSDL;
83:     http://www.w3.org/TR/2003/WD-wsdl12-20030303/#ietf-draft
84:     for now we go with the basic 'it's XML' response -->
85:     <mime-mapping>
86:         <extension>wsdl</extension>
87:         <mime-type>text/xml</mime-type>
88:     </mime-mapping>
89:
90:
91:     <mime-mapping>
92:         <extension>xsd</extension>
93:         <mime-type>text/xml</mime-type>
94:     </mime-mapping>
95:
96:     <welcome-file-list id="WelcomeFileList">
97:         <welcome-file>index.jsp</welcome-file>
98:         <welcome-file>index.html</welcome-file>
99:         <welcome-file>index.jws</welcome-file>
100:     </welcome-file-list>
101:
102: </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 -->
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 project4-7.0/webapps/warehouse/xsl/.
7:
8:      For instance, to access the contents of
9:      project4-7.0/webapps/warehouse/xml/foo.xml, you
10:     could use:
11:
12:         xsl:variable name="root" select="document('../xml/foo.xml')"
13:
14:     To include project4-7.0/webapps/warehouse/xsl/bar.xsl, you
15:     could use:
16:
17:         xsl:include href="bar.xsl"
18:
19:     To import project4-7.0/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:
```