

Session 12

Expression Language (EL)

1

Reading

■ Reading

- Head First - pages 368-401
- Sun Java EE 5 - Chapter 5 in the Tutorial
java.sun.com/javase/5/docs/tutorial/doc/JavaEETutorial.pdf

■ Reference

- JSTL Reference (Chapter 3 and Appendix A) - link on CSE336 Web site (References Section)
- JSP 2.1 Specification - link on CSE336 Web site (References Section) - This document contains a good description of EL (Chapter 2)
- EL Specification - same link as the EL spec

<http://www-128.ibm.com/developerworks/java/library/j-jstl0211.html>

www.onjava.com/pub/a/onjava/2003/11/05/jsp.html

© Robert Kelly, 2001-2009

2

Lecture Objectives

- Understand how EL can reduce the use of Java (e.g., scriptlets and JSP Expressions) in JSPs
- Understand how to use EL to reference object properties
- Understand how to combine references so that you can obtain properties of properties
- Know that an EL reference will look for the matching property in one of the shared objects
- Know the implicit objects available to you in EL
- Understand the type structure of Map objects
- Understand that objects containing matching get and set methods are considered to have properties

© Robert Kelly, 2001-2009

3

Example - EL Counter

- Let's look at making the counter example better
- We can use EL to replace the getProperty tag with an EL expression
- Previous approach:

```

<p>The counter is initially:
    <jsp:getProperty name="b"
        property="count" /></p>
<p>The counter is now:
    <%= b.fetchAndAdd() %> </p>
<p>The counter is now:
    <jsp:getProperty name="b"
        property="count" /> </p>
    
```



© Robert Kelly, 2001-2009

4

EL Variables

`${product}`

- The web container evaluates a variable that appears in an expression by looking up its value
- For example, when evaluating the expression `${product}`, the container will look for the name "product" in the page, request, session, and application scopes and will return its value (in the first scope in which it encounters the value)
- If product is not found, null is returned. A variable that matches one of the implicit objects will return that implicit object instead of the variable's value

© Robert Kelly, 2001-2009

7

Counter Example Revisited

```
<jsp:useBean id="b" class="lectures.CountBean2"
scope="application" />
...
<p>The counter is initially:
  ${b.count} </p>
<p>The counter is now:
  <%= b.fetchAndAdd() %> </p>
```

- Let's look at the example again
- How does EL find the bean?
- How does EL get the value of count?
- What would happen if we removed the `useBean` statement?

© Robert Kelly, 2001-2009

8

What if Your Bean Contains Objects?

```
public class CountBean2 implements java.io.Serializable {
    private int count = 0;
    private Calendar calendar = new GregorianCalendar();

    public int getCount() {
        return (count); }
    public int fetchAndAdd() {
        int temp=count;
        count++;
        return (temp); }
    public void setCount(int newCount) {
        this.count = count; }
    public Calendar getCalendar() {
        return (calendar); }
    public void setCalendar(Calendar newCalendar) {
        this.calendar = newCalendar; }
}
```

Today is: `<jsp:getProperty name="b" property="calendar" >`

Can your JSP include a reference to the
TimeZone property of the calendar?

© Robert Kelly, 2001-2009

9

With EL You Can

```
<p>and by the way, the time zone is now  
    ${b.calendar.timeZone.displayName} </p>
```

- The above JSP code will cause the browser to display:

More on the
EL syntax
later

JSP Counter

This JSP will print and increment the value of the counter

The counter is initially: 10

The counter is now: 10

The counter is now: 11

and by the way, the time zone is now Eastern Standard Time

[Re-count](#)

© Robert Kelly, 2001-2009

10

EL Syntax

This is the dot (.) operator

Today is: `#{b.d.hours}`

■ EL Implicit Object

- pageScope
- requestScope
- sessionScope
- applicationScope
- param
- paramValues
- header
- headerValues
- cookie
- initParam
- pageContext

■ Attribute name

- In pageScope
- In requestScope
- In sessionScope
- In applicationScope

Either a map key or a bean property (depending on whether b is a map or a bean)

A variable on the left side of a dot is either a Map (something with keys) or a bean (something with properties)

b.d evaluates to either a Map value or a bean property value (d is either a Map name or a bean property name)


11

© Robert Kelly, 2001-2009

Are We on Track?

- Using your laptop, code a JSP that displays the servlet path (in your request object)
 - Use a scriptlet to store your request object in the session as an attribute
 - Use EL to display the result
- Not everyone has a laptop, so please work in groups.

Remember, your request object exhibits "bean-like" behavior



12

© Robert Kelly, 2001-2009

Were We on Track?

```
<html>
  <head>
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>EL Sample Code</h1>
    <% session.setAttribute("r", request); %>
    <p>
      ${r.servletPath}
    </p>
  </body>
</html>
```

© Robert Kelly, 2001-2009

13

Map

- A Map is an object that maps keys to values
 - Cannot contain duplicate keys
 - Each key can map to at most one value
 - Contains a set of Map.Entry objects
- A Map.Entry object has 2 properties
 - key - Object representing the key under which this item is stored
 - value - Object representing the value corresponding to the key

A Map is an object that implements the Map interface and is instantiated as a HashMap, Hashtable, TreeMap, etc.

© Robert Kelly, 2001-2009

14

EL Implicit Objects

- The EL implicit objects are not the same as the JSP implicit objects (except pageContext)
- Example: sessionScope is a Map of session attributes
- Example: param is a Map of the request parameters
- Example: paramValues is a Map of the request parameters (with possibly more than one value per name)

When do you have an HTML parameter with more than one value?

How do we access paramValues in EL?

© Robert Kelly, 2001-2009

15

EL [] Operator

- The [] operator is more powerful than the dot operator
- These EL expressions are equivalent:

`${header.host}` ↔ `${header["host"]}`

- Are these EL expressions equivalent?

`${header.User-Agent}` ? ↔ `${header["User-Agent"]}`

When you use the dot operator, header can be a bean or a Map

When you use the dot operator, the name on the right must be a legal Java name

When you use the [] operator, the identifier on the left can be either a bean, a Map, a List or an Array

© Robert Kelly, 2001-2009

16

EL [] Operator

- Meaning of a String parameter in []
 - Map - MapEntry key (i.e., name in one of the name value pairs)
 - Bean - bean property
 - Array - index into the array
 - List - index into the list

`${headerValues.Accept["0"]}` ↔ `${headerValues.Accept[0]}`

The array index is coerced to an int

If the parameter is not a String, the parameter is evaluated (e.g. check shared objects)

© Robert Kelly, 2001-2009

17

Example

- If png is the first MIME type accepted by the browser

`${header.Accept}` and `${headerValues.Accept["0"]}`

Should evaluate to

`image/png`

© Robert Kelly, 2001-2009

18

Implicit Objects for JSP and EL

- Most EL and JSP implicit objects are not the same
- All but one EL implicit object is a Map
 - `sessionScope` (the EL implicit object) is a Map of Session attributes
 - `session` (the JSP implicit object) is the Session object

How would EL evaluate
`${sessionScope.maxInactiveInterval}` ?

© Robert Kelly, 2001-2009

19

Example (from Head First)

- Q: What is the text plus EL that is equivalent to:
Host is: `<%= request.getHeader("host") %>`

- Answer:
Host is: `${header["host"]}`
or
Host is: `${header.host}`

What is the meaning of
`Host is: ${header[host]}`

Hint: `header[host]` is not the same as `header["host"]`

© Robert Kelly, 2001-2009

20

pageContext Implicit Object

- The pageContext implicit EL object refers to the real pageContext object
- You can treat it as you would a bean ("bean-like" behavior)
- What is the meaning of

`${pageContext.request.method}`

What does the above expression evaluate to?

What does `${requestScope.method}` evaluate to?

© Robert Kelly, 2001-2009

21

Example (from Head First)

- Display the value of the cookie with name x
- JSP

```
<% Cookie[] cookies = request.getCookies();  
   for (int i=0; i<cookies.length; i++)  
       if ((cookies[i].getName()).equals("x")) {  
           out.println(cookies[i].getValue());  
       }  
   %>
```

- EL
- This is a Map of
cookieName/cookieObject

`${cookie.x.value}`

This is a Cookie object, which looks like a bean.
(Why?)

© Robert Kelly, 2001-2009

22

Are We on Track?

- Using your laptop, code a JSP that displays the server info (in your servlet context object)
 - Use a scriptlet to store your session object in the session as an attribute
 - Use EL to display the result
- Not everyone has a laptop, so please work in groups.

You will need to refer to your servlet context from your session



© Robert Kelly, 2001-2009

23

Were We on Track?

```
<html>
  <head>
    <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>EL Sample Code</h1>
    <% session.setAttribute("s", session);%>
    <p>
      ${s.servletContext.serverInfo}
    </p>
  </body>
</html>
```

© Robert Kelly, 2001-2009

24

EL Literals

- The JSP expression language defines the following literals:
 - Boolean: true and false
 - Integer: as in Java
 - Floating point: as in Java
 - String: with single and double quotes; " is escaped as \", ' is escaped as \', and \ is escaped as \\.
 - Null: null

© Robert Kelly, 2001-2009

25

EL Operators

- In addition to the . and [] operators, EL provides:
 - Arithmetic: +, - (binary), *, / and div, % and mod, - (unary)
 - Logical: and, &&, or, ||, not, !
 - Relational: ==, eq, !=, ne, <, lt, >, gt, <=, ge, >=, le.
(comparisons can be made against other values, or against boolean, string, integer, or floating point literals)
 - Empty: The empty operator is a prefix operation that can be used to determine whether a value is null or empty.
 - Conditional: A ? B : C. Evaluate B or C, depending on the result of the evaluation of A.

Example: `${!empty cookie.userName}`

© Robert Kelly, 2001-2009

26

Have You Satisfied the Lecture Objectives?

- Understand how EL can reduce the use of Java (e.g., scriptlets and JSP Expressions) in JSPs
- Understand how to use EL to reference object properties
- Understand how to combine references so that you can obtain properties of properties
- Know that an EL reference will look for the matching property in one of the shared objects
- Know the implicit objects available to you in EL
- Understand the type structure of Map objects
- Understand that objects containing matching get and set methods are considered to have properties

© Robert Kelly, 2001-2009

27

Assignment 4a

- When you first attempted HW#4a, you populated the JSP form components with getProperty tags. Now you can populate them with EL statements,

© Robert Kelly, 2001-2009

28