có thể dùng EL hoặc scriptlet để xuất giá trị thuộc tính của 1 đối tượng trong jsp

**What and Why use Expression Language (EL)?**

**Expression Language (EL)** is a feature used in JavaServer Pages (JSP) and JavaServer Faces (JSF) that allows developers to access and manipulate application data easily without needing to write complex Java code. EL simplifies the integration of dynamic content in web applications by allowing the use of simple expressions to access and bind data directly in JSP/JSF pages.

**Reasons to Use EL:**

1. **Simplified Syntax**: EL provides a straightforward syntax to access and manipulate data, making it easier for developers to work with.
2. **Decoupling Logic from Presentation**: By using EL, you keep the presentation logic (in JSP/JSF pages) separate from the business logic (in Java classes), promoting a cleaner and more maintainable codebase.
3. **Enhanced Readability**: EL expressions are easier to read and understand compared to equivalent scriptlet code in JSP.
4. **Scope Management**: EL supports accessing objects stored in various scopes (page, request, session, and application), facilitating the management of web application data.

**How to Write with the EL Syntax?**

EL expressions are written using the ${} syntax. Within these braces, you can write expressions to access data, perform operations, and invoke methods.

**Basic Syntax Examples:**

* ${user.name}: Accesses the name property of the user object.
* ${user.age > 18 ? 'Adult' : 'Minor'}: Uses a conditional (ternary) operator to check if user.age is greater than 18.

**Immediate Evaluation**

Immediate evaluation in EL refers to evaluating expressions as soon as they are encountered in the JSP or JSF page. EL expressions are evaluated at runtime, providing real-time access to data and variables.

**How to Use Scoped Variables in EL Expressions?**

EL allows accessing variables stored in different scopes using predefined implicit objects:

1. **pageContext**: Accesses attributes at the page scope.
2. **requestScope**: Accesses attributes at the request scope.
3. **sessionScope**: Accesses attributes at the session scope.
4. **applicationScope**: Accesses attributes at the application scope.

**Examples:**

* ${pageContext['attributeName']}: Accesses an attribute stored in the page scope.
* ${requestScope['attributeName']}: Accesses an attribute stored in the request scope.
* ${sessionScope['attributeName']}: Accesses an attribute stored in the session scope.
* ${applicationScope['attributeName']}: Accesses an attribute stored in the application scope.

**Implicit Variables in EL**

EL provides several implicit variables that are available by default:

* pageContext: The context for the entire JSP page.
* param: A map of request parameters.
* paramValues: A map of request parameter arrays.
* header: A map of request headers.
* headerValues: A map of request header arrays.
* cookie: A map of cookie values.
* initParam: A map of context initialization parameters.

**Examples:**

* ${param['username']}: Accesses the value of the request parameter username.
* ${cookie['sessionId'].value}: Accesses the value of the cookie sessionId.

**Conditional Evaluation**

EL supports conditional expressions using the ternary operator (? :). This allows for inline conditional logic within EL expressions.

**Example:**

* ${user.age > 18 ? 'Adult' : 'Minor'}: Checks if user.age is greater than 18 and returns 'Adult' if true, otherwise returns 'Minor'.

**Putting it All Together**

Here is an example JSP snippet demonstrating the use of EL with various scopes, implicit variables, and conditional evaluation:

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Expression Language Example</title>

</head>

<body>

<!-- Accessing a session-scoped variable -->

<p>Welcome, ${sessionScope['user'].name}!</p>

<!-- Accessing a request parameter -->

<p>Your username is: ${param['username']}</p>

<!-- Using a conditional expression -->

<p>Age status: ${user.age > 18 ? 'Adult' : 'Minor'}</p>

<!-- Accessing a cookie value -->

<p>Session ID: ${cookie['sessionId'].value}</p>

<!-- Accessing an application-scoped variable -->

<p>Application name: ${applicationScope['appName']}</p>

</body>

</html>

**JavaBean** là các class Java thuần, dùng để thể hiện các thực thể trong các chương trình Java.

Một lớp JavaBean sẽ có 3 đặc điểm như sau về mặt cú pháp:

1. Là một public class  
2. Có các thuộc tính private  
3. Ứng với mỗi thuộc tính, lớp sẽ cung cấp một cặp phương thức setter / getter để truy cập và thao tác với giá trị của từng thuộc tính.



