Persistent

# Java Server Pages II

Writing script free JSP

# Agenda

- EL
  - Understand how EL can reduce the use of script lets and expressions
  - Understand how to use EL to reference object properties and map keys
  - Know the implicit objects available to you in EL
- JSTL
  - Core
  - Internationalization
  - SQL
  - Functions
- Standard actions

# Example

- System definition/design
  - Various items are auctioned on a website.
  - The item bean has attributes like item name, description and starting price.

- Requirement
  - The programmer has created a form to fill up details of an item.
  - He needs to write a servlet which extracts the form’s data, populates the bean and forwards to a JSP where the item details will be displayed.

Confidential © 2020 Persistent Systems

# The servlet

```
public void doPost(HttpServletRequest req,
 HttpServletResponse res)
{
    Item item = new Item();
    item.setItemName(
                request.getParameter(“itemName”));
    item.setDescription(
                request.getParameter(“description”));
    // some more code

    request.setAtrribute(“ITEM”, item);

    req.getRequestDispatcher(“some-page.jsp”)
    .forward(req, res);
}
```

Confidential © 2020 Persistent Systems

Persistent

## The view

- **With scripting**

  <html>……..

  <%=((com.someapp.bean.Item)request.getAttribute(“ITEM”)).getItemName() %>

  <%=((com.someapp.bean.Item)request.getAttribute(“ITEM”)).getDescription() %> ……

  </html>

- **With JSP standard actions**

  <html>……..

  <jsp:useBean  id=“ITEM” class=“com.someapp.bean.Item” scope=“request”/>

  <jsp:getProperty name=“ITEM” property=“itemName”/>

  <jsp:getProperty name=“ITEM” property=“description”/>……

  </html>

- **With EL**

  <html>……..

  ${ITEM.itemName}……

  </html>

Confidential © 2020 Persistent Systems

# Example contd.

- System definition/design
  - A user can auction an item. The auction bean has attributes like item, start date, end date and minimum increment. An auction bean is composed of an item bean.

- Requirement
  - The programmer has created a form to fill up details of an auction. He needs to write a servlet which extracts the form’s data, populates the bean and forwards to a JSP where the auction and item details will be displayed.

Confidential © 2020 Persistent Systems

## The servlet

```
public static void doPost(HttpServletRequest req, HttpServletResponse res)
{
    Auction auction = new Auction();
    auction.setItem(new Item());
    auction.getItem().setItemId(request.getParameter(“itemId”));
    auction.setStartDate(request.getParameter(“startDate”));
    auction.setEndDate(request.getParameter(“endDate”));
    // some more code
    // code to retrieve the item details from a database
    request.setAtrribute(“AUCTION”, auction);
    req.getRequestDispatcher(“some-page.jsp”).forward(req, res);
}
```

Confidential © 2020 Persistent Systems

# The view

- **With scripting**

  <html>……..

  <%=((Auction)request.getAttribute(“AUCTION”)).getStartDate()%>

  <%=((Auction)request.getAttribute(“AUCTION”)).getEndDate()%>

  <%=((Auction)request.getAttribute(“AUCTION”)).getItem().getItemName()%>……

  </html>

- **With JSP standard actions**

  <html>……..

  <jsp:useBean  id=“AUCTION” class=“com.someapp.bean.Auction” scope=“request”/>

  <jsp:getProperty name=“AUCTION” property=“startDate”/>

  <jsp:getProperty name=“AUCTION” property=“endDate”/>

  <jsp:getProperty name=“AUCTION” property=“item”/>……

  </html>

- **With EL** <html>……..

  ${AUCTION.item.itemName}……

  </html>

**What we wanted:**
Apple IPhone
**What we got:**
`com.someapp.bean.Item@109238`

Confidential © 2020 Persistent Systems

Persistent

## EL variables

${AUCTION}

- The container evaluates a variable that appears in an expression by looking up its value
- For example, when evaluating the expression ${AUCTION}, the container will look for the name “AUCTION” in the page, request, session, and application scopes and will return its value
- If it is not found, null is returned.
- EL has its own set of implicit objects. A variable that matches one of the implicit objects will return that implicit object instead of the variable's value

Confidential © 2020 Persistent Systems

# EL implicit objects

```
<p> Item name: ${AUCTION.item.itemName} </p>
```

**EL implicit object**

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

Either a key in a map or property of a bean depending upon whether `AUCTION` is a ma p or a bean

- A name on the left-hand side of a dot evaluates to either a map or a bean

- A name on the right-hand side of a dot is either a key in a map or property of a bean.

**OR**

**Attribute in**

- page scope i.e. `javax.servlet.jsp.PageContext`
- request scope i.e. `javax.servlet.http.HttpServletRequest`
- session scope i.e. `javax.servlet.http.HttpSession`
- application scope i.e. `javax.servlet.ServletContext`

Confidential © 2020 Persistent Systems

# EL implicit objects contd….

- Not all EL implicit objects are the same as the JSP implicit objects except pageContext

- For example,
  - sessionScope is a map of session attributes
  - param is a map of the request parameters
  - paramValues is a map of the request parameters (with possibly more than one value per name)

Confidential © 2020 Persistent Systems

# The [ ] operator

- The [ ] operator is more powerful than the dot operator.
- The following EL expressions are equivalent

  ${header.host}

  ${header[“host”]}

- When you use the dot operator
  - name on the left must evaluate to either a map or a bean
  - name on the right must be a legal Java name
- When you use the [ ] operator
  - name on the left can be evaluate to any one of either a map, bean,  list or an array.

Confidential © 2020 Persistent Systems

# The [ ] operator contd…..

- Meaning of the string parameter in [ ] :
  - Map – key
  - Bean – property
  - Array – index into the array
  - List – index into the list

Confidential © 2020 Persistent Systems

## Using [ ] with an array or arraylist

- **In a servlet**

```
String [] movies = {
                                        “Life is beautiful”,
                                        “Children of heaven”,
                                        “The pursuit of happyness”,
                                        “Babel”
                        };
        request.setAttribute(“movies”, movies);
```

- **In the JSP**

```
First movie is ${movies[0]}
```

**o/p:** First movie is Life is beautiful

```
Second movie is ${movies[“1”]}
```

**o/p:** Second movie is Children of heaven

Confidential © 2020 Persistent Systems

## Using either operator with beans and maps

- **In a servlet**

```
Map<String, String> stateHeads = new HashMap<String, String>();
stateHeads.put(“Afghanistan”, ”Hamid Karzai”);
stateHeads.put(“Bhutan”, ”J.K.N. Wangchuk”);
stateHeads.put(“Djibouti”, ”Omar Guelleh”);
stateHeads.put(“Nepal”, ”Ram Baran Yadav”);
request.setAttribute(“stateHeads”, stateHeads);
```

- **In the JSP**

Head of state of Bhutan is ${stateHeads.Bhutan}

**o/p:** Head of state of Bhutan is J.K.N. Wangchuk

Head of state of Nepal is ${stateHeads[“Nepal”]}

**o/p:** Head of state of Nepal is Ram Baran Yadav

Confidential © 2020 Persistent Systems

# Nested expressions

## In a servlet

```
Map<String, String> stateHeads = new HashMap<String, String>();
stateHeads.put(“Afghanistan”, ”Hamid Karzai”);
stateHeads.put(“Bhutan”, ”J.K.N. Wangchuk”);
stateHeads.put(“Djibouti”, ”Omar Guelleh”);
stateHeads.put(“Nepal”, ”Ram Baran Yadav”);
String [] countries = { “Afghanistan”, “Bhutan”, “Djibouti”,  “Nepal” };
request.setAttribute(“countries”, countries);
request.setAttribute(“stateHeads”, stateHeads);
```

## In the JSP

```
Head of state is ${stateHeads[countries[2]]}
Head of state is ${stateHeads[“Djibouti”]}
```

**o/p:** Head of state is Omar Guelleh

Confidential © 2020 Persistent Systems

# Accessing context init. params.

**In web.xml**

```
<context-param>
    <param-name>WEBMASTER_EMAIL</param-name>
    <param-value>master@email.web</param-value>
</context-param>
```

**In JSP with scripting**

```
Email is:
<%=application.getInitParameter("WEBMASTER_EMAIL")%>
```

**In JSP with EL**

```
Email is: ${initParam.WEBMASTER_EMAIL}
```

Confidential © 2020 Persistent Systems

# 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 \’
    - \ is escaped as \\
  - Null
    - null

Confidential © 2020 Persistent Systems

# EL operators

- In addition to the **.** and [ ] operators, EL provides:
  - **Arithmetic**

    + - * / and div % and mod
    - (unary)

  - **Logical**

    and &&
    or ||
    not !

  - **Relational**

    == eq != ne < lt > gt
    <= ge >= le

  - **empty**
    - The empty operator is a prefix operation that can be used to determine whether a value is null or empty.

    ${!empty sessionScope.USER.userName}

  - **Conditional**

    A ? B : C. Evaluate B or C, depending on the result of the evaluation of A.

Confidential © 2020 Persistent Systems

# Introducing JSTL

- The Java Server Pages Standard Tag Library encapsulates, as simple tags, core functionality common to many JSP applications.

- JSTL has support for common tasks such as iteration and conditionals, manipulating XML documents, internationalization, and SQL tags.

- JSTL makes it easy for page designers to write presentation logic on a JSP rather than using scripting.

- JSTL is not part of the Servlet and JSP APIs. Before you can use JSTL, you need to put two files, jstl.jar and standard.jar into the WEB-INF/lib directory of your web app.

Confidential © 2020 Persistent Systems

## So, what is the problem with scripting ?

- Ugly mash-up of Java code with html tags can make your pages a maintenance nightmare.
- Poor separation of concerns especially with regards to larger project teams.
- Scriptlets are not unit-testable.
- Scriptlets are not re-usable

Confidential © 2020 Persistent Systems

# JSTL tag libraries

Core

Variable support, flow control and url management

<@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

XML

Core, flow control, transformation

<@taglib prefix="x" uri="http://java.sun.com/jsp/jstl/xml" %>

Internationalization

Locale, message formatting, number and date formatting

<@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>

Database

SQL query and update

<@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>

Functions

Collection length, string manipulation

<@taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>

Confidential © 2020 Persistent Systems

# Core tags – variable support - `<c:set>` - Part I

- Make a new entry in a map or set a value in a map.

**In a servlet**

```
Map<String,String> passports = new HashMap<String,String>();
passports.put("IND2398732","Sonia Gandhi");
passports.put("ITA7895092","Sonia Gandhi");
passports.put("CO4565626","Robert Vadra");
session.setAttribute("PASSPORTS", passports);
```

**In a JSP**

```
<c:set target="${sessionScope.PASSPORTS}" property="ITA7895092"
        value="Rahul Gandhi"/>
```

Must evaluate to either a bean or a map. In this case evaluates to a map

If `target` is a bean, this must evaluate to the name of a property. If `target` is a map this is treated as value of either an existing or new key.

If `target` is a bean, this is the value of its specified property. If `target` is a map this is the value of the specified key.

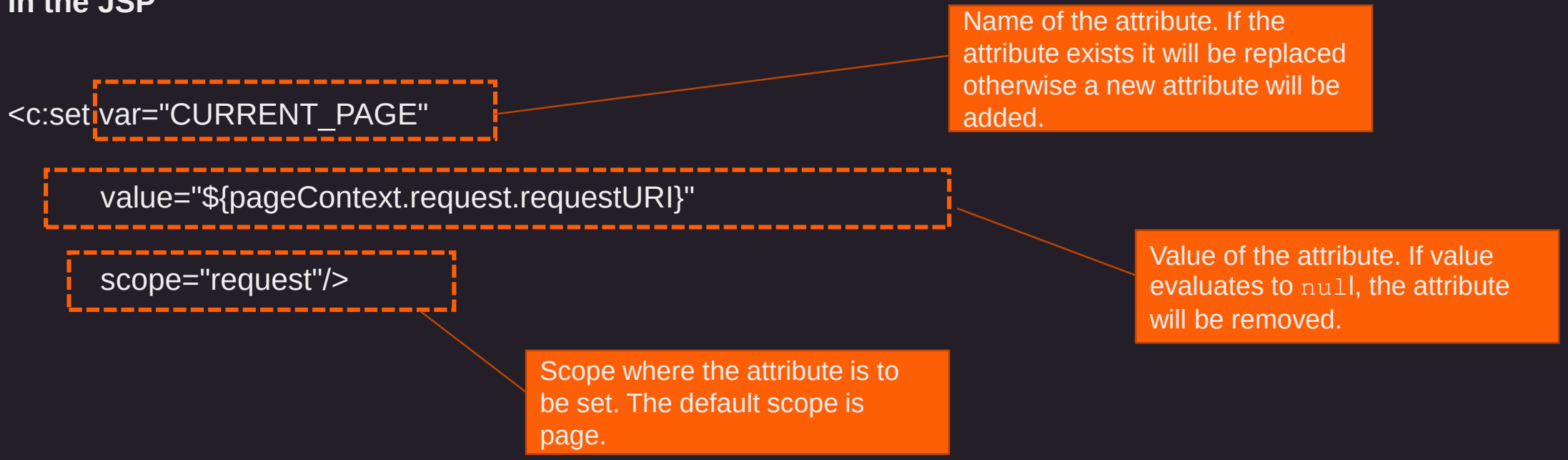
Confidential © 2020 Persistent Systems

Persistent

# Core tags – variable support - `<c:set>` - Part II

- Set or create a attribute in any one of either page, request, session or application scope.

**In the JSP**

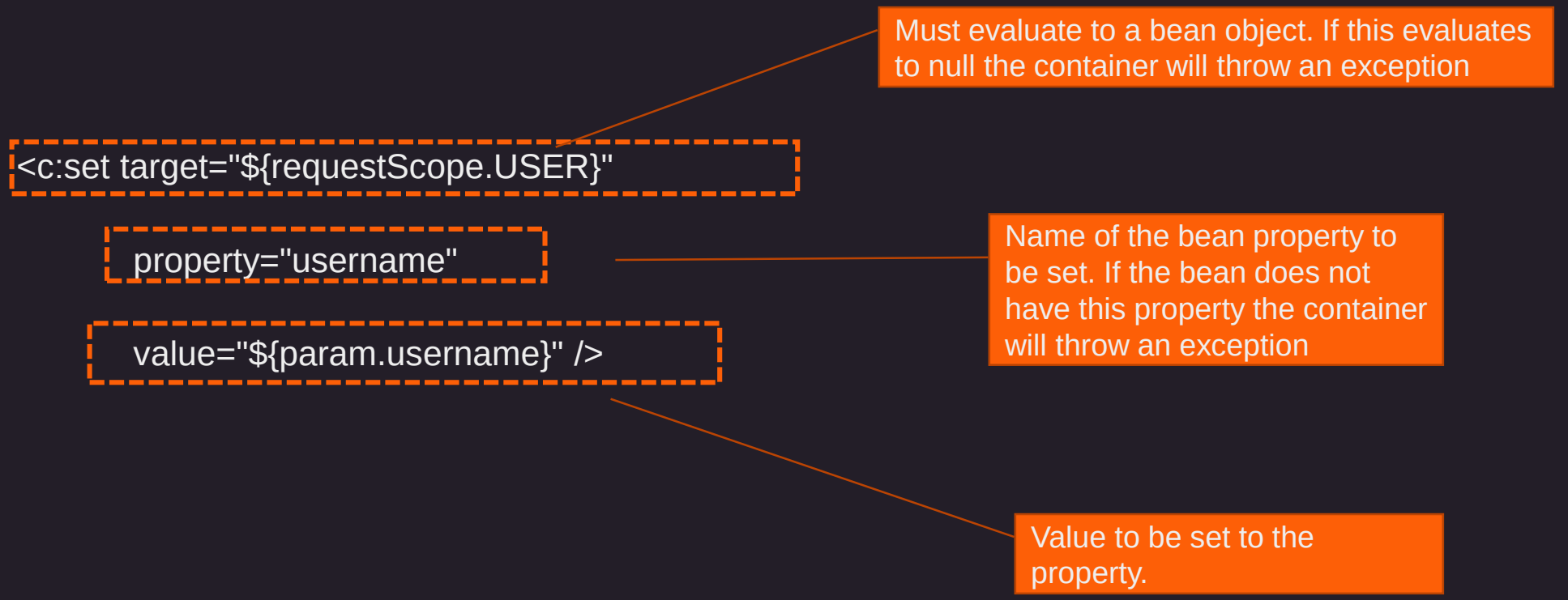



Confidential © 2020 Persistent Systems

# Core tags – variable support - `<c:set>` - Part III

- Set the value of a bean property

  **In the JSP**




Confidential © 2020 Persistent Systems

# Core tags – variable support - <c:remove>

- Remove an attribute

**In the JSP**

```
<c:remove var="USER" scope="request"/>
```

Name of the attribute to be removed. Must be a string literal. EL expression is not allowed

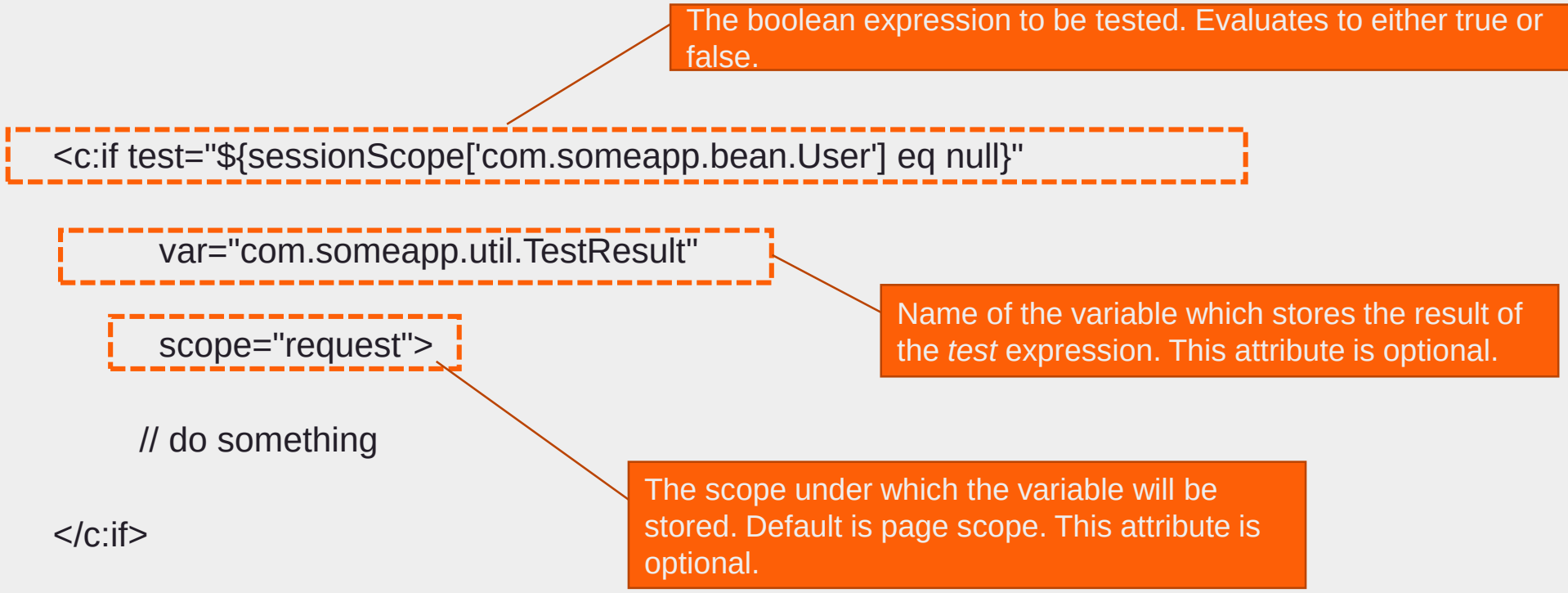
Scope from where the attribute is to be removed. If not specified the attribute is searched for and removed from all scopes.

Confidential © 2020 Persistent Systems

# Core tags – conditional - `<c:if>`

- Allows conditional execution of parts of a page based on a *test* (conditional) expression.

**In the JSP**




Confidential © 2020 Persistent Systems

Persistent

# Core tags – conditional - `<c:choose>`

- An if…..else if…..else version.

**In the JSP**

```
<c:choose>
    <c:when
        test="${sessionScope['com.someapp.bean.User'].memberType == 'Trial'}">
            // do something
    </c:when>
    <c:when
        test="${sessionScope['com.someapp.bean.User'].memberType == 'Premium'}">
            // do something
    </c:when>
    <c:otherwise>
            // do something else
    </c:otherwise>
</c:choose>
```

`c:when` with a boolean expression to be tested.

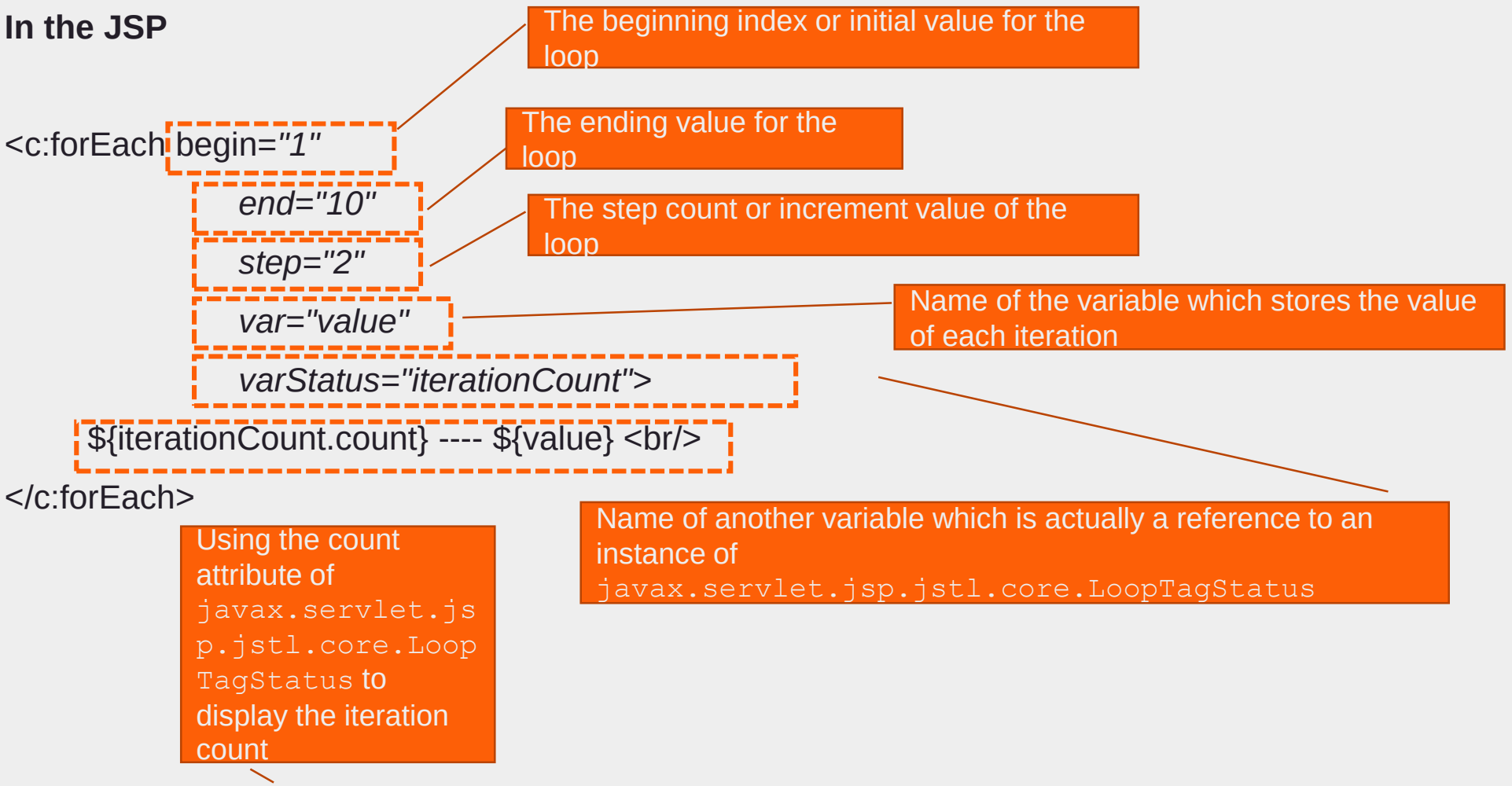
`c:otherwise` is executed when everything else fails (evaluates to false). This is optional.

Confidential © 2020 Persistent Systems

# Core tags – iteration - `<c:forEach>` - Part I

- Tag for basic looping and iterate over collections including maps.

**In the JSP**




Confidential © 2020 Persistent Systems

Persistent

# Core tags – iteration - <c:forEach> - Part II

- Tag for basic looping and iterating over collections including maps.

**In the servlet**

```
ArrayList<String> movies = new ArrayList<String>();
movies.add("Life is beautiful");
movies.add("Children of heaven");
movies.add("The pursuit of happyness");
movies.add("Babel");
request.setAttribute("com.app.util.Movies", movies);
```

**In the JSP**

```
<c:forEach  items="${requestScope['com.app.util.Movies']}"
            var="movie">
    ${movie} <br/>
</c:forEach>
```

Collection to be iterated over.

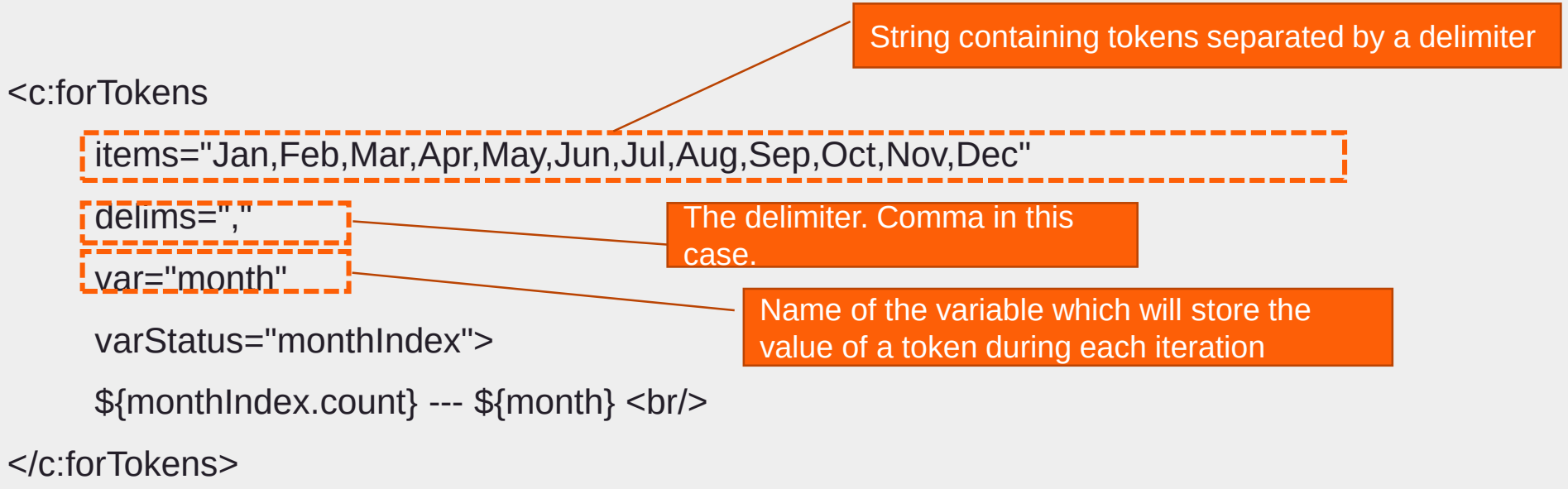
Name of the variable which stores one value from the collection  during each iteration

Confidential © 2020 Persistent Systems

# Core tags – iteration - `<c:forTokens>`

- Tag for iterating over tokens separated by a delimiter.

**In the JSP**




Confidential © 2020 Persistent Systems

Persistent

# Core tags – URL management - `<c:import>`

- Imports the contents (response) of the specified resource through the given url. This is then either included as part of the current JSP's response or stored in a variable.

**In the JSP**

**Capture the response of another JSP page and store it in a variable in the request scope**

```
<c:import   url="incl.jsp"
                    var="capturedContent"
                    scope="request" />
```

**Capture the response of a external resource**

```
<c:import url="http://www.google.co.in/index.html" />
```

**Capture the response of a resource located in another web application hosted on the same server**

```
<c:import url="/some-servlet"
            context="/some-web-app"/>
```

Confidential © 2020 Persistent Systems

# Core tags – URL management - `<c:url>`

- Rewrite or encode url's returned from a JSP page.

  **In the JSP**

  <a href="<c:url value="some-page.jsp"/>">Some page hyper link</a>

  **O/p (as seen in the page source)**

  With cookies enabled

  http://some-host/some-web-app/some-page.jsp

  With cookies disabled

  http://some-host/some-web-app/some-page.jsp;jsessionid=24EEE28B40E840E153766C1F4F6EACC4

Confidential © 2020 Persistent Systems

# Core tags – URL management - `<c:redirect>`

- Issues a http redirect.

  **In the JSP**

  <c:redirect url="first.jsp"/>

Confidential © 2020 Persistent Systems

# Core tags – miscellaneous - `<c:catch>`

- Encloses a block of JSP which might throw an exception.
  - Assuming the *User* bean in request scope does not have a *memberType* property, the above snippet throws an exception which is caught and stored in the variable named *exception*. This variable always has page scope.

**In the JSP**

```
<c:catch var="exception">
        ${requestScope.USER.memberType}
</c:catch>

${exception.message}
```

Confidential © 2020 Persistent Systems

# Core tags – miscellaneous - <c:out>

- Evaluates an expression and writes the result (output) to the current JSP's response.

**In the JSP**

```
<c:out value="Some string"/>
```

Displays ***Some string***

```
<c:out value="${1 - 2}"/>
```

Displays ***-1***

```
<c:out value="${param.username}"
       default="<unknown user>"
       escapeXml="true"/>
```

Displays the value of a request parameter named ***username***.
If the parameter is not found displays ***<unknown user>***.
The ***escapeXml*** attribute ensures that html/xml special characters like <, >, &, ' and " are escaped i.e. converted into their proper equivalents <, > and so on.

Confidential © 2020 Persistent Systems

# Core tags – miscellaneous - `<c:param>`

- Pass request parameters to a resource while importing or redirecting

```
<c:import url="some-page.jsp">
        <c:param name="SOME_REQ_PARAM" value="Some value"/>
</c:import>
```

  The passed request parameters can be accessed on the "imported" page through the implicit *request* object or *param* variable.

  When used with a <c:redirect> the request parameter is appended to the url as a query string.

- Encode a url with request (get) parameters while rewriting a url.

```
<c:url value="some-page.jsp" var="encodedURL">
        <c:param name="REQ_PARAM_ONE" value="First value"/>
        <c:param name="REQ_PARAM_TWO" value="Second value"/>
</c:url>

<a href="${encodedURL}">Hyperlink to page</a>
```

Confidential © 2020 Persistent Systems

# Internationalization – date formatting

- Format a date

**In a servlet**

```
Date date = new Date();
request.setAttribute(“someDate",date);
```

**In the JSP**

```
<fmt:formatDate value="${requestScope.someDate}“
    type="both"
    dateStyle="full“
    timeStyle="long"
    timeZone="US/Eastern" />
```

Must evaluate to an object of type `java.util.Date`

Indicates which fields are to be displayed. Must be any one of either *time*, *date* or *both*

Indicate how the date and time should be formatted. Valid values are *default*, *full*, *long*, *medium* or *short*.

Indicate the time zone in which date and/or time should be displayed .

**Output**

Monday, June 11, 2012 7:17:27 AM EDT

Confidential © 2020 Persistent Systems

# Internationalization – number formatting

- Format a number

**In the JSP**

```
<fmt:formatNumber value="1234567890“
                  type="currency“
                  maxFractionDigits="4“
                  minFractionDigits="2“
                  currencyCode="INR" />
```

Can be anything that can be interpreted as a number.

Any one of either *number*, *currency* or *percentage*. Indicates the type of numeric value being formatted.

Indicate the maximum and minimum digits to be displayed after the decimal point.

Specify currency for the numerical value being displayed. This code is used to determine the currency symbol to display as part of the formatted value.

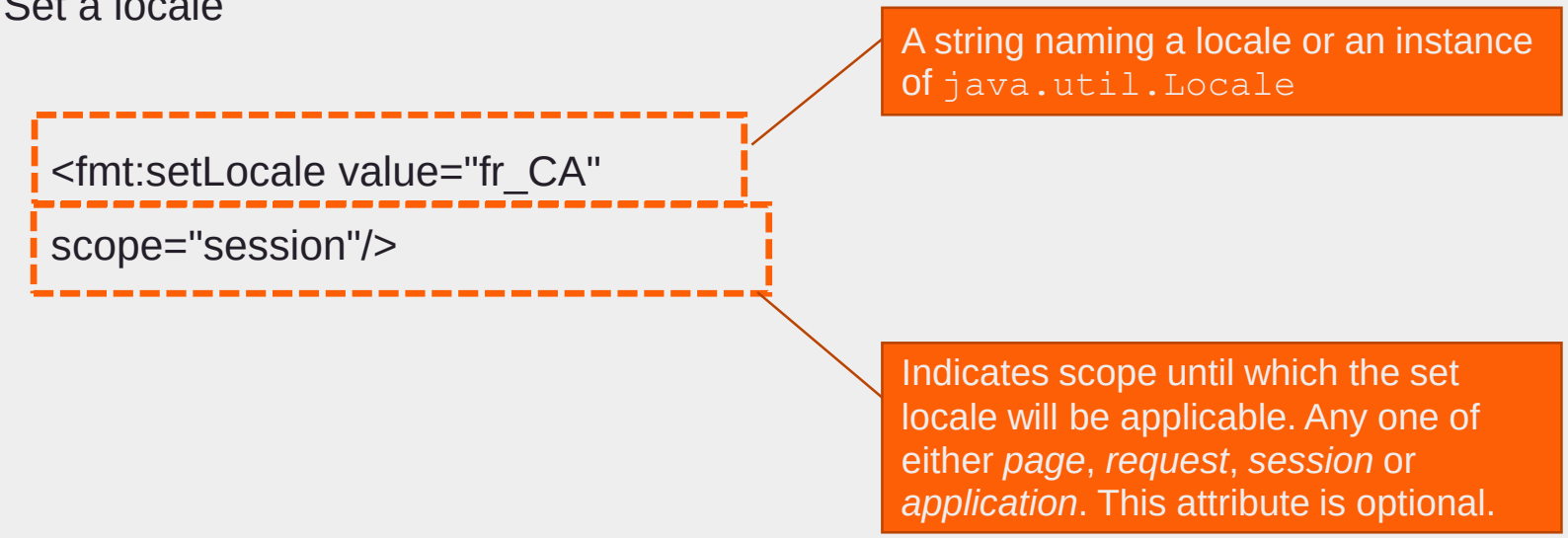
**Output**

INR 1,234,567,890.00

Confidential © 2020 Persistent Systems

# Internationalization – setting locale - `<fmt:setLocale>`

- Set a locale




Confidential © 2020 Persistent Systems

Persistent

# Internationalization - messaging

- Content localization with locale specific resource bundles

  **Step 1**

  Create a file named Messages_en_US.properties with the following content

  com.webapp.messages.greeting=Welcome

  Create another file named Messages_fr_FR.properties with the following content

  com.webapp.messages.greeting=Bon jour

  Both the files should be in your classpath. In this case under the com.webapp package

  **Step 2: In the JSP**

  <fmt:setBundle basename="com.webapp.Messages" />

  <fmt:message key="com.webapp.messages.greeting"/>

  Test the code by changing the locale with the <fmt:setLocale> tag covered earlier

Name of the resource bundle without any locale specific suffixes or filename extensions.

Display localized messages with keys as given in the resource bundle

Confidential © 2020 Persistent Systems

# SQL – setting the data source

- Create a new data source (database connection)

```
<sql:setDataSource                                    url="jdbc:mysql://some-host/test-db"
    driver="com.mysql.jdbc.Driver"          user=“some-username“
    password=“some-password"/>
```

If the *var* attribute is not present, then this action has the effect of setting the default datasource for use by *sql* tags that don't specify an explicit datasource.

Confidential © 2020 Persistent Systems

# SQL – queries

- Execute a sql query

```
<sql:query var="resultset"
                    sql="select * from sometable" />
<c:forEach items="${resultset.rows}"
        var="row">
    ${row.fieldname1} -- ${row.fieldname2}
    <br/>
</c:forEach>
```

Confidential © 2020 Persistent Systems

# SQL – updates

- Execute a sql insert or update

```
<sql:update
        sql="update sometable set fieldname1 = ?
                        where      fieldname2 = ?">
        <sql:param value=“somevalue“/>
        <sql:param value="4“/>
</sql:update>

<sql:update
        sql="insert into
sometable(fieldname1,fieldname2)
        values(?,?)">
        <sql:param value=“some-value-1“/>
        <sql:param value=“some-value-2“/>
</sql:update>
```

Confidential © 2020 Persistent Systems

# Functions - I

- JSTL has the following utility methods which can be used in an EL.
  - fn:contains
    - Tests if a string contains the given substring
    - **Example:** ${fn:contains("Hello","el")}
  - fn:containsIgnoreCase
    - Case in-sensitive version of fn:contains
  - fn:startsWith
    - Tests if a string starts with the given substring
    - **Example:** ${fn:startsWith("Hello","Hell")}
  - fn:endsWith
    - Tests whether a string ends with the given substring

Confidential © 2020 Persistent Systems

# Functions - II

- fn:indexOf
  - Returns the index within a string of the first occurrence of a given substring
  - Example: ${fn:indexOf("Hello","ll")}
- fn:trim
  - Removes leading and trailing white spaces from a string
- fn:length
  - Returns the size of a collection or length of a string
  - Example: ${fn:length(fn:trim(“ Muay Thai "))}
- fn:toUpperCase
  - Converts a string into upper case
- fn:toLowerCase
  - Converts a string into lower case

Confidential © 2020 Persistent Systems

# JSP standard actions

- A set of standard tags available with every web container.
- Do not require inclusion of a any external libraries
- Do not need a *taglib* directive
- By default use a prefix of *jsp*
- Tags for working with Java beans, request dispatching, embedding applets and other basic operations.

Confidential © 2020 Persistent Systems

# JSP actions - <jsp:useBean> - Part I

- Make an existing bean available to a JSP or instantiate a bean and populate its properties.

**In the servlet**

```
User user = new User();
user.setUsername("Some username");
request.setAttribute(“someBeanName", user);
```

**In the JSP**

```
<jsp:useBean id=“someBeanName"
             class="com.webapp.bean.User“
             scope="request"/ >
```

Name which will be used to lookup the bean in the specified scope. Also the name with which the bean can be accessed on the JSP page.

Package qualified name of the bean’s class

Scope in which the bean is to be looked for. Default is *page* scope.

The above snippet attempts to look for a bean named “someBeanName” in the request scope. If found, it is made available to the JSP with the same name. If not found a new object will be created, saved in the specified scope and made available to the JSP under the name “someBeanName”.

Confidential © 2020 Persistent Systems

# JSP actions - <jsp:useBean> - Part II

- Locate or instantiate a bean and assign it to a variable of type specified in the type attribute.

- The value of type can be the same as class, a superclass of class, or an interface implemented by class.

```
<jsp:useBean id=“someBeanName"

type="java.lang.Object“

        class="com.webapp.bean.User“
                                scope="request" />
```

Confidential © 2020 Persistent Systems

Persistent

# JSP actions - <jsp:useBean> - Part III

- Locate or instantiate a bean and assign it to a variable of type specified in the type attribute.

```
<jsp:useBean id=“someBeanName"
                                type="java.lang.Object"

beanName="com.webapp.bean.User"
                                scope="request" />
```

Confidential © 2020 Persistent Systems

Persistent

# JSP actions - <jsp:setProperty> - Part I

- Used along with the <jsp:useBean> to set the value of a bean's property

  - Set the value of all properties in a bean with values of matching request parameters.

```
<jsp:setProperty      name="someBeanName" property="*"/>
```

Must match value of the `id` attribute in the `<jsp:useBean>` tag

Set the value of a specific bean property with the value of a matching request parameter

```
<jsp:setProperty      name="someBeanName"property="username"/>
```

Confidential © 2020 Persistent Systems

# JSP actions - <jsp:setProperty> - Part II

Set the value of a specific bean property with value of the named request parameter

```
<jsp:setProperty name=“someBeanName“ property="username“ param="usrnm"/>
```

Set the value of a bean property either statically or through an expression

```
<jsp:setProperty name=“someBeanName"
property="username"
                                    value=“some user name"/>
```

Confidential © 2020 Persistent Systems

# JSP actions - <jsp:getProperty>

- Used along with the <jsp:useBean> to retrieve the value of a bean's property and display it on the page

```
<jsp:getProperty
name="someBeanName"
property="username" />
```

Must match value of the `id` attribute in the `<jsp:useBean>` tag

Confidential © 2020 Persistent Systems

# JSP actions - <jsp:include>

- Allow inclusion of a static (html) or dynamic (jsp, servlet) resource in a JSP.


  - If the url points to a static resource, its content is included as is in the JSP.
  - If the url points to a dynamic resource, it acts on the request and sends back a result (response) that is included in the JSP.
  - When the include action is complete, the container continues processing remainder of the JSP file.

Confidential © 2020 Persistent Systems

# JSP actions - <jsp:forward>

- Forwards the current request to another resource (jsp or servlet)

```
<jsp:forward page="/some-servlet"/>
```

URL of either a JSP or servlet

Confidential © 2020 Persistent Systems

# JSP actions - <jsp:param>

- Pass request parameters to a resource while *includ*ing or *forward*ing

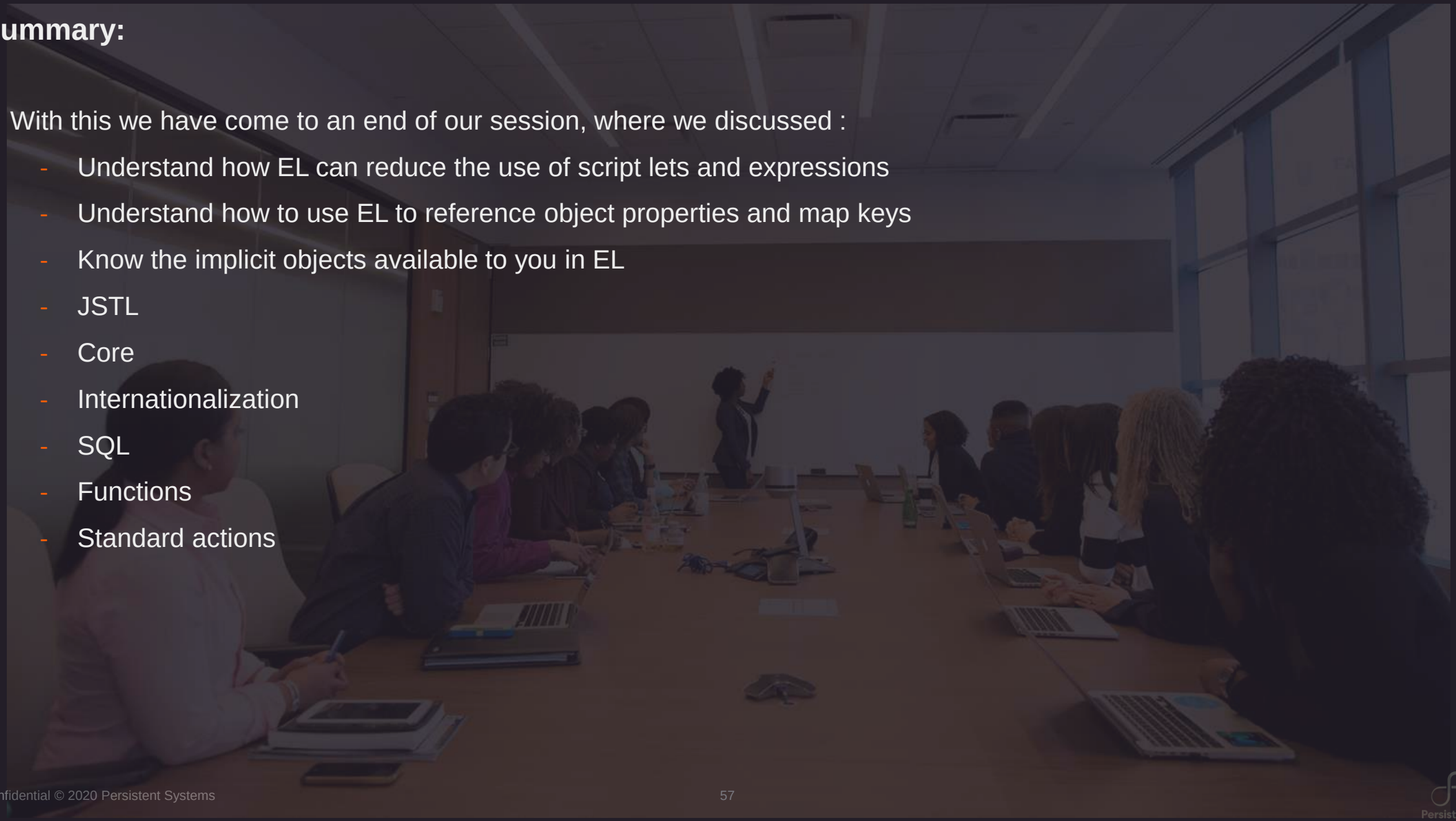
- Parameter values can be retrieved in the targeted resource through the *request* object or *param* implicit object (in EL)

```
<jsp:include page=“some-page.jsp">
        <jsp:param  value=“some-param-name"
                                name=“some value"/>
</jsp:include>

<jsp:forward page="/some-servlet">
        <jsp:param  value=“some-param-name"
                                name=“some value"/>
</jsp:forward>
```

Confidential © 2020 Persistent Systems

## Summary:



- With this we have come to an end of our session, where we discussed :
  - Understand how EL can reduce the use of script lets and expressions
  - Understand how to use EL to reference object properties and map keys
  - Know the implicit objects available to you in EL
  - JSTL
  - Core
  - Internationalization
  - SQL
  - Functions
  - Standard actions

Confidential © 2020 Persistent Systems

Persistent

# Appendix

Thank You

Persistent

# Thank you