

SpEL

Summary

SpEL is the expression language for spring first introduced in Spring 3.0 which is used powerfully and flexibly. It brings the property value actively using the expression language that can be approached directly to bean object. Moreover, SpEL can be applied using the <spring:eval> tag in jsp.

Description

Application of SpEL using Bean Configuration File

When a value is configured on bean property, it is possible to access other beans or properties.

- Example to access from the **following bean**.

```
<bean id="springTest" ..>
    <property name="test" value="Sample" />
</bean>
```

```
<bean id="testNames">
    <property name="name" value="#{springTest.test}" />
</bean>
```

- Following is the example of accessing to **property using <util:properties>**.

globals.properties

```
driverClassName=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:1623/EASYCOMPANY
username=tes
password=tesAdmin
```

context-datasource.xml

```
<util:properties id="dbprops" location="classpath:/egovframework/property/globals.properties" />
    <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-
method="close">
        <property name="driverClassName" value="#{dbprops['driverClassName']}" />
        <property name="url" value="#{dbprops['url']}" />
        <property name="username" value="#{dbprops['username']}" />
        <property name="password" value="#{dbprops['password']}" />
    </bean>
```

- Following is the example of **inserting to variable directly using <util:properties>**.

```
@Value("#{dbprops.driverClassName}")
private String driverClassName;
```

or

```
@Value("#{dbprops}")
private Dbproperties dbprops;
```

SpEL Application in JSP

Instead of EL in JSP, the value can be printed using SpEL of Spring 3.0. In order to use SpEL in JSP, the tag library should be added.

<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>

Use SpEL in JSP using <spring:eval> tag.

The model object can be used directly.

```
<spring:eval expression="sampleVO.money"/>
```

If the return value of method is string, method itself can be called.

```
<spring:eval expression="sampleVO.toString()"/>
```

In addition, pointer registered in convergence service as shown in @NumberFormat, @DateTimeFormat can be automatically applied.

Following is a part of sampleVO

```
/** Balance */
@NumberFormat(pattern = "###,##0")
private Integer money;
<spring:eval expression="sampleVO.money"/>
```

If it is applied as shown above, comma(,) is displayed every 3 digit according to input value.

Input value: 1234000 output value: 1,234,000

SpEL can be applied using new without setting directly to annotation in the model.

```
<spring:eval expression='new java.text.DecimalFormat("###,##0").format(price)'/>
```

SpEL 예제

이름:

통장 잔액: (숫자만 입력하세요)

SpEL Sample

SpEL 적용 데이터1 : Admin님의 통장 잔고는 1,246,000원 입니다.
toString 적용 데이터 : 이름은 Admin, 현재 잔액은 1246000원

SpEL 적용 데이터2 : SampleProduct의 가격은 1,234,500원 입니다.

Reference

- [Spring Framework - Reference Document / 7. Spring Expression Language \(SpEL\)](#)
- [SpEL Example](#)