

Spring Expression Language [SpEL]:

Expression Language is a programming language, it will provide simplified syntaxes to manipulate Objects and their properties.

EX:

1.JSP EL: To evaluate the objects and their properties like request, session, application,.... and their parameters and attributes.

2.Struts2.x OGNL: To evaluate the objects and their properties like Value Stack, CentralContext, request, application.....

3.JBOSS EL: To evaluate Objects and their properties which are related to the JBOSS implementations.

4.SpEL: To evaluate bean objects and their properties in Spring applications

SpEL: It is an Expression Language, it has provided simplified syntaxes to manipulate objects and their properties during Runtime of the applications

To prepare and Evaluate Expressions in SpEL, Spring has provided very good Predefined Library in the form of "org.springframework.expression" package.

In Spring applications, if we want to prepare and evaluate expressions we have to use the following steps.

1.Create ExpressionParser object:

ExpressionParser is able to manage expressions and it able to have expression evaluation mechanisms.

To represent Expression Parser Spring Framework has provided a predefined interface in the form of org.springframework.expression.ExpressionParser .

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

For ExpressionParser interface , Spring framework has provided a predefined implementation class in the form of
"org.springframework.expression.spel.standard.SpelExpressionParser" .

EX: ExpressionParser parser = new SpelExpressionParser();

Note: ExpressinParser is able to evaluate the expressions by using StandardEvaluationContext, it able to evaluate the expressions against objects by preparing Object Graphs internally.

2.Create Expression object:

In SpEL, Expression object is able to represent single Expression. To represent Expression , SpEL has provided a predefined interface in the form of "org.springframework.expression.Expression" . For Expression interface SpEL has provided a predefined implementation class in the form of "org.springframework.expression.spel.standard.SpelExpression" .

To prepare expression and to get Expression object we have to use the following method from ExpressionParser .

```
public Expression parseExpression(String expression)
```

EX: Expression expr = parser.parseExpression("10+10");

3.Get Result of the Expression Evaluation:

To get expression result we have to use the following method from Expresion.

```
public Object getValue()
```

EX: int val = (Object) expr.getValue();

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

EX:

```
package com.durgasoft.test;

import org.springframework.expression.Expression;
import org.springframework.expression.ExpressionParser;
import org.springframework.expression.spel.standard.SpelExpressionParser;

public class Test {

    public static void main(String[] args) throws Exception {
        ExpressionParser parser = new SpelExpressionParser();
        Expression expr = parser.parseExpression("10+10");
        int val1 = (Integer) expr.getValue();
        System.out.println(val1);

        expr = parser.parseExpression("10*10");
        int val2 = (Integer) expr.getValue();
        System.out.println(val2);

        expr = parser.parseExpression("'abc'+'def'");
        String val3 = (String) expr.getValue();
        System.out.println(val3);
    }
}
```

EXample with StandardEvaluationContext:**CalculatorBean.java**

```
package com.durgasoft.beans;

public class CalculatorBean {
    private int num1;
    private int num2;
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

setXXX() and getXXX()

```
public int add() {  
    return num1+num2;  
}  
public int sub() {  
    return num1-num2;  
}  
public int mul() {  
    return num1*num2;  
}  
}
```

Test.java

```
package com.durgasoft.test;  
  
import org.springframework.expression.Expression;  
import org.springframework.expression.ExpressionParser;  
import org.springframework.expression.spel.standard.SpelExpressionParser;  
import org.springframework.expression.spel.support.StandardEvaluationContext;  
  
import com.durgasoft.beans.CalculatorBean;  
  
public class Test {  
    public static void main(String[] args) {  
        CalculatorBean cal = new CalculatorBean();  
        StandardEvaluationContext context = new  
StandardEvaluationContext(cal);  
        ExpressionParser parser = new SpelExpressionParser();  
  
        Expression expr1 = parser.parseExpression("num1");  
        expr1.setValue(context, "10");  
    }  
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
Expression expr2 = parser.parseExpression("num2");
expr2.setValue(context, "5");

System.out.println("num1 :"+cal.getNum1());
System.out.println("num2 :"+cal.getNum2());
System.out.println("ADD :"+cal.add());
System.out.println("SUB :"+cal.sub());
System.out.println("MUL :"+cal.mul());
    }
}
```

SpEL Features:

- 1.Expressions
- 2.Operators
- 3.Variables
- 4.Method Invocations
- 5.Collections

1.Expressions

SpEL is able to allow two types of Expressions.

- 1.Literal Expressions
- 2.Regular Expressions

1.Literal Expressions

---> It able to allow only literals inside the Expressions.

2.Regular Expressions

---> It able to check the specified String against the provided Regular Expressions, If the provided String is as per the provided Regular Expression then it will return true value otherwise it will return false value.

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

To compare String with Regular Expression we have to use "matches" operator, it is a boolean operator, it will check whether the provided String is satisfying the provided Regular Expression or not.

Syntax:

'String_Data' matches 'Reg_Expression'

Example:

```
package com.durgasoft.test;
import org.springframework.expression.Expression;
import org.springframework.expression.ExpressionParser;
import org.springframework.expression.spel.standard.SpelExpressionParser;
public class Test {

    public static void main(String[] args)throws Exception {
        ExpressionParser parser = new SpelExpressionParser();
        Expression expr = parser.parseExpression("10 + 10");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("'abc' + 'def'");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("0xABCDEF*10+6");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("'Spring' matches 'Sp.*'");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("'abc@durgasoft.com' matches '[a-z]*@durgasoft.com'");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("'abc@gmail.com' matches '[a-z]*@durgasoft.com'");
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,


```
System.out.println(expr.getValue());
```

```
}
```

```
}
```

2.Operators:

In SpEL, to prepare Expressions we are able to use the following operators.

i).Arithmetic Operators:

+, -, *, /, %,.....

ii).Logical Operators:

and or &&

or or ||

not or !

iii).Comparison Operators:

eq or ==

ne or !=

lt or <

le or <=

gt or >

ge or >=

iv).Ternary Operator:

Expr1? Expr2: Expr3;

v).Type Operator[T]:

It able to represent a particular class type or interface type in SpEL.

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Syntax:

T(Class_Name)

vi).Safe Navigation Operator:

In general, in JAVA applications, if we access any instance variable or method on a reference variable contains null value then JVM will rise an exception like java.lang.NullPointerException. In SpEL , to avoid NullPointerExceptions we are able to use "Safe Navigation" operator.

Syntax:

var_Name?.methid_or_Var_Name()

Example:

```
package com.durgasoft.beans;

public class User {
    private String unname;

    public String getUnname() {
        return unname;
    }

    public void setUnname(String unname) {
        this.unname = unname;
    }
}
```

Test.java

```
package com.durgasoft.test;
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,


```
import org.springframework.expression.Expression;
import org.springframework.expression.ExpressionParser;
import org.springframework.expression.spel.standard.SpelExpressionParser;
import org.springframework.expression.spel.support.StandardEvaluationContext;

import com.durgasoft.beans.User;

public class Test {

    public static void main(String[] args) throws Exception {
        ExpressionParser parser = new SpelExpressionParser();
        Expression expr = parser.parseExpression("10 + 20");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("10 * 20");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("true and true");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("true && false");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("true or true");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("true || false");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("10 ne 20");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("10 >= 20");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("10 lt 5");
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
System.out.println(expr.getValue());

expr = parser.parseExpression("10 ge 5");
System.out.println(expr.getValue());

expr = parser.parseExpression("10 eq 10? 'condition is true': 'condition is
false'");
System.out.println(expr.getValue());

expr = parser.parseExpression("T(Thread).MIN_PRIORITY");
System.out.println(expr.getValue());

expr = parser.parseExpression("T(Integer).toString(10)");
System.out.println(expr.getValue());

User u = new User();
StandardEvaluationContext context = new StandardEvaluationContext(u);
expr = parser.parseExpression("uname?.toUpperCase()");
System.out.println(expr.getValue(context));
}
}
```

3.Variables

In SpEL, if we want to access any variable which is already existed in StandardEvaluationContext then we have to use the following syntax.

#var_Name.

If we want to set variable to StandardEvaluationContext then we have to use the following method.

```
public void setVariable(String var_Name, Object val)
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Example:**MyMath.java**

```
package com.durgasoft.beans;
public class MyMath {
    private int num1;
    private int num2;

    setXXX() and getXXX()

    public int add() {
        return num1+num2;
    }
    public int sub() {
        return num1-num2;
    }
    public int mul() {
        return num1*num2;
    }
}
```

Test.java

```
package com.durgasoft.test;
import org.springframework.expression.Expression;
import org.springframework.expression.ExpressionParser;
import org.springframework.expression.spel.standard.SpelExpressionParser;
import org.springframework.expression.spel.support.StandardEvaluationContext;
import com.durgasoft.beans.MyMath;
public class Test {
    public static void main(String[] args)throws Exception {
        MyMath math = new MyMath();
        StandardEvaluationContext context = new
StandardEvaluationContext(math);
        context.setVariable("number1", 10);
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
context.setVariable("number2", 5);
ExpressionParser parser = new SpelExpressionParser();
Expression expr1 = parser.parseExpression("num1 = #number1");
Expression expr2 = parser.parseExpression("num2 = #number2");

System.out.println(math.add());
System.out.println(math.sub());
System.out.println(math.mul());
    }
}
```

4. Method Invocations

In SpEL, we are able to declare methods and we are able to access those methods as per the requirement.

If we want to declare a user-defined method and if we want to access a user-defined method through an expression, then we have to use the following steps.

1. Create StandardEvaluationContext object.
2. Register Method with a name.
public void registerFunction(String meth_Name, Method m)
3. Prepare an Expression with method call and access it.

Example:

MyString.java

```
package com.durgasoft.beans;
public class MyString {
    public static void reverseString(String str) {
        StringBuffer sb = new StringBuffer(str);
        System.out.println(sb.reverse());
    }
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonline@training@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Test.java

```
package com.durgasoft.test;
import java.lang.reflect.Method;
import org.springframework.expression.Expression;
import org.springframework.expression.ExpressionParser;
import org.springframework.expression.spel.standard.SpelExpressionParser;
import org.springframework.expression.spel.support.StandardEvaluationContext;
public class Test {
    public static void main(String[] args) throws Exception {
        StandardEvaluationContext context = new StandardEvaluationContext();
        Class cls = Class.forName("com.durgasoft.beans.MyString");
        Method method = cls.getDeclaredMethod("reverseString", new
Class[] {java.lang.String.class});
        context.registerFunction("reverse", method);
        context.setVariable("str", "Durga Software Solutions");

        ExpressionParser parser = new SpelExpressionParser();
        Expression expr = parser.parseExpression("#reverse(#str)");
        expr.getValue(context);

        // Accessing Predefined methods
        expr = parser.parseExpression("new java.util.Date().toString()");
        System.out.println(expr.getValue());

        expr = parser.parseExpression("'Durga Software
Solutions'.toUpperCase()");
        System.out.println(expr.getValue());
    }
}
```

5.Collections

In SpEL, we are able to declare Collections and we are able to access the content from Collection objects by using the following Expression Syntax.

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Collection_Ref_Var.?Condition_Expression

Example:

City_State.java

```
package com.durgasoft.beans;

public class City_State {
    private String city;
    private String state;

    public City_State(String city, String state) {
        this.city = city;
        this.state = state;
    }
    setXXX() and getXXX()
    public String toString() {
        return city+"="+state;
    }
}
```

City_State_Collection.java

```
package com.durgasoft.beans;

import java.util.ArrayList;

public class City_State_Collection {
    private ArrayList<City_State> city_State = new ArrayList<City_State>();
    public ArrayList<City_State> getCity_State(){
        city_State.add(new City_State("Hyd","Tel"));
        city_State.add(new City_State("VJA","AP"));
        city_State.add(new City_State("Warangal","Tel"));
        city_State.add(new City_State("Vizag","AP"));
        city_State.add(new City_State("KNGR","Tel"));
    }
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,


```
        city_State.add(new City_State("TRPTI","AP"));
        return city_State;
    }
}

Test.java

package com.durgasoft.test;

import java.util.ArrayList;

import org.springframework.expression.Expression;
import org.springframework.expression.ExpressionParser;
import org.springframework.expression.spel.standard.SpelExpressionParser;
import org.springframework.expression.spel.support.StandardEvaluationContext;

import com.durgasoft.beans.City_State;
import com.durgasoft.beans.City_State_Collection;

public class Test {

    public static void main(String[] args)throws Exception {
        City_State_Collection collection = new City_State_Collection();
        StandardEvaluationContext context = new
StandardEvaluationContext(collection);
        ExpressionParser parser = new SpelExpressionParser();
        Expression expr = parser.parseExpression("city_State?[state == 'AP']");
        ArrayList<City_State> al = (ArrayList<City_State>)expr.getValue(context);
        System.out.println(al);
    }
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,