Spring Expression Dili (SpEL)

Spring EL Nedir?

- SpEL, Spring'in **expression dili**dir
- **Unified EL**'e benzer fakat ondan daha gelişmiştir
- **Object graph query** işlemini ve **manipülasyonu**nu sağlar
- Method invocation, değer set etme, nesne yaratma vs yapılabilir

Spring EL ile Çalışma Adımları

SpelExpressionParser implementasyonu default impl.dır

1 *Parser oluştur*

```
ExpressionParser parser = new SpelExpressionParser();
```

Oluşturulan parser ile String ifade parse edilir ve Expression elde edilir

2 *Expression'ı parse et*

```
Expression exp = parser.parseExpression("'Hello World'");
```

3 *Expression'ı evaluate et*

```
String message = (String) exp.getValue();
```

getValue metodu expression'ı evaluate ederek değerini döndürür

Spring EL Expression Örnekleri

JAVA Eğitimleri

```
ExpressionParser parser = new SpelExpressionParser();
Expression exp = parser.parseExpression("'Hello World'.bytes");
byte[] bytes = (byte[]) exp.getValue();
```

```
//getBytes().length metodu cagrilir...
Expression exp = parser.parseExpression("'Hello
World'.bytes.length");
int length = (Integer) exp.getValue();
```

```
Expression exp = parser.parseExpression("new String('hello
world').toUpperCase()");
String message = exp.getValue(String.class);
```

FQN yazılarak herhangi
Bir sınıftan nesne yaratmak
Da mümkündür

Spring EL ve Evaluation Context

- Tanımlanan expression'ları nesneler üzerinde evaluate etmek ve bu nesnelerin property'lerine değer set etmek için kullanılır

```
ExpressionParser parser = new SpelExpressionParser();
Expression exp = parser.parseExpression("firstName");

Owner owner = new Owner();
owner.setFirstName("Ali");
owner.setLastName("Yücel");

StandardEvaluationContext context = new StandardEvaluationContext();
context.setRootObject(owner);

String firstName = (String) exp.getValue(context);
exp = parser.parseExpression("address?.city");
String firstName = (String) exp.getValue(context);
exp = parser.parseExpression("lastName");
String lastName = (String) exp.setValue(context,"Yılmaz");
```

Expression evaluation işleminin üzerinde gerçekleştirildiği nesneye root object denir

Spring EL Değişkenleri ile Değer Set Etme

JAVA Eğitimleri

```
ExpressionParser parser = new SpelExpressionParser();

Expression exp = parser.parseExpression("firstName = #newName");

Owner owner = new Owner();
owner.setFirstName("Ali");
owner.setLastName("Yücel");

StandardEvaluationContext context = new StandardEvaluationContext();
context.setRootObject(owner);

context.setVariable("newName", "Veli");
```

Context içerisinde newName İsimli bir değişken ve değeri Tanımlanır.

Daha sonra bu değişkenin değeri firstName Property'sine assign edilir

```
String firstName = (String) exp.getValue(context);
```

Spring EL'de Fonksiyon Tanımlama

```
ExpressionParser parser = new SpelExpressionParser();
StandardEvaluationContext context = new StandardEvaluationContext();

Method method = StringUtils.class.getDeclaredMethod("reverseString",
new Class[] { String.class })
context.registerFunction("reverseString", method);
```

Reflection ile elde edilen metot nesnesi register edilip expression İçerisinde kullanılabilir

```
String helloWorldReversed =
parser.parseExpression("#reverseString('hello'
world)").getValue(context, String.class);
```

SpEL ve Bean Tanımları

```
<bean id="numberGuess" class="x.y.NumberGuess">
    <property name="randomNumber" value="#{ T(java.lang.Math).random()* 100.0 }" />
</bean>
```

Bean tanımlarındaki property değerlerinde SpEL ifadeleri kullanılabilir
T() ifadesi ile burada bir java tipi FQN ile ifade edilmiştir

```
<bean id="taxCalculator" class="x.y.TaxCalculator">
    <property name="defaultLocale" value="#{ systemProperties['user.region'] }"/>
</bean>
```

JVM sistem değişkenlerine erişmek de mümkündür

```
<bean id="shapeGuess" class="org.spring.samples.ShapeGuess">
    <property name="initialShapeSeed" value="#{ numberGuess.randomNumber }"/>
</bean>
```

Expression içerisinde başka bir bean'ın property değerine
Erişmek de mümkündür

SpEL ve Annotasyon Tabanlı Konfigürasyon

```
@Component
public class FooBean {

    @Value("#{ systemProperties['user.region'] }")
    private String defaultLocale;

    public void setDefaultLocale(String defaultLocale) {
this.defaultLocale = defaultLocale;
    }

...
}
```

Metot ve field düzeyinde kullanılabilir

Metot input parametre tanımlarında da kullanılabilir

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com