Tag Libraries

What are Tag Libraries

- Collections of custom JSP tags
  - Often looking like scripting languages
- Load one of the many on the web
- Or build your own
  - Unlikely given the plenty of choices available

JSTL: The most common tag lib

- Available at jakarta.apache.org
- Core lib (c.tld): scripting language
- Database lib (sql.tld): Support for database/jsp programming
- Installation:
  - Place jstl.jar and standard.jar at application’s lib or <home>/lib
  - Place tld files in WEB-INF
JSTL Expressions

- Many JSTL tags have attributes whose values are JSTL expressions
  - Enclosed in `${...}`
  - `<c:out value='${request.v}'/>`
  - is shorthand for
    - `<% String attr=request.getParameter("v");%>`
    - `<%= v %>`
- Declare on top of your JSP
  - `<%@ taglib uri="/jstl-c" prefix="c"%>`
  - Put in your WEB-INF/web.xml
  - `<taglib>
      <taglib-uri>/jstl-c</taglib-uri>
      <taglib-location>/WEB-INF/c.tld</taglib-location>
    </taglib>`

Access to Bean Properties

- Assume your JSP uses a bean named `myBean` that has property `prop` and corresponding `getProp` and `setProp`
- `<c:out value='${myBean.prop}'/>` stands for `<%= myBean.getProp() %>`

Scripting Language Features

- Variable definition
  - `<c:set var='value' value='${...}' scope='session'/>`
  - `<c:remove var='v'/>`
- Weak typing
  - Strings to numbers
  - Integers to reals
  - And many more
Flow Control Tags

- `<c:if test='${boolean expression}'>`
  body of tag evaluated if boolean is true
- `<c:choose>`
  `<c:when test='${boolean expression}'>`
  …
  `<c:when>`
  …
  `<c:when test='${boolean expression}'>`
  …
  `<c:when>`
  `<c:choose>`

forEach

- Iterate over start to end value, arrays, Collection, Iterator, Enumeration, Map, comma separated string
- `<c:forEach var='i' begin='0' end='10' step='1'>`
  loop body
  `<c:forEach>`
- `<c:forEach var='ck' items='${pageContext.request.cookies}'>`
  loop body
  `<c:forEach>`

JSTL support for databases

- Obtain a Result, say, r
  assume Connection conn is available
  String sqlString;
  PreparedStatement pstmt = null;
  ResultSet resultSet = null;
  try {
    sqlString = "some query goes here";
    pstmt = conn.prepareStatement(sqlString);
    resultSet = pstmt.executeQuery();
    r = ResultSupport.toResult(resultSet); }…
- `<c:forEach var='tuple' items='${r.rows}'>`
  loop body
  `<c:forEach>`