Dt : 27/12/2023

Expression Language(EL):

=>Expression Language will provide more flexibility to access the objects of JSP like request,response,out,...

=>Expression Language is a newly added feature to JSP-Programming and which is known as "JSP-EL".

syntax:

$(Expression)

=>The following are implicit objects of Expression Language(EL):

1. ***applicationScope***
2. ***sessionScope***
3. ***requestScope***
4. ***pageScope***
5. ***param***
6. ***paramValues***
7. ***header***
8. ***headerValues***
9. ***cookie***
10. ***initParam***
11. ***pageContext***

1.applicationScope:

=>'appliactionScope' is used to access attribute from ServletContext Object.

2.sessionScope:

=>'sessionScope' is used to access attribute from HttpSession Object.

3.requestScope:

=>'requestScope' is used to access attribute from jsp-request-Object.

4.pageScope:

=>'pageScope' is used to access attribute from jsp-pageContext-Object.

5.param:

=>'param' is used to access single para-value from jsp-request-Object.

6.paramValues:

=>'paramValue' is used to multiple para-values from jsp-request-Object in the form of String-Array

7.header:

=>'header' is used to access single header value from jsp-request-Object.

8.headerValues:

=>'headerValues' is used to access multiple header values from jsp-request-Object

in the form of String-Array

9.cookie:

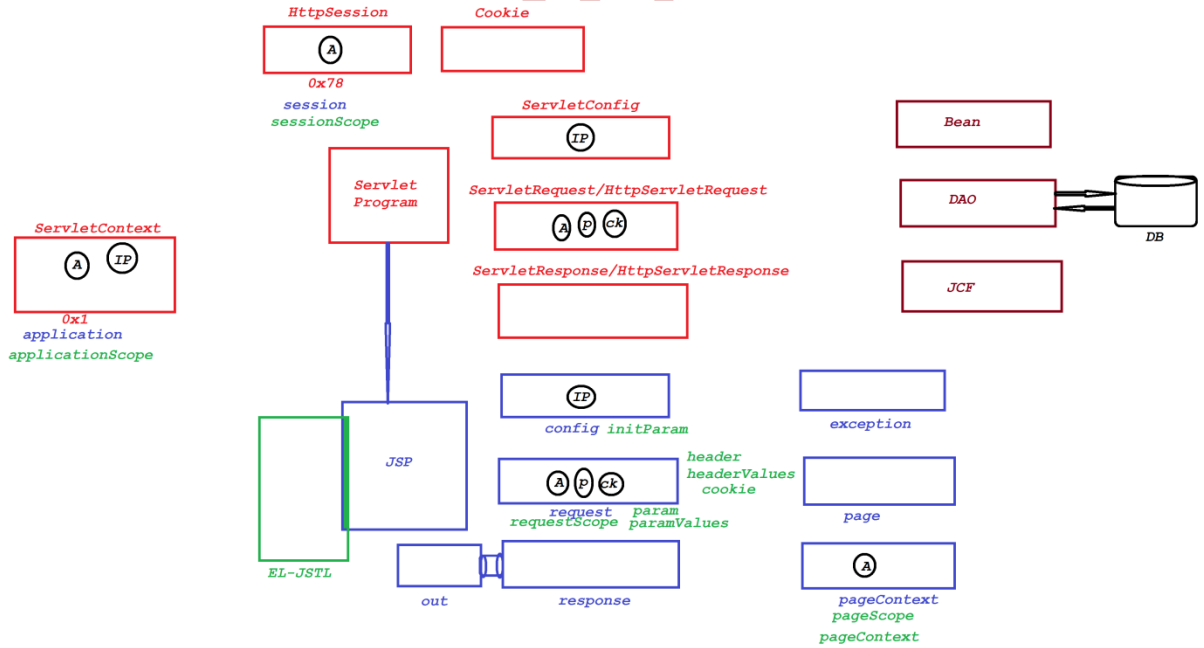
=>'cookie' is used to access cookie from jsp-request-Object

10.initParam:

=>'initParam' is used to access initialization para-value from jsp-config-Object

11.pageContext:

=>'pageContext' is an implicit object refering jsp-pageContext-Object



==========

**imp*

Complete Summary of Objects Generated from CoreJava and AdvJava:

CoreJava Objects:(Container Objects:

1.User defined Class Objects

2.String-Objects

(a)String-Class-Object

(b)StringBuffer-Class-Object

(c)StringBuilder-Class-Object

3.WrapperClass Objects

(a)Byte Object

(b)Short Object

(c)Integer Object

(d)Long Object

(e)Float Object

(f)Double Object

(g)Character Object

(h)Boolean Object

4.Array Objects

(a)Array holding User defined Class Objects

(b)Array holding WrapperClass Objects

(c)Array holding String Objects

(d)Array holding Array-Objects(Jagged Array)

(e)Array holding Dis-Similer Objects(Object Array

5.Collection<E> Objects

(a)Set<E> Objects

(i)HashSet<E> Object

(ii)LinkedHashSet<E> Object

(iii)TreeSet<E> Object

(b)List<E> Objects

(i)ArrayList<E> Object

(ii)LinkedList<E> Object

(iii)Vector<E> Object

=>Stack<E> Object

(c)Queue<E> Objects

(i)PriorityQueue<E> Object

=>Deque<E> Objects

(ii)ArrayDeque<E> Object

(iii)LinkedList<E> Object

6.Map<K,V> Objects

(a)HashMap<K,V> Object

(b)LinkedHashMap<K,V> Object

(c)TreeMap<K,V> Object

(d)Hashtable<K,V> Object

7.Enum<E> Objects

Total : 31 Objects

JDBC Objects:

1.Connection Object

2.Statement Object

3.PreparedStatement Object

4.CallableStatement Object

5.ResultSet Object

(i)Scrollable ResultSet Object

(ii)NonScrollable ResultSet Objects

6.RowSet Objects

(a)JdbcRowSet Object

(b)CachedRowSet Objects

=>WebRowSet Objects

(i)FilterRowSet Object

(ii)JoinRowSet Object

7.DatabaseMetaData Object

8.ParameterMetaData Object

9.ResultSetMetaData Object

10.RowSetMetaData Object

---------- Total : 10 Objects

Servlet Objects:

1.ServletContext Object

2.ServletConfig Object

3.ServletRequest/HttpServletRequest Object

4.ServletResponse/HttpServletResponse Object

5.PrintWriter Object

6.HttpSession Object

7.Cookie Object

8.Bean Objects

9.DAO Objects

10.JCF Objects

---------- Total : 10 Objects

JSP Implicit Objects:

1.application

2.config

3.request

4.response

5.out

6.session

7.exception

8.page

9.pageContext

Total : 9 Objects

EL Implicit Objects:

1.applicationScope

2.sessionScope

3.requestScope

4.pageScope

5.param

6.paramValues

7.header

8.headerValues

9.cookie

10.initParam

11.pageContext

Total : 11 Objects

****imp***

JSTL(JSP Standard Tag Lib):

=>This JSTL represents a set of tags to simplify the JSP development.

Advantages of JSTL:

(i)Fast Development

(ii)Code Reusability

(iii)No need to use scriptlet tag

The following are the JSTL tags:

(1)Core Tags

(2)Function Tags

(3)Formatting Tags

(4)XML Tags

(5)SQL Tags

Note:

=>To work with EL-JSTL,download the following Jar-file:

External JAR : jstl-1.2.jar(Download from Internet)

=>This JSTL-Jar must be copied into "lib" folder of WEB-INF

=>we use "@taglib" directive tag to declare JSTL Tags.

---

(1)Core Tags:

=>These JSTL core tags provides variable support,URL management, flow control etc. (Basic code writing)

syntax:

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

The following are the List of JSTL Core Tags:

c:out ->It displays the result of an expression, similar to <%=...%> tag.

c:import->It Retrives relative or an absolute URL.

c:set-> It sets the value to variable.

c:remove->It is used for removing the variable .

c:catch-> It is used for Catching any Throwable exception that occurs in the body.

c:if-> It is an conditional tag

c:choose, c:when, c:otherwise->

It is the simple conditional tag that includes its body content if the evaluated condition is true.

**imp*

c:forEach-> It is the basic iteration tag.

c:forTokens-> It iterates over tokens which is separated by the supplied delimeters.

c:param-> It adds a parameter in a containing 'import' tag's URL.

c:redirect-> It redirects the browser to a new URL and supports the

context-relative URLs.

c:url-> It creates a URL with optional query parameters.

==========

(2)Function Tags:

=>These JSTL function tags provides a number of standard functions, most of these functions are common string manipulation functions.

syntax:

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

List of Some Function tags:

fn:contains :It is used to test if an input string containing the specified substring or not.

fn:containsIgnoreCase(): It is used to test if an input string contains the specified substring as a case insensitive way.

fn:endsWith() : It is used to test if an input string ends with the

specified suffix.

fn:indexOf(): It returns an index within a string of first occurrence of a specified substring.

fn:trim(): It removes the blank spaces from both the ends of a string.

fn:startsWith(): It is used for checking whether the given string is started with a particular string value or not.

fn:split(): It splits the string into an array of substrings.

fn:toLowerCase(): It converts all the characters of a string to lower case.

fn:toUpperCase(): It converts all the characters of a string to uppercase.

fn:substring(): It returns the subset of a string according to the given start and end position.

fn:length(): It returns the number of characters inside a string, or

the number of items in a collection.

fn:replace(): It replaces all the occurrence of a string with another string sequence.

==========

(3)Formatting Tags:

=>These formatting tags provide support for message formatting,number formating and date formatting etc.

=>These formatting tags are also used for internationalized web sites to display and format text, time,date and numbers.

syntax:

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

List of some Formatting tags:

fmt:parseNumber : It is used to Parse the string representation of a currency, percentage or number.

fmt:formatNumber : It is used to format the numerical value with specific format or precision.

fmt:parseDate : It parses the string representation of a time and date.

fmt:setTimeZone : It stores the time zone inside a time zone configuration variable.

**imp*

fmt:formatDate : It formats the time and date using the supplied pattern and styles.

==========

(4)XML Tags:

=>The JSTL XML tags are used for providing a JSP-centric way of manipulating and creating XML documents.

syntax:

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

(5)SQL Tags:

=>The SQL tag library allows the tags to Interact with RDBMS

(Relational Databases) such as Microsoft SQL Server, mySQL,

or Oracle.

syntax:

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

Note:

=>In realtime we must not have JSP Centric XML and DB

Connections,because of this reason XML tags and SQL Tags are

less used when compared to other tags.

==========