The Expression Language

Kevin Jones
@kevinrjones

Why an Expression Language?

Why an Expression Language?

- Previous versions on JSP relied on Java as the scripting language

Why an Expression Language?

- **Previous versions on JSP relied on Java as the scripting language**
  - Made it difficult to produce dynamic pages

Why an Expression Language?

- **Previous versions on JSP relied on Java as the scripting language**
  - Made it difficult to produce dynamic pages
  - JSP 2.0 introduced an EL

Why an Expression Language?

- **Previous versions on JSP relied on Java as the scripting language**
  - Made it difficult to produce dynamic pages
  - JSP 2.0 introduced an EL
  - EL is very 'page-author' friendly

Why an Expression Language?

- **Previous versions on JSP relied on Java as the scripting language**
  - Made it difficult to produce dynamic pages
  - JSP 2.0 introduced an EL
  - EL is very 'page-author' friendly
  - Allows page authors a limited form of page scripting

Expression Language

- **Based on**
  - Nested properties
  - Access to collection classes
  - Operators (Relational; logical; arithmetic)
  - Extensibility functions (mapped to static Java methods)
  - A set of implicit objects

Using the Expression Language

- **Syntax**
  - Always use **`${expr}`** construct
- **EL can be used on page**
  - This will simply output the value of the expression

```
<span>Name: ${user.name}</span>
```

Expression Language in Attributes

- EL can be used in attributes

```
<a:name value="${somename}"/>
<a:name value="${last}${first}">
```

Expression Language Operators

- **Expressions can use operators**
  - Mathematical
    - + - * / div % mod == eq != ne
    - < lt > gt <= le >= ge
  - logical
    - && and ! not || or
  - empty operator
    - empty

Accessing JavaBeans

- **Can get bean property's**
  - use either . (dot) or [] syntax
    - user.name
    - user["name"]
  - can use [] for access to lists, maps, arrays and beans
  - Can nest arbitrarily
    - a["name"].first
    - a["name"]["first"]
    - etc…

Implicit Objects

- **EL has access to a set of implicit objects**
  - All Maps (except pageContext)

pageContext	pageScope
requestScope	sessionScope
applicationScope	param
paramValues	header
headerValues	cookie
initParam	

```
Host header is ${header.Host}
Host header is ${header["Host"]}
Error status is ${pageContext.errorData.status}
```

Summary

- **The Expression Language**
  - Not a "full" language
  - Gives access to properties on JavaBeans
  - Can use the '.' or '[]' syntax
  - Has access to built in intrinsics
  - Has a set of operators
  - Can access collections