String Util Service

Summary

This skill helps to use various kinds of functions to deal with character string data required for system development. There are four kinds such as **EgovStringUtil Service** to handle character string; **EgovNumericUtil Service** to handle numbers; **EgovDateUtil Service** to handle data type, and **EgovObjectUtil Service** to create objects.

Description

1. EgovStringUtil Service

Pattern matching

It inspects whether String coincides with specific Pattern(regular expression).

Sample Source

```
@Test
public void testPatternMatch() throws Exception {
  // pattern match success
Stringstr = "abc-def";
pattern = "*-*";
assertTrue(EgovStringUtil.isPatternMatching(str, pattern));

  // pattern match failure
str = "abc";
assertTrue(!EgovStringUtil.isPatternMatching(str, pattern));
}
```

Formatting

Convert various types of data to specific String Format.

Sample Source

```
@Test
public void testTypeConversion() throws Exception {
  // int => string
assertEquals("1", EgovStringUtil.integer2string(1));

  // long => string
assertEquals("1000000000", EgovStringUtil.long2string(1000000000));

  // float => string
assertEquals("34.5", EgovStringUtil.float2string(34.5f));

  // double => string
assertEquals("34.5", EgovStringUtil.double2string(34.5));

  // string =>int
assertEquals(1, EgovStringUtil.string2integer("1"));
assertEquals(0, EgovStringUtil.string2integer(null, 0));

  // string => float
assertEquals(Float.valueOf(34.5f), Float.valueOf(EgovStringUtil.string2float("34.5")));
assertEquals(Float.valueOf(10.5f), Float.valueOf(EgovStringUtil.string2float(null, 10.5f)));

  // string => double
assertEquals(Double.valueOf(34.5), Double.valueOf(EgovStringUtil.string2double("34.5")));
```

```
assertEquals(Double.valueOf(34.5), Double.valueOf(EgovStringUtil.string2double(null, 34.5)));

  // string => long
assertEquals(100000000, EgovStringUtil.string2long("100000000"));
assertEquals(100000000, EgovStringUtil.string2long(null, 100000000));
}
```

Substring

Getting part of all strings.

Sample Source

```
@Test
public void testToSubString() throws Exception {
String source = "substring test";

assertEquals("test", EgovStringUtil.toSubString(source, 10));
assertEquals("string", EgovStringUtil.toSubString(source, 3, 9));
}
```

Trim

Removewhite characters that exists before and after it in the whole strings.

Sample Source

```
@Test
public void testStringTrim() throws Exception {
Stringstr = "   substring   ";

assertEquals("substring"   , EgovStringUtil.trim(str));
assertEquals("substring   ", EgovStringUtil.ltrim(str));
assertEquals("   substring", EgovStringUtil.rtrim(str));
}
```

Concatenate

Combine two strings to create astring.

Sample Source

```
@Test
public void testConcat() throws Exception {
String str1 = "substring";
String str2 = "test";

assertEquals("substringtest", EgovStringUtil.concat(str1, str2));
}
```

Find

Find whether there are specific String Pattern in the whole string.

Sample Source

```
@Test
public void testFindPattern() throws Exception {
String pattern = "\\d{4}-\\d{1,2}-\\d{1,2}";
```

```
  // Find the matching pattern.
  Matcher matcher = Pattern.compile(pattern).matcher("2009-02-03");
assertTrue(matcher.find());

  // Find the matching pattern.
matcher = Pattern.compile(pattern).matcher("abcdef2009-02-03abcdef");
assertTrue(matcher.find());

  // Fail to find the matching pattern.
matcher = Pattern.compile(pattern).matcher("abcdef2009-02-A3abcdef");
assertFalse(matcher.find());
}
```

2. EgovNumericUtil Service

Check number, add, subtract, multiply, divide, round up, round down function

Number Check Function

Inspect whether given character string is number format.

Sample Source

```
@Test
public void testIsNumber() throws Exception {
assertFalse(EgovNumericUtil.isNumber("abc"));
assertFalse(EgovNumericUtil.isNumber("!@"));
assertFalse(EgovNumericUtil.isNumber("ab-123"));
assertTrue(EgovNumericUtil.isNumber("-123"));
assertTrue(EgovNumericUtil.isNumber("1234"));
}
```

Add Function

Perform addition of values of 2 character strings.

Sample Source

```
@Test
public void testPlus() throws Exception {
assertEquals("400",        EgovNumericUtil.plus("151", "249"));
assertEquals("400.0000", EgovNumericUtil.plus("151.7531", "248.2469"));
assertEquals("400.000",   EgovNumericUtil.plus("151.7531", "248.2469", 3));
assertEquals("399.9654", EgovNumericUtil.plus("151.7531", "248.2123"));
assertEquals("399.966",   EgovNumericUtil.plus("151.7531", "248.2123", 3,
EgovNumericUtil.ROUND_UP));
assertEquals("399.965",   EgovNumericUtil.plus("151.7531", "248.2123", 3,
EgovNumericUtil.ROUND_DOWN));
assertEquals("399.97",    EgovNumericUtil.plus("151.7531", "248.2123", 2,
EgovNumericUtil.ROUND_HALF_UP));
}
```

Subtract Function

Perform subtraction of 2 character strings values.

Sample Source

```
@Test
public void testMinus() throws Exception {
assertEquals("89",         EgovNumericUtil.minus("240", "151"));
```

```
assertEquals("96.4938",   EgovNumericUtil.minus("248.2469", "151.7531"));
assertEquals("96.49380", EgovNumericUtil.minus("248.2469", "151.7531", 5));
assertEquals("96.4592",   EgovNumericUtil.minus("248.2123", "151.7531"));
assertEquals("96.460",    EgovNumericUtil.minus("248.2123", "151.7531", 3,
EgovNumericUtil.ROUND_UP));
assertEquals("96.459",    EgovNumericUtil.minus("248.2123", "151.7531", 3,
EgovNumericUtil.ROUND_DOWN));
assertEquals("96.46",     EgovNumericUtil.minus("248.2123", "151.7531", 2,
EgovNumericUtil.ROUND_HALF_UP));
}
```

Multiplication Function

Perform multiplication of 2 character string values.

Sample Source

```
@Test
public void testMultiply() throws Exception {
assertEquals("180",         EgovNumericUtil.multiply("15", "12"));
assertEquals("189.6135",   EgovNumericUtil.multiply("15.23", "12.45"));
assertEquals("189.61350", EgovNumericUtil.multiply("15.23", "12.45", 5));
assertEquals("189.614",    EgovNumericUtil.multiply("15.23", "12.45", 3,
EgovNumericUtil.ROUND_UP));
assertEquals("189.613",    EgovNumericUtil.multiply("15.23", "12.45", 3,
EgovNumericUtil.ROUND_DOWN));
assertEquals("189.61",     EgovNumericUtil.multiply("15.23", "12.45", 2,
EgovNumericUtil.ROUND_HALF_UP));
}
```

Divide Function

Perform division of 2 character strings.

Sample Source

```
@Test
public void testDivide() throws Exception {
assertEquals("1.25",   EgovNumericUtil.divide("15", "12"));

  Class<Exception>exceptionClass = null;
try {
        assertEquals("1.22329317269076305220883534136555", EgovNumericUtil.divide("15.23",
"12.45"));
  } catch (Exception e) {
        log.error("### Exception : " + e.toString());
        exceptionClass = (Class<Exception>) e.getClass();
  } finally {
        assertEquals(ArithmeticException.class, exceptionClass);
  }

assertEquals("1.22",   EgovNumericUtil.divide("15.23", "12.45", 5));
assertEquals("1.224", EgovNumericUtil.divide("15.23", "12.45", 3, EgovNumericUtil.ROUND_UP));
assertEquals("1.223", EgovNumericUtil.divide("15.23", "12.45", 3, EgovNumericUtil.ROUND_DOWN));
assertEquals("1.22",   EgovNumericUtil.divide("15.23", "12.45", 2,
EgovNumericUtil.ROUND_HALF_UP));
}
```

Round-up, Round-down Function

Perform Round-off, Round-up and Round-down of given values.

Sample Source

```
@Test
public void testScale() throws Exception {
assertEquals("151.754", EgovNumericUtil.setScale("151.7531", 3, EgovNumericUtil.ROUND_UP));
assertEquals("151.753", EgovNumericUtil.setScale("151.7531", 3, EgovNumericUtil.ROUND_DOWN));
assertEquals("151.753", EgovNumericUtil.setScale("151.7531", 3, EgovNumericUtil.ROUND_HALF_UP));
}
```

3. EgovDateUtil Service

This provides skills to calculate date, inquire the current date, checkthe day of the week and the date format

Calculate Date Function

This skill inquires the date calculated by adding the relevant year, month or date to the given date.

Sample Source

```
@Test
public void testCalcDate() throws Exception {
assertEquals("20100114",   EgovDateUtil.getCalcDateAsString ("2009", "3", "20", 300, "day"));
assertEquals("2010",        EgovDateUtil.getCalcYearAsString ("2009", "3", "20", 300, "day"));
assertEquals("01",          EgovDateUtil.getCalcMonthAsString("2009", "3", "20", 300, "day"));
assertEquals("14",          EgovDateUtil.getCalcDayAsString   ("2009", "3", "20", 300, "day"));

assertEquals(2010,          EgovDateUtil.getCalcYearAsInt ("2009", "3", "20", 300, "day"));
assertEquals(1,             EgovDateUtil.getCalcMonthAsInt("2009", "3", "20", 300, "day"));
assertEquals(14,            EgovDateUtil.getCalcDayAsInt   ("2009", "3", "20", 300, "day"));
}
```

Calculate the start date and end date as well as the date/millisecond number between two times.

Sample Source

```
@Test
public void testDayCount() throws Exception {
assertEquals(90,   EgovDateUtil.getDayCount("20090101", "20090401"));
assertEquals(90,   EgovDateUtil.getDayCountWithFormatter("20090101", "20090401", "yyyyMMdd"));
assertEquals(182, EgovDateUtil.getDayCountWithFormatter("2008/12/01", "2009/06/01",
"yyyy/MM/dd"));
}

@Test
public void testTimeCount() throws Exception {
assertEquals(86400000, EgovDateUtil.getTimeCount("20090401",         "20090402"));
assertEquals(60000,      EgovDateUtil.getTimeCount("20090301000000", "20090301000100"));
assertEquals(3600000,   EgovDateUtil.getTimeCount("20090301000000", "20090301010000"));
}
```

Inquiry Function of the Current Date

This inquires the current date.

Sample Source

```
@Test
public void testCurrentDate() throws Exception {
assertEquals(Calendar.getInstance().get(Calendar.YEAR),          EgovDateUtil.getCurrentYearAsInt());
```

```
assertEquals(Calendar.getInstance().get(Calendar.MONTH) + 1,
EgovDateUtil.getCurrentMonthAsInt());
assertEquals(Calendar.getInstance().get(Calendar.DAY_OF_MONTH),
EgovDateUtil.getCurrentDayAsInt());
assertEquals(Calendar.getInstance().get(Calendar.HOUR_OF_DAY),
EgovDateUtil.getCurrentHourAsInt());
assertEquals(Calendar.getInstance().get(Calendar.MINUTE),
EgovDateUtil.getCurrentMinuteAsInt());
}
```

Inquiry Function of the Day of the Week

This inquires the relevant day of the week for the date entered.

Sample Source

```
@Test
public void testGetDayOfWeek() throws Exception {
assertEquals("Sun", EgovDateUtil.getDayOfWeekAsString("2009", "03", "22"));
assertEquals("Mon", EgovDateUtil.getDayOfWeekAsString("2009", "03", "23"));
assertEquals("Tue", EgovDateUtil.getDayOfWeekAsString("2009", "03", "24"));
assertEquals("Wed", EgovDateUtil.getDayOfWeekAsString("2009", "03", "25"));
assertEquals("Thur", EgovDateUtil.getDayOfWeekAsString("2009", "03", "26"));
assertEquals("Fri", EgovDateUtil.getDayOfWeekAsString("2009", "03", "27"));
assertEquals("Sat", EgovDateUtil.getDayOfWeekAsString("2009", "03", "28"));
}
```

Inquiry of the number of relevant day of the week between two dates.

Sample Source

```
@Test
public void testGetDayOfWeek() throws Exception {
assertEquals(5,   EgovDateUtil.getDayOfWeekCount("20090301", "20090331", "Sunday"));
assertEquals(4,   EgovDateUtil.getDayOfWeekCount("20090301", "20090331", "Saturday"));
assertEquals(22, EgovDateUtil.getDayOfWeekCount("20090101", "20090531", "Sun"));
assertEquals(52, EgovDateUtil.getDayOfWeekCount("20090101", "20091231", "Sun"));
assertEquals(52, EgovDateUtil.getDayOfWeekCount("20090101", "20091231", "Fri"));
assertEquals(52, EgovDateUtil.getDayOfWeekCount("20090101", "20091231", "Sat"));
}
```

Checkup Function of the Date Format

This provides to check up on the feasibility of the relevant date format.

Sample Source

```
@Test
public void testDateFormatCheck() throws Exception {
  // If format is different, ParseException occurs
  Class<Exception>exceptionClass = null;

try {
        dateFormatCheck = EgovDateUtil.dateFormatCheck("20090300");
  } catch (Exception e) {
        exceptionClass = (Class<Exception>) e.getClass();
  } finally {
        assertEquals(ParseException.class, exceptionClass);
  }
}
```

4. EgovObjectUtil Service

This service creates objects with class names. It can also instantiate the objects in various formats including the basic creator without parameter or creator with parameter.

Instantiation Function

Instantiating the object in the format of basic creator without parameter.

Sample Source

```
@Test
public void testInstantiate() throws Exception {
Stringclass Name = "java.lang.String";
Objectobject = EgovObjectUtil.instantiate(className);

Stringstring = (String) object;
string = "eGovFramework";
assertEquals("Framework", string.substring(4));
}
```

Instantiation Function – Including Parameter of Creator

Instantiating the object in the format of creator in the format with parameter.

Sample Source

```
@Test
public void testInstantiateParamConstructor() throws Exception {
StringclassName = "java.lang.StringBuffer";
String[] types = new String[]{"java.lang.String"};
Object[] values = new Object[]{"E-Government Common Service"};

StringBuffersb = (StringBuffer)EgovObjectUtil.instantiate(className, types, values);
sb.append(" and Development Framework Construction Project");
assertEquals("E-Government Common Service and Development Framework Construction Project",
sb.toString());
}
```

N. Reference

- Jakarta Regexp