

# Anyframe Util Plugin



Version 1.1.0

저작권 © 2007-2014 삼성SDS

본 문서의 저작권은 삼성SDS에 있으며 Anyframe 오픈소스 커뮤니티 활동의 목적하에서 자유로운 이용이 가능합니다. 본 문서를 복제, 배포할 경우에는 저작권자를 명시하여 주시기 바라며 본 문서를 변경하실 경우에는 원문과 변경된 내용을 표시하여 주시기 바랍니다. 원문과 변경된 문서에 대한 상업적 용도의 활용은 허용되지 않습니다. 본 문서에 오류가 있다고 판단될 경우 이슈로 등록해 주시면 적절한 조치를 취하도록 하겠습니다.

---

|                              |    |
|------------------------------|----|
| I. Introduction .....        | 1  |
| II. Utility .....            | 2  |
| 1. DateUtil의 활용 .....        | 3  |
| 2. DigestUtil의 활용 .....      | 14 |
| 3. NumberUtil의 활용 .....      | 15 |
| 4. StringUtil의 활용 .....      | 18 |
| 5. ValidationUtil의 활용 .....  | 36 |
| 6. ThreadLocalUtil의 활용 ..... | 38 |

---

# I.Introduction

Util plugin은 Anyframe core 서비스에 포함되어 있는 Date, Digest, Number, String, Validation, ThreadLocal 유틸 활용을 위한 샘플 화면을 가지고 있는 샘플 코드로 구성되어 있다.

## Installation

Command 창에서 다음과 같이 명령어를 입력하여 Util plugin을 설치한다.

```
mvn anyframe:install -Dname=util
```

installed(mvn anyframe:installed) 혹은 jetty:run(mvn clean jetty:run) command를 이용하여 설치 결과를 확인해볼 수 있다.

| Plugin Name  | Version Range     |
|--|-------------------|
| core [ <a href="http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html">http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html</a> ] | 2.0.0 > * > 1.4.0 |

---

## II.Utility

Util plugin은 core service에서 제공하는 Date, Digest, Number, String, Validation, ThreadLocal 각각의 utility 에 대해서 쉽게 활용할 수 있는 샘플 화면을 제공한다.

| Name            | Description  |
|-----------------|--|
| DateUtil        | Joda Time을 사용하여 날짜,시간 및 요일 계산, 유효성 체크와 포맷 변경 등의 기능을 제공한다.                            |
| DigestUtil      | 해당하는 문자열에 대해서 원하는 character set 또는 base64 , 알고리즘을 사용해 인코딩/디코딩 하는 기능을 제공한다.           |
| NumberUtil      | 숫자 관련 계산,검색, 변환 및 유효성 체크 기능을 제공한다.   |
| StringUtil      | 문자열 관련 검색, 변환, 치환, 유효성 체크 등의 기능을 제공한다.   |
| ValidationUtil  | 정규 표현식을 활용한 주민번호,법인번호,사업자,외국인 등록번호등의 유효성 체크와 전화,휴대전화,이메일,카드번호 등의 포맷 유효성 체크 기능을 제공한다. |
| ThreadLocalUtil | ThreadLocal을 Map과 유사한 인터페이스로 사용할 수 있는 기능을 제공한다.                                      |

---

# 1.DateUtil의 활용

Joda Time 을 사용하여 날짜, 시간 및 요일 계산, 유효성 체크와 포맷 변경 등의 기능을 제공한다.

다음은 DateUtil이 가지고 있는 메소드에 대한 설명이다.

| Method  | Description   | Example   | Remarks  |
|---|---|---|--|
| <b>getCurrentDateTimeString()</b>                                     |   |   | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>getCurrentDateTime()<br>사용                  |
| <b>getCurrentDateTimeString(String str)</b>                           |   |   | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>getCurrentDateTime(String<br>pattern)<br>사용 |
| <b>getThisMonth ()</b>  | 현재 달과 연도를 문자열로 반환   | DateUtil.getThisMonth()   |  |
| <b>getThisYear ()</b>   | 현재 연도를 문자열로 반환  | DateUtil.getthisYear()  |  |
| <b>getDayOfWeek (String str)</b>                                      | 입력받은 일자의 요일 반환  | DateUtil.getDayOfWeek()   |  |
| <b>getDayOfWeek (String str, Boolean abbreviation, Locale locale)</b> | 입력받은 일자의 요일 반환 - Locale 정보를 받아 해당하는 언어에 대해서 약어로 보여주거나 전체 요일 형태로 보여준다. | DateUtil.getDayOfWeek("2011-02-04", true, Locale.US) = "Fri"                                  |  |
| <b>getDays (Calendar cal1, Calendar cal2)</b>                         | 입력받은 두 날짜 사이의 일자 계산   | DateUtil.getDays(new GregorianCalendar(2010, 10, 14), new GregorianCalendar(2010, 11, 28))=44 |  |
| <b>getDays (String startDate, String endDate)</b>                     | 입력받은 두 날짜 사이의 일자 계산   | DateUtil.getDays("2010-11-24", "2010-12-30")=36   |  |
| <b>getDays (String startDate, String endDate, String pattern)</b>     | 입력받은 형식의 두 날짜 사이의 일자 계산   | DateUtil.getDays("2010-11-24", "2010-12-30", "yyyy-MM-dd")=36                                 |  |
| <b>equals (Date date1, String date2)</b>                              | 입력받은 두 일자가 같은지 여부   | DateUtil.equals(new Date(1292252400000I), "2010-12-14")=true                                  |  |
| <b>equals (Date date1, String date2, String date2format)</b>          | 입력받은 두 일자가 같은지 여부   | DateUtil.equals(new Date(1292252400000I), "2010/12/14", "yyyy/MM/dd")=true                    |  |
|   |   |   |  |

| Method   | Description  | Example  | Remarks          |
|--|--|--|------------------|
| <b>equals</b> (Date date1, Date date2)   | 입력받은 두 일자가 같은지 여부  | DateUtil.equals(new Date(1292252400000l), new Date(1292252400000l))=true           |                  |
| <b>greaterThan</b> (Date date1, String date2)  | 입력받은 두 일자의 배치비교  | DateUtil.greaterThan(new Date(1292311593000l), "2010-12-02")=true                  |                  |
| <b>greaterThan</b> (Date date1, String date2, String date2format)                      | 입력받은 두 일자의 배치비교  | DateUtil.greaterThan(new Date(1292311593000l), "2010/12/02", "yyyy/MM/dd")=true    |                  |
| <b>greaterThan</b> (Date date1, Date date2)  | 입력받은 두 일자의 배치비교  | DateUtil.greaterThan(new Date(1292311593000l), new Date(1292252400000l)) =true     |                  |
| <b>greaterThan</b> (Timestamp timestamp1, Timestamp timestamp2)                        | 입력받은 두 java.sql.Timestamp 비교                                       | DateUtil.greaterThan(new Timestamp(1292311593000l), new Timestamp(1292311503000l)) | since core 1.0.4 |
| <b>greaterThan</b> (Timestamp timestamp1, String timestamp2)                           | 입력받은 java.sql.Timestamp 과 문자열과 비교                                  | DateUtil.greaterThan(new Timestamp(1292311593000l), "2012-05-21")                  | since core 1.0.4 |
| <b>greaterThan</b> (Timestamp timestamp1, String timestamp2, String timestamp2pattern) | 입력받은 java.sql.Timestamp timestamp2pattern의 형식의 문자열 비교              | DateUtil.greaterThan(new Timestamp(1292311593000l), "2012-05-21", "yyyy-MM-dd")    | since core 1.0.4 |
| <b>getEndDate</b> (String startDay, int intervalDays)                                  | 입력받은 날짜에서 intervalDays 만큼 더한 날짜 반환                                 | DateUtil.getEndDate("2011-01-01", 8)=2011-01-09                                    |                  |
| <b>addDays</b> (String str, int days)  | 입력받은 일자에 대해서 해당 일만큼 더한 일자 반환. 마이너스 일자는 입력받은 일자보다 이전의 일자로 계산해서 반환   | DateUtil.addDays("2010-12-18", 30)=2011-01-17                                      |                  |
| <b>addMonths</b> (String str, int months)  | 입력받은 일자에 대해서 해당 개월수만큼 더한 일자 반환. 마이너스 개월수는 입력받은 일자보다 이전 일자로 계산해서 반환 | DateUtil.addMonths("2010-12-18", 2)=2011-02-18                                     |                  |
| <b>addYears</b> (String str, int years)  | 입력받은 일자에 대해서 해당 년수만큼 더한 일자 반환. 마이너스 년수는 입력받은 일자보다 이전 일자로 계산해서 반환   | DateUtil.addYears("2010-12-18", 2)=2012-12-18                                      |                  |
| <b>addYearMonthDay</b> (String str, int years, int months, int days)                   | 입력된 일자에 대해서 년,월, 일만큼 가감해서 계산한 일자 반환                                | DateUtil.addYearMonthDay("2010-04-18", 2, 4, 3)=2012-08-21                         |                  |

| Method   | Description                | Example  | Remarks  |
|--|----------------------------|--|--|
| <b>getFirstDateOfMonthString</b><br>(String str)     |                            |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>getFirstDateOfMonth(String<br>str) 사용 |
| <b>getLastDateOfMonthString</b><br>(String str)      |                            |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>getLastDateOfMonth(String<br>str) 사용  |
| <b>getFirstDateOfPrevMonthString</b><br>(String str) |                            |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>getFirstDateOfPrevMonth(St<br>str) 사용 |
| <b>getLastDateOfPrevMonthString</b><br>(String str)  |                            |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>getLastDateOfPrevMonth(St<br>str) 사용  |
| <b>isDate</b> (String str)                           | 입력된 일자가 유효한 일자인<br>지 체크    | DateUtil.isDate("2010-12-01")=true               |  |
| <b>isDate</b> (String str, String pattern)           | 입력된 일자가 패턴에 맞는 유효한 일자인지 체크 | DateUtil.isDate("2010/12/01", "yyyy/MM/dd")=true |  |
| <b>isTime</b> (String str)                           | 입력된 시간이 유효한 시간인지 체크        | DateUtil.isTime("11:56")=true                    |  |
| <b>isTime</b> (String str, String pattern)           | 입력된 시간이 패턴에 맞는 유효한 시간인지 체크 | DateUtil.isTime("13:56:24", "HH:mm:ss")=true     |  |
| <b>string2Date</b> (String str)                      |                            |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>stringToDate(String<br>date) 사<br>용   |
| <b>string2Date</b> (String str, String pattern)      |                            |  | <b>deprecated</b><br>since<br>core<br>1.5.0  |

| Method  | Description             | Example                  | Remarks   |
|---|-------------------------|--------------------------|---|
|   |                         |                          | stringToDate(String date, String pattern) 사용  |
| <b>date2String</b> (Date date)  |                         |                          | <b>deprecated</b><br>since core 1.5.0<br>dateToString(Date date) 사용   |
| <b>date2String</b> (Date date, String pattern)                              |                         |                          | <b>deprecated</b><br>since core 1.5.0<br>dateToString(Date date) 사용   |
| <b>string2String</b> (String str, String basePattern, String wantedPattern) |                         |                          | <b>deprecated</b><br>since core 1.5.0<br>convertStringDateFormat(String str, String basePattern, String wantedPattern) 사용 |
| <b>getRandomDate</b> ()   | 1900~2100 사이의 랜덤한 일자 반환 | DateUtil.getRandomDate() |   |
| <b>string2SQLDate</b> (String str)  |                         |                          | <b>deprecated</b><br>since core 1.5.0<br>stringToSQLDate(String str) 사용   |
| <b>string2SQLDate</b> (String str, String pattern)                          |                         |                          | <b>deprecated</b><br>since core 1.5.0<br>stringToSQLDate(String str, String   |



| Method   | Description | Example | Remarks  |
|--|-------------|---------|--|
|  |             |         | pattern) 사용  |
| <b>string2Timestamp</b><br>(String str)                        |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>stringToTimestamp(String<br>str) 사용                             |
| <b>string2Timestamp</b><br>(String str, String<br>pattern)     |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>stringToTimestamp(String<br>str,<br>String<br>pattern)<br>사용    |
| <b>timestamp2String</b><br>(Timestamp date)                    |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>timestampToString(Timestar<br>date) 사<br>용                      |
| <b>timestamp2String</b><br>(Timestamp date,<br>String pattern) |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>timestampToString(Timestar<br>date,<br>String<br>pattern)<br>사용 |
| <b>string2Calender</b><br>(String str)                         |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>stringToCalendar(String<br>str) 사용                              |
| <b>calendar2String</b><br>(Calendar calendar)                  |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0  |

| Method  | Description                             | Example  | Remarks  |
|---|---|--|--|
|   |   |  | calendarToString(Calendar calendar) 사용                                     |
| <b>getMinutes</b><br>(Calendar cal1, Calendar cal2)             | 입력된 두 일자 사이의 분을 계산                      | DateUtil.getMinutes(new GregorianCalendar(2010, 11, 14, 12, 34, 12), new GregorianCalendar(2010, 11, 14, 13, 32, 12))=58 |  |
| <b>getMinutes</b> (String date1, String date2)                  | 입력된 두 일자 사이의 분을 계산                      | DateUtil.getMinutes("20111010120015", "20111010123015")=38   |  |
| <b>getYesterday</b> ()  | 어제 일자 반환                                | DateUtil.getYesterday()  |  |
| <b>getYesterday</b> (String pattern)                            | 포맷에 맞는 어제 일자 반환                         | DateUtil.getYesterday("yyyy.MM.dd")  |  |
| <b>getCalendar</b> ()   | 한국 시간대에 맞는 java.util.Calendar 타입의 일자 반환 | DateUtil.getCalendar()   |  |
| <b>getDates</b> (String startDay, String endDay)                | 두 일자 사이의 일자 목록 반환                       | DateUtil.getDates("2010-12-14", "2010-12-20")  |  |
| <b>getDates</b> (String startDay, String endDay, String format) | 두 일자 사이의 일자 목록 반환                       | DateUtil.getDates("2010/12/14", "2010/12/20", "yyyy/MM/dd")  |  |
| <b>getCurrentDateString</b> ()                                  |   |  | <b>deprecated</b><br>since core 1.5.0<br>getCurrentDate() 사용               |
| <b>getCurrentDateString</b> (String pattern)                    |   |  | <b>deprecated</b><br>since core 1.5.0<br>getCurrentDate(String pattern) 사용 |
| <b>getCurrentDate</b> ()  | 현재 일자를 java.sql.Date 타입으로 반환            | DateUtil.getCurrentDate()  | <b>deleted</b><br>since core 1.5.0<br>getCurrentSQLDate() 사용               |
| <b>getCurrentTime</b> ()  | 현재 시각을 java.sql.Time 타입으로 반환            | DateUtil.getCurrentTime()  | <b>deleted</b><br>since core 1.5.0   |

| Method  | Description                             | Example                                     | Remarks  |
|---|---|---|--|
|   |   |   | getCurrentTime()<br>사용   |
| <b>getCurrentTimeString</b><br>( )  |   |   | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>getCurrentTime()<br>사용            |
| <b>getCurrentTimestamp</b><br>( )   | 현재 시각을<br>java.sql.Timestamp 타입으로<br>반환 | DateUtil.getCurrentTimestamp()              | <b>deleted</b><br>since<br>core<br>1.5.0<br><br>getCurrentTimestamp()<br>사용          |
| <b>getCurrentTimestampString</b><br>( )   |   |   | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>getCurrentTimestampString()<br>사용 |
| <b>replaceYear</b> (Date<br>date, int year)   | 입력한 일자의 연도 값을 변경<br>하여 반환               | DateUtil.replaceYear(new Date(). 2012)      | since<br>core<br>1.0.4   |
| <b>getDate</b> (int year,<br>int month, int day,<br>int hourOfDay, int<br>minute, int second) | 입력한 년,월,일,시,분,초의 값<br>을 가지는 Date를 반환    | DateUtil.getDate(2012,2,3,4,10,10)          | since<br>core<br>1.0.4   |
| <b>getLastDayOfMonth</b><br>(String inputDate)  | 입력한 일자의 달의 마지막날<br>을 반환                 | DateUtil.getLastdayOfMonth("2012-05-21")    | since<br>core<br>1.0.4   |
| <b>isLastDateOfMonth</b><br>(String inputDate)  | 입력한 일자가 달의 마지막날<br>인지 여부                | DateUtil.isLastdayOfMonth("2012-05-21")     | since<br>core<br>1.0.4   |
| <b>getLastDayOfMonthInt</b><br>(String ymd)   | 입력한 일자의 달의 마지막날<br>을 반환                 | DateUtil.getLastdayOfMonthInt("2012-05-21") | since<br>core<br>1.0.4   |
| <b>isLeapYear</b> (String<br>inputDate)   | 입력한 일자가 윤년인지 여부                         | DateUtil.isLeafYear("2012-05-31")           | since<br>core<br>1.0.4   |
| <b>isLeapYear</b> (int<br>year)   | 입력한 년도가 윤년인지 여부                         | DateUtil.isLeafYear(2012)                   | since<br>core<br>1.0.4   |
| <b>getCurrentHour</b> ( )   |   |   | <b>deprecated</b><br>since<br>core<br>1.0.4  |

| Method                                     | Description            | Example   | Remarks   |
|--|------------------------|---|---|
| <b>getCurrentDateTime</b><br>( )           | 현재일시를 문자열로 반환          | DateUtil.getCurrentDateTime()                             | at core<br>1.0.4<br>deprecated<br><br>since<br>core<br>1.5.0 변경된 명<br>명규칙<br>적용에<br>따른<br><b>deprecated</b><br>해제 |
| <b>getCurrentDay</b> ( )                   |                        |   | <b>deprecated</b><br>since<br>core<br>1.0.4<br><br>getCurrentDateAsString()<br>사용                                 |
| <b>getCurrentTime</b><br>(String pattern)  |                        |   | <b>deprecated</b><br>since<br>core<br>1.0.4   |
| <b>getCurrentDay</b><br>(String pattern)   |                        |   | <b>deprecated</b><br>since<br>core<br>1.0.4   |
| <b>getTimeStamp</b> ( )                    |                        |   | <b>deprecated</b><br>since<br>core<br>1.0.4<br>getCurrentTimestamp()<br>사용  |
| <b>getDefaultDatePattern</b><br>( )        |                        |   | <b>deprecated</b><br>since<br>core<br>1.0.4   |
| <b>getFirstDateOfMonth</b><br>(String str) | 입력일 기준 이달 첫번째 일자<br>반환 | DateUtil.getFirstDateOfMonth<br>("2010-12-18")=2010-12-01 | at core<br>1.0.4<br>deprecated<br><br>since<br>core<br>1.5.0 변경된 명<br>명규칙<br>적용에<br>따른<br><b>deprecated</b><br>해제 |

| Method  | Description                           | Example   | Remarks   |
|---|---------------------------------------|---|---|
| <b>getLastDateOfMonth</b><br>(String str)   | 입력일 기준 이달 마지막 일자 반환                   | DateUtil.getLastDateOfMonth("2010-11-20")=2010-11-30                            | at core 1.0.4 deprecated<br><br>since core 1.5.0 변경된 명명규칙 적용에 따른 <b>deprecated</b> 해제 |
| <b>getFirstDateOfPrevMonth</b><br>(String str)                                      | 입력일 기준 전달의 첫번째 일자 반환                  | DateUtil.getFirstDateOfPrevMonth("2010-11-20")=2010-10-01                       | at core 1.0.4 deprecated<br><br>since core 1.5.0 변경된 명명규칙 적용에 따른 <b>deprecated</b> 해제 |
| <b>getLastDateOfPrevMonth</b><br>(String str)                                       | 입력일 기준 전달의 마지막 일자 반환                  | DateUtil.getLastDateOfPrevMonth("2010-11-20")=2010-10-31                        | at core 1.0.4 deprecated<br><br>since core 1.5.0 변경된 명명규칙 적용에 따른 <b>deprecated</b> 해제 |
| <b>convertDateFormat</b><br>(String date, String basePattern, String wantedPattern) | 입력된 문자열의 날짜 형식을 변환하여 반환한다.            | DateUtil.convertDateFormat("20101214", "yyyyMMdd", "yyyy-MM-dd") = "2010-12-14" | since core 1.5.0  |
| <b>stringToDate</b><br>(String date)  | 문자열을 java.util.Date 타입으로 변경           | DateUtil.stringToDate("2010-12-14")   | since core 1.5.0  |
| <b>stringToDate</b><br>(String date, String pattern)                                | 특정 날짜 패턴의 문자열을 java.util.Date 타입으로 변경 | DateUtil.stringToDate("2010-12-14 16:26:33", "yyyy-MM-dd HH:mm:ss")             | since core 1.5.0  |

| Method  | Description                                | Example   | Remarks          |
|---|--|---|------------------|
| <b>dateToString</b> (Date date)                           | java.util.Date 타입을 문자열로 변경                 | DateUtil.dateToString(new Date(1292311593557l))=2010-12-14                                    | since core 1.5.0 |
| <b>dateToString</b> (Date date, String pattern)           | java.util.Date 타입을 입력된 패턴에 맞는 문자열로 변경      | DateUtil.dateToString(new Date(1292311593557l), "yyyy/MM/dd")=2010/12/14                      | since core 1.5.0 |
| <b>stringToSQLDate</b> (String date)                      | 문자열을 java.sql.Date 타입으로 변환                 | DateUtil.stringToSQLDate("2010-12-14")  | since core 1.5.0 |
| <b>stringToSQLDate</b> (String date, String pattern)      | 패턴에 맞게 들어온 문자열을 java.sql.Date 타입으로 변환      | DateUtil.stringToSQLDate("2010/12/14", "yyyy/MM/dd")  | since core 1.5.0 |
| <b>stringToTimestamp</b> (String date)                    | 문자열을 java.sql.Timestamp 타입으로 변환            | DateUtil.string2Timestamp("2010-12-14")   | since core 1.5.0 |
| <b>stringToTimestamp</b> (String date, String pattern)    | 패턴에 맞게 들어온 문자열을 java.sql.Timestamp 타입으로 변환 | DateUtil.string2SQLDate("2010/12/14", "yyyy/MM/dd")   | since core 1.5.0 |
| <b>timestampToString</b> (Timestamp date)                 | java.sql.Timestamp 타입의 일자를 문자열로 변환         | DateUtil.timestamp2String(new Timestamp(1292311593000l))                                      | since core 1.5.0 |
| <b>timestampToString</b> (Timestamp date, String pattern) | java.sql.Timestamp 타입의 일자를 패턴에 맞는 문자열로 변환  | DateUtil.timestamp2String(new Timestamp(1292311593000l), "yyyy/MM/dd HH/mm")=2010/12/14 16/26 | since core 1.5.0 |
| <b>stringToCalendar</b> (String date)                     | 문자열을 java.util.Calendar 타입으로 변환            | DateUtil.string2Calendar("20101214123412")  | since core 1.5.0 |
| <b>calendarToString</b> (Calendar calendar)               | java.util.Calendar 타입의 일자를 문자열로 변환         | DateUtil.calendar2String(new GregorianCalendar(2010, 11, 14, 12, 34, 12))=20101214123412000   | since core 1.5.0 |
| <b>getCurrentSQLDate</b> ()                               | 현재 일자를 java.sql.Date 타입으로 반환               | DateUtil.getCurrentSQLDate()  | since core 1.5.0 |
| <b>getCurrentSQLTime</b> ()                               | 현재 시각을 java.sql.Time 타입으로 반환               | DateUtil.getCurrentSQLTime()  | since core 1.5.0 |
| <b>getCurrentSQLTimestamp</b> ()                          | 현재 시각을 java.sql.Timestamp 타입으로 반환          | DateUtil.getCurrentSQLTimestamp()   | since core 1.5.0 |
| <b>getCurrentDate</b> ()                                  | 현재 일자를 문자열로 반환                             | DateUtil.getCurrentDate()   | since core 1.5.0 |
| <b>getCurrentDate</b> (String pattern)                    | 현재 일자를 입력한 형식의 문자열로 반환                     | DateUtil.getCurrentDate("yyyyMMdd")   | since core 1.5.0 |
| <b>getCurrentTime</b> ()                                  | 현재 시각을 문자열로 반환                             | DateUtil.getCurrentTime()   | since core 1.5.0 |

| Method  | Description            | Example                                   | Remarks          |
|---|------------------------|---|------------------|
| <b>getCurrentDateTime</b><br>(String pattern) | 현재일시를 지정한 형식의 문자열로 반환  | DateUtil.getCurrentDateTime("yyyy-MM-dd") | since core 1.5.0 |
| <b>getCurrentTimestamp</b><br>( )             | 현재 Timestamp값을 문자열로 반환 | DateUtil.getCurrentTimestampString()      | since core 1.5.0 |

## 2.DigestUtil의 활용

해당하는 문자열에 대해서 charset 또는 base64 등 기능을 사용해 인코딩/디코딩 하는 기능을 제공한다.

다음은 DigestUtil이 가지고 있는 메소드에 대한 설명이다.

| Method  | Description                 | Example   | Remarks  |
|---|-----------------------------|---|--|
| <b>encodeCharset</b> (String str, String charsetName)     |                             |   | <b>deprecated</b><br>since core 1.5.0<br><br>convertStringCharset(String str, String charset) 사용 |
| <b>decodeCharset</b> (String str, String charsetName)     |                             |   | <b>deprecated</b><br>since core 1.5.0<br><br>convertStringCharset(String str, String charset) 사용 |
| <b>encodeBase64</b> (String str)                          | base64 기준으로 인코딩             | DigestUtil.encodeBase64("Anyframe Java Test")=QW55ZnJhbWUgSmF2YSBUZXN0                |  |
| <b>decodeBase64</b> (String str)                          | base64 기준으로 디코딩             | DigestUtil.encodeBase64("QW55ZnJhbWUgSmF2YSBUZXN0")=Anyframe Java Test                |  |
| <b>encodePassword</b> (String password, String algorithm) | 해당하는 알고리즘에 맞게 문자열 변환        | DigestUtil.encodePassword("test1234", "SHA")=9bc34549d565d9505b287de0cd20ac77be1d3f2c |  |
| <b>convertStringCharset</b> (String str, String charset)  | 문자열의 character set을 변환하여 반환 | DigestUtil.convertStringCharset("Anyframe Java Test", "US-ASCII")                     | since core 1.5.0   |



# 3.NumberUtil의 활용

숫자 관련 계산, 검색, 변환 및 유효성 체크 기능 제공

다음은NumberUtil이 가지고 있는 메소드에 대한 설명이다.

| Method   | Description                                    | Example  | Remarks                               |
|--|--|--|---------------------------------------|
| <b>checkNumberType</b><br>(String str, String check)           | 입력받은 문자열이 양수, 음수, 정수, 실수 인지 값을 받아 옳고 그름을 체크한다. | NumberUtil.checkNumberType("+1234", "positive")=true                     |                                       |
| <b>formatNumber</b><br>(double d, String format)               | 입력한 double값을 포맷에 맞는 문자열 반환                     | NumberUtil.formatNumber(12345.67d, "###,###.#")="12,345.7"               | since core 1.0.4                      |
| <b>formatNumber</b> (int intValue, String format)              | 입력한 int 값을 포맷에 맞는 문자열 반환                       | NumberUtil.formatNumber(1023412123, "###,###")=1,023,412,123             |                                       |
| <b>formatNumber</b> (long l, String format)                    | 입력한 long 값을 포맷에 맞는 문자열 반환                      | NumberUtil.formatNumber(1023412123L, "###,###")=1,023,412,123            | since core 1.0.4                      |
| <b>formatNumber</b> (String str, String format)                | 입력한 문자열값을 포맷에 맞는 문자열 반환                        | NumberUtil.formatNumber("1234567", "#,##0.000") = "1,234,567.000"        | since core 1.0.4                      |
| <b>formatNumberByLocale</b> (int intValue, Locale locale)      | 입력한 Locale에 맞는 통화 표기를 가져옴                      | NumberUtil.formatNumberByLocale(3527900, Locale.KOREA)=₩3,527,900        |                                       |
| <b>formatNumberByPoint</b> (double inputValue, int point)      | 입력한 double 값에 따른 형식으로 입력값을 변환                  | NumberUtil.formatNumberByPoint(10231023123.1213, 2) = "0,231,023,123.12" |                                       |
| <b>getRandomNumber</b> (Class<T> targetClass)                  | 랜덤한 수 조회                                       | NumberUtil.getRandomNumber(Integer.class)                                |                                       |
| <b>getRandomNumber</b> (Class<T> targetClass, boolean sign)    | 랜덤한 수 조회 - 음수인 수 조회할 수 있음                      | NumberUtil.getRandomNumber(Integer.class, false)                         |                                       |
| <b>getRandomNumber</b> (Class<T> targetClass, int fixedLength) | 지정된 길이만큼의 랜덤한 수 조회                             | NumberUtil.getRandomNumber(Integer.class, 10)                            |                                       |
| <b>getRandomNumber</b> (Class<T> targetClass, T min, T max)    | 최대 최소값 사이의 랜덤한 수 조회                            | NumberUtil.getRandomNumber(Integer.class, 10, 100)                       |                                       |
| <b>hasNumber</b> (String str)                                  | 입력된 문자열에 숫자가 있는지 확인                            | NumberUtil.hasNumber("str1ing")=true                                     |                                       |
| <b>integer2string</b> (int intValue)                           |  |  | <b>deprecated</b><br>since core 1.5.0 |

| Method  | Description                                     | Example   | Remarks  |
|---|---|---|--|
|   |   |   | integerToString(int intValue) 사용   |
| <b>isNumber</b> (String str)                                      | 입력받은 문자열이 숫자인지 체크                               | NumberUtil.isNumber("12312312")==true                 |  |
| <b>nullToZero</b> (BigDecimal bgint)                              | 입력받은 숫자가 null이면 ""로 변환                          | NumberUtil.nullToZero(null)=""                        | <b>deprecated</b><br>since core 1.5.0<br>bigDecimalToString(BigDecimal bgint) 사용 |
| <b>replaceNumber</b> (double source, int target, int replacement) | 입력받은 수에 대해 target 숫자를 찾아서 replacement 숫자로 바꿔준다. | NumberUtil.replaceNumber(123456.2, 234, 567)=156756.2 |  |
| <b>string2integer</b> (String str)                                |   |   | <b>deprecated</b><br>since core 1.5.0<br>stringToInteger(String str) 사용          |
| <b>toKorean</b> (char cNumber)                                    | 0-9의 char값을 한글로 변환                              | NumberUtil.toKorean(1) = "일"                          | <b>deprecated</b><br>since core 1.5.0<br>대체메소드 지원하지 않음.                          |
| <b>toKorean</b> (int i)   | 0-9의 int값을 한글로 변환                               | NumberUtil.toKorean(1) = "일"                          | <b>deprecated</b><br>since core 1.5.0<br>대체메소드 지원하지 않음.                          |
| <b>toKorean</b> (String str)                                      | "0"-"9"의 문자열을 한글로 변환                            | NumberUtil.toKorean("1") = "일"                        | <b>deprecated</b><br>since core 1.5.0<br>대체메소드 지원하지 않음.                          |

| Method  | Description  | Example   | Remarks  |
|---|--|---|--|
| <b>toLeftZeroPaddingString</b><br>(int i, int len)                        | integer 값의 길이에 len 만큼 0을 채운 문자열을 반환한다.                 | NumberUtil.toLeftZeroPaddingString(12, 3) = "012"     | <b>deprecated</b><br>since core 1.5.0<br>intToZeroPadString(int i, int len) 사용 |
| <b>bigDecimalToString</b><br>(BigDecimal bgint)                           | 입력받은 bigDecimal이 null이면 ""로 변환                         | NumberUtil.bigDecimalToString(null) = ""              | since core 1.5.0   |
| <b>stringToBigDecimal</b><br>(String str)                                 | 입력받은 문자열을 bigDecimal로 변환                               | NumberUtil.stringToBigDecimal("1")                    | since core 1.5.0   |
| <b>intToString</b> (int number)   | 입력받은 int를 문자열로 변환                                      | NumberUtil.intToString(1)="1"                         | since core 1.5.0   |
| <b>stringToInt</b> (String number)  | 입력받은 String을 int로 변환<br>null이거나 "" 인 경우 0반환            | NumberUtil.stringToInt("")=0                          | since core 1.5.0   |
| <b>stringToInt</b> (String number, int defaultValue)                      | 입력받은 String을 int로 변환<br>null이거나 "" 인 경우 defaultValue반환 | NumberUtil.intToString("", -1)=-1                     | since core 1.5.0   |
| <b>substringToInt</b><br>(String number, int beginIndex, int size)        | 입력한 문자열의 일부분을 int로 변환한다.                               | NumberUtil.substringToInt("123456789", 5, 3) = 678    | since core 1.5.0   |
| <b>isDigit</b> (String number)  | 입력한 문자열이 숫자로만 구성되어 있는지 여부 체크                           | NumberUtil.isDigit("1234A")=false                     | since core 1.5.0   |
| <b>isNotDigit</b> (String number)   | 입력한 문자열에 숫자가 아닌 다른 문자가 존재하는지 여부 체크                     | NumberUtil.isNotDigit("12345ABC") = true              | since core 1.5.0   |
| <b>substringToBigDecimal</b><br>(String number, int beginIndex, int size) | 입력한 문자열의 일부분을 BigDecimal로 변환한다.                        | NumberUtil.subStringToBigDecimal("123456789", 5, 3)   | since core 1.5.0   |
| <b>intToZeroPadString</b><br>(int number, int size)                       | integer 값의 길이에 size 만큼 0을 채운 문자열을 반환한다.                | NumberUtil.toLeftZeroPaddingString(12, 3) = "012"     | since core 1.5.0   |
| <b>formatNumber</b><br>(short number, String format)                      | short 값을 입력한 형식의 문자열로 변환한다.                            | NumberUtil.formatNumber(12345, "###,###") => "12,345" | since core 1.5.0   |

## 4.StringUtil의 활용

문자열 관련 검색, 변환, 치환, 유효성 체크 등의 기능을 제공한다.

다음은 StringUtil이 가지고 있는 메소드에 대한 설명이다.

| Method  | Description  | Example   | Remarks  |
|---|--|---|--|
| <b>addSpace</b> (String str, int length)                                | 주어진 String에 주어진 길이만큼 공백 문자열을 붙인다. - 주어진 길이가 0보다 작을 경우 무시된다.  | StringUtil.addSpace("12345", 5)="12345 "  | since core 1.0.4   |
| <b>addStringToArray</b> (String array[], String str)                    | 주어진 String[]에 하나의 String을 추가한다.                              | StringUtil.addStringToArray(test, "ddd")  | since core 1.0.4   |
| <b>applyInitialLaw</b> (String str)                                     | 입력된 문자열에 두음법칙을 적용하여 반환한다.                                    | StringUtil.applyInitialLaw("림업예제")="임업예제"   | since core 1.0.4   |
| <b>arrayToDelimitedString</b> (Object array[], String delimiter)        | Object[]를 입력으로 받아 delimiter로 각 element를 연결하여 String을 생성한다.   | String[] test = {"aaa", "bbb", "ccc"}; StringUtil.arrayToDelimitedString(test, ",")="aaa,bbb,ccc" | since core 1.0.4   |
| <b>arrayToCommaDelimitedString</b> (Object array[])                     |  |   | <b>deprecated</b><br>since core 1.5.0<br><br>arrayToDelimitedString(Object array) 사용                       |
| <b>asteriskToSpace</b> (String str)                                     | 입력인자로 전달된 문자열에 '*'나 '**'가 있으면 공백으로 변환한다.                     | StringUtil.convertAsteriskToSpace("test**space")="test space"                                     | since core 1.0.4   |
| <b>changeFirstCharacterCase</b> (boolean capitalize, String str)        | 주어진 String의 첫번째 글자를 대문자나 소문자로 변환한다.                          | StringUtil.changeFirstCharacterCase(true, "abcd")="Abcd"  | since core 1.0.4   |
| <b>collectionToCommaDelimitedString</b> (Collection<String> collection) |  |   | <b>deprecated</b><br>since core 1.5.0<br><br>collectionToDelimitedString(Collection<String> collection) 사용 |
| <b>commaDelimitedStringToSet</b> (String str)                           | 주어진 String에 대해 ", "(delimiter)를 이용하여 tokenize한 후 Set으로 뽑아낸다. | StringUtil.commaDelimitedStringToSet("aaa,bbb,ccc")   | since core 1.0.4   |
| <b>commaDelimitedStringToStringArray</b> (String str)                   |  |   | <b>deprecated</b><br>since core 1.5.0  |

| Method  | Description                                 | Example   | Remarks  |
|---|---|---|--|
|   |   |   | tokenizeToStringArray(String str) 사용   |
| <b>compareTo</b> (String sourceStr, String anotherStr)        | 두 스트링을 비교한다.                                | StringUtil.compareTo(String sourceStr, String anotherStr)                   |  |
| <b>containsInvalidChars</b> (String str, char[] invalidChars) |   |   | <b>deprecated</b><br>since core 1.5.0<br>containsAny(String str, char[] chars) 사용                  |
| <b>containsInvalidChars</b> (String str, String invalidChars) |   |   | <b>deprecated</b><br>since core 1.5.0<br>containsAny(String str, String chars) 사용                  |
| <b>containsMaxSequence</b> (String str, String maxSeqNumber)  |   |   | <b>deprecated</b><br>since core 1.5.0<br>containsMaxOccurences(String str, String maxSeqNumber) 사용 |
| <b>convertToCamelCase</b> (String underscore)                 | 입력된 under score 형태의 문자열을 camel case 형태로 변환  | StringUtil.convertToCamelCase("anyframe_java_test")="anyframeJavaTest"      |  |
| <b>convertToCamelCase</b> (String targetString, char posChar) | 주어진 char의 형태에 맞게 잘라진 문자열을 camel case 형태로 변환 | StringUtil.convertToCamelCase("anyframe-java-test", '-')="anyframeJavaTest" |  |
| <b>convertToUnderScore</b> (String camelCase)                 | camel case 형태의 문자열을 under score 형태의 문자열로 변환 | StringUtil.convertToUnderScore("anyframeJavaTest")="anyframe_java_test"     |  |
| <b>countPattern</b> (String str, String pattern)              |   |   | <b>deprecated</b><br>since core 1.5.0  |

| Method   | Description  | Example  | Remarks   |
|--|--|--|---|
|  |  |  | countMatches(String str, String sub) 사용   |
| <b>decode</b> (String source, String target, String result, String base) | 첫번째 문자열과 두번째 문자열을 비교해서 같으면 세번째 문자열을 다르면 네번째 문자열을 반환한다. | StringUtil.decode("Java", "Good", "Bad")="bad" | "Test",   |
| <b>deleteChars</b> (String str, String chars)                            |  |  | <b>deprecated</b><br>since core 1.5.0<br><br>deleteAny(String str, String charsToDelete) 사용         |
| <b>deletePattern</b> (String str, String pattern)                        |  |  | <b>deprecated</b><br>since core 1.5.0<br><br>deleteMatches(String str, String substr) 사용            |
| <b>delimitedStringToStringArray</b> (String str, String delim)           |  |  | <b>deprecated</b><br>since core 1.5.0<br><br>tokenizeToStringArray(String str, String delimiter) 사용 |
| <b>fillString</b> (String originalStr, char ch, int ciphers)             |  |  | <b>deprecated</b><br>since core 1.5.0<br><br>leftPad(String str, int size, String padStr) 사용        |

| Method  | Description                                  | Example                         | Remarks  |
|---|--|---------------------------------|--|
| <b>getByteLength</b><br>(char charat)                           | 인자로 받은 character 의 바이트 길이를 반환한다.             | StringUtil.getByteLength('a')   |  |
| <b>getByteLength</b><br>(String str)                            | 인자로 받은 문자열에 대해서 byte 단위에 대해서 길이 계산해서 총 길이 반환 | StringUtil.getByteLength("abc") | since<br>core<br>1.0.4   |
| <b>getContainsCount</b><br>(String str, char[] chars)           |  |                                 | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>대체메<br>소드 지<br>원하지<br>않음.                                   |
| <b>getContainsCount</b><br>(String str, String sub)             |  |                                 | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>countMatches(String<br>str,<br>String<br>sub)사용             |
| <b>getContainsCountIgnoreCase</b><br>(String str, String sub)   |  |                                 | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>countMatchesIgnoreCase(Str<br>str,<br>String<br>sub) 사<br>용 |
| <b>getContainsCountIgnoreCase</b><br>(String str, char[] chars) |  |                                 | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>대체메<br>소드 지<br>원하지<br>않음.                                   |
| <b>getCutString</b><br>(String str, int length)                 |  |                                 | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>left(String<br>str, int<br>length)<br>사용                    |

| Method  | Description   | Example  | Remarks   |
|---|---|--|---|
| <b>getLastString</b><br>(String origStr, String strToken)           | 입력된 문자열을 주어진 token에 대해서 분리 후 마지막 문자열 반환               | StringUtil.getLastString("Anyframe_Java_Test", "_")="Test"               |   |
| <b>getLength</b> (String str)                                       | 전달받은 스트링의 길이를 반환                                      | StringUtil.getLength("길이")=2   |   |
| <b>getRandomString</b><br>(int count)                               | 특정한 길이 만큼의 랜덤한 문자열을 반환한다.                             | StringUtil.getRandomString(8)  |   |
| <b>getRandomString</b><br>(int minSize, int maxSize)                | 최소, 최대 자리수 사이의 랜덤한 문자열을 반환한다.                         | StringUtil.getRandomString(10, 15)                                       |   |
| <b>getRandomString</b><br>(int count, char startChar, char endChar) | 특정한 알파벳 사이의 지정된 길이만큼의 랜덤한 문자열을 반환한다.                  | StringUtil.getRandomString(10, 'a', 'e')                                 |   |
| <b>getRandomStringByCharset</b><br>(int count, String charset)      | 특정한 길이만큼의 주어진 캐릭터 셋 문자열을 반환한다.                        | StringUtil.getRandomStringByCharset(20, "UTF-8")                         |   |
| <b>getRandomStringByKorean</b><br>(int count)                       | 특정한 길이만큼의 한글 문자열을 반환한다.                               | StringUtil.getRandomStringByKorean(20)                                   |   |
| <b>getStringArray</b><br>(String str, String strToken)              |   |  | <b>deprecated</b><br>since core 1.5.0<br><br>tokenizeToStringArray(String str, String delimiter) 사용 |
| <b>getTokens</b> (String lst)                                       | 입력된 문자열을 ,(콤마)에 대해서 분리 후 List<String>으로 반환            | StringUtil.getTokens("Anyframe,Java,Test")                               |   |
| <b>getTokens</b> (String lst, String separator)                     | 입력된 문자열을 주어진 separator에 대해서 분리 후 List<String>으로 반환    | StringUtil.getTokens("Anyframe/Java/Test", "/")                          |   |
| <b>hasLength</b> (String inputString)                               | 주어진 String 객체가 0보다 큰 길이를 가지고 있는지 검사한다.                | StringUtil.hasLength("222")  |   |
| <b>hasText</b> (String str)   | 주어진 String 객체가 Whitespace가 아닌 문자를 가지고 있는지 검사한다.       | StringUtil.hasText(" test ") = true ,<br>StringUtil.hasText(" ") = false | since core 1.0.4  |
| <b>hexToString</b> (String str)                                     | 코드를 받아 문자열로 변환함 (유니코드)                                | StringUtil.hexToString("1111")="표"                                       | since core 1.0.4  |
| <b>indexOf</b> (String str, String search)                          | 기준 문자열에서 찾고자 하는 문자열이 포함되어 있는 경우 그 첫번째 문자열의 인덱스를 반환한다. | StringUtil.indexOf("Anyframe Java Test", "Java")=9                       |   |



| Method  | Description   | Example  | Remarks  |
|---|---|--|--|
| <b>indexOfIgnoreCase</b> (String str, String search)  | 대소문자를 구분없이 기준 문자열에서 찾고자 하는 문자열이 포함되어 있는 경우 그 첫 번째 문자열의 인덱스를 반환한다. | StringUtil.indexOfIgnoreCase("Anyframe Java Test", "java")=9 |  |
| <b>integer2string</b> (int integer)                   |   |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.intValueToString(int integer) 사용 |
| <b>isAlpha</b> (String str)                           |   |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>isLetter(String str) 사용                     |
| <b>isAlphaNumeric</b> (String str)                    |   |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>isLetterOrDigit(String str) 사용              |
| <b>isDigit</b> (String str)                           |   |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.isDigit(String str) 사용           |
| <b>isEmpty</b> (String str)                           | 주어진 문자열이 null 또는 공백일 경우 참 반환                                      | StringUtil.isEmpty("")=true                                  |  |
| <b>isEmptyTrimmed</b> (String str)                    | trim한 문자열이 null 또는 공백일 경우 참 반환                                    | StringUtil.isEmptyTrimmed(" ")                               |  |
| <b>isFormattedString</b> (String str, String pattern) | 주어진 String이 특정한 포맷 (Regular Expression) 으로 구성되었는지를 검사한다.          | StringUtil.isFormattedString("no", "false no")=true          | since<br>core<br>1.0.4   |
| <b>isHangul</b> (char str)                            | 주어진 character가 한글인지의 여부를 판별한다.                                    | StringUtil.isHangul("가")=true                                | since<br>core<br>1.0.4   |
| <b>isHangul</b> (String str, boolean checkForAll)     | 주어진 String에 대해서, 한글로만 되어 있는지 혹은 한글이 포함되어 있는지를 판별한다.               | StringUtil.isHangul("가abc", false)=true                      | since<br>core<br>1.0.4   |

| Method                                | Description                                   | Example                                       | Remarks   |
|---------------------------------------|---|---|---|
| <b>isLetter</b> (String str)          | 주어진 String이 '문자'로만 구성되어 있는지를 판별한다.            | StringUtil.isLetter("test가나")=true            | since<br>core<br>1.0.4<br><br><b>since<br/>core<br/>1.5.0 :</b><br>"" 에 대<br>해서<br><b>false</b> 를<br>리턴하<br>도록 변<br>경 |
| <b>isLetterOrDigit</b> (String str)   | 주어진 String이 '문자'나 '숫자'로만 구성되어 있는지를 판별한다.      | StringUtil.isLetterOrDigit("12가나@#%")=false   | since<br>core<br>1.0.4<br><br><b>since<br/>core<br/>1.5.0 :</b><br>"" 에 대<br>해서<br><b>false</b> 를<br>리턴하<br>도록 변<br>경 |
| <b>isNotEmpty</b> (String str)        | 주어진 문자열이 null 또는 공백이 아닐 경우 참 반환               | StringUtil.isNotEmpty("abc")=true             |   |
| <b>isNotNumeric</b> (String str)      |   |   | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.isNotDigit(String str) 사용                               |
| <b>isSpaceOnly</b> (String str)       |   |   | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>isWhiteSpaceOnly(String str) 사용                                    |
| <b>left</b> (String str, int len)     | 주어진 String 객체에 대해서 주어진 길이만큼 왼쪽 부분을 떼어 반환한다.   | StringUtil.left("1234567", 3)="123"           | since<br>core<br>1.0.4  |
| <b>leftPad</b> (String str, int size) | 해당하는 문자열에 대해서 입력된 길이만큼 부족한 길이를 왼쪽부터 공백으로 채워넣음 | StringUtil.leftPad("Anyframe", 12)="Anyframe" |   |

| Method   | Description  | Example   | Remarks   |
|--|--|---|---|
| <b>leftPad</b> (String str, int size, char padChar)  | 해당하는 문자열에 대해서 입력된 길이만큼 부족한 길이를 왼쪽부터 지정된 character로 채워넣는다.      | StringUtil.leftPad("Anyframe", 'a')="aaaaAnyframe"              | 12,   |
| <b>leftPad</b> (String str, int size, String padStr) | 해당하는 문자열에 대해서 입력된 길이만큼 부족한 길이를 왼쪽부터 지정된 문자열로 채워넣는다.            | StringUtil.leftPad("Anyframe", "Java")="JavaAnyframe"           | 12,   |
| <b>leftTrim</b> (String str)                         | 문자열의 왼쪽의 공백 문자열 제거   | StringUtil.leftTrim(" Anyframe Java Test")="Anyframe Java Test" |   |
| <b>newLineToSpace</b> (String str)                   | CRLF(newLine)가 포함된 문자열을 입력인자로 받아 CRLF(개행문자)를 SPACE로 변환하여 리턴한다. | StringUtil.newLineToSpace("\r\n test")=" test"                  | since core 1.0.4  |
| <b>null2str</b> (String org, String converted)       |  |   | <b>deprecated</b><br>since core 1.5.0<br><br>nullToString(String str, String defaultStr) 사용 |
| <b>null2str</b> (String org)                         |  |   | <b>deprecated</b><br>since core 1.5.0<br><br>nullToString(String str) 사용                    |
| <b>nullToEmpty</b> (String str)                      | 주어진 String 객체를 검사하여 null일 경우 "" 을 반환하고, 아니면 원본을 반환한다.          | StringUtil.nullToEmpty(test) = ""                               | since core 1.0.4  |
| <b>numberFormat</b> (double d, String format)        |  |   | <b>deprecated</b><br>since core 1.5.0<br><br>NumberUtil.formatNumber(d, String format) 사용   |
| <b>numberFormat</b> (float f, String format)         |  |   | <b>deprecated</b><br>since core 1.5.0<br><br>NumberUtil.formatNumber(f, String format) 사용   |

| Method   | Description | Example | Remarks  |
|--|-------------|---------|--|
|  |             |         | format)<br>사용  |
| <b>numberFormat</b> (int i, String format)           |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.formatNumber(in<br>i, String<br>format)<br>사용              |
| <b>numberFormat</b> (long l, String format)          |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.formatNumber(l<br>l, String<br>format)<br>사용               |
| <b>numberFormat</b> (short s, String format)         |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.formatNumber(s<br>s, String<br>format)<br>사용               |
| <b>nl</b> (Object inputObject, Object defaultObject) |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>nullToObject(Object<br>inputObject,<br>Object<br>defaultObject)<br>사용 |
| <b>nl</b> (String inputString, String defaultString) |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>nullToString(String<br>inputString,<br>String<br>defaultString)<br>사용 |
| <b>padding</b> (int size, char padChar)              |             |         | <b>deprecated</b><br>since   |

| Method  | Description  | Example  | Remarks   |
|---|--|--|---|
|   |  |  | core<br>1.5.0<br><br>repeat(int<br>size,<br>char ch)<br>사용  |
| <b>pathClean</b> (String<br>path)                         | String으로 표현된 path 정보를 표준화한다.                       | StringUtil.pathClean("../aaaa\\bbbb\\<br>\\cccc\\\\dddd")="aaaa/bbbb/cccc/<br>dddd"        | since<br>core<br>1.0.4  |
| <b>pathEquals</b> (String<br>path1, String<br>path2)      | 주어진 두 개의 path 정보 String에 대해서, 표준화 후 같은 경로인지를 판별한다. | StringUtil.pathEquals("../aaaa\\bbbb\\<br>\\cccc\\\\dddd", "aaaa/bbbb/cccc/<br>dddd")=true | since<br>core<br>1.0.4  |
| <b>randomAlphabetic</b><br>(int randomLength)             | 인자로 주어진 길이만큼의 랜덤 알파벳 문자열을 생성한다.                    | StringUtil.randomAlphabetic(10)  | since<br>core<br>1.0.4  |
| <b>removeAll</b> (String<br>str, String<br>charsToDelete) |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>deleteAny(String<br>str,<br>String<br>charsToDelete)       |
| <b>removeChar</b> (String<br>str, char remove)            |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>deleteAny(String<br>str, char<br>charToDelete)<br>사용       |
| <b>removeCharAll</b><br>(String str)                      |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>deleteAny(String<br>str,<br>char[]<br>charsToDelete)<br>사용 |
| <b>removeCharAll</b><br>(String str, char[]<br>chars)     |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0   |

| Method  | Description                                  | Example  | Remarks  |
|---|--|--|--|
|   |  |  | deleteAny(String str, char[] charsToDelete) 사용   |
| <b>removeEscapeChar</b><br>(String input)                             |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>htmlUnEscape(String input) 사용                             |
| <b>removeWhitespace</b><br>(String str)                               | 문자열의 모든 공백 문자열 제거                            | StringUtil.removeWhitespace("Anyframe Java Test")="AnyframeJavaTest"   |  |
| <b>replace</b> (String str, String replacedStr, String replaceStr)    | 입력된 문자열에 대해서 해당하는 character를 찾아 주어진 문자열로 변경  | StringUtil.replace("Anyframe/Common", "/", " ")="Anyframe Common"  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>찾는 문자열이 존재하지 않는 경우 입력된 문자열을 그대로 리턴하도록 수정. 대체 메소드 지원하지 않음. |
| <b>replaceAll</b> (String source, String regex, String replacement)   | 입력된 문자열이 주어진 문자열과 일치하는 모든 문자열을 바꿔야할 문자열로 변경  | StringUtil.replaceAll("Anyframe Java Test", "Anyframe", "Enterprise")="Enterprise Java Test"   |  |
| <b>replaceFirst</b> (String source, String regex, String replacement) | 입력된 문자열이 주어진 문자열과 일치하는 첫번째 문자열을 바꿔야할 문자열로 변경 | StringUtil.replaceFirst("Anyframe Java Test Anyframe Java Test", "Anyframe", "Enterprise")="Enterprise Java Test Anyframe Java Test" |  |
| <b>replaceHtmlEscape</b><br>(String input)                            |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0  |

| Method   | Description   | Example   | Remarks  |
|--|---|---|--|
|  |   |   | htmlEscape(String input) 사용  |
| <b>replaceLast</b> (String source, String regex, String replacement)         | 입력된 문자열이 주어진 문자열과 일치하는 마지막 문자열을 바꿔야할 문자열로 변경        | StringUtil.replaceLast("Anyframe Java Test Anyframe Java Test", "Anyframe", "Enterprise")="Anyframe Java Test Enterprise Java Test" |  |
| <b>replacePattern</b> (String text, String searchString, String replacement) |   |   | <b>deprecated</b><br>since core 1.5.0<br><br>replaceAll(String str, String regex, String replacement) 사용 |
| <b>reverse</b> (String str)  | 입력된 문자열의 순서를 반대로 바꿈                                 | StringUtil.reverse("Anyframe Java Test")="tseT avaJ emarfynA"   | since core 1.0.4   |
| <b>right</b> (String str, int len)   | 주어진 String 객체에 대해서 주어진 길이만큼 오른쪽 부분을 떼어 반환한다.        | StringUtil.right("1234567", 3)="567"  | since core 1.0.4   |
| <b>rightPad</b> (String str, int size)                                       | 해당하는 문자열에 대해서 입력된 길이만큼 부족한 길이를 오른쪽부터 공백으로 채워넣는다.    | StringUtil.rightPad("bat", 5) = "bat "  |  |
| <b>rightPad</b> (String str, int size, char padChar)                         | 해당하는 문자열에 대해서 입력된 길이만큼 부족한 길이를 오른쪽부터 지정된 문자로 채워넣는다. | StringUtil.rightPad("bat", 5, 'z') = "batzz"  |  |
| <b>rightPad</b> (String str, int size, String padChar)                       | 해당하는 문자열에 대해서 입력된 길이만큼 부족한 길이를 오른쪽부터 지정된 문자로 채워넣는다. | StringUtil.rightPad("bat", 5, "yz") = "batyz"   |  |
| <b>rightTrim</b> (String str)  | 문자열의 오른쪽의 공백 문자열 제거                                 | StringUtil.rightTrim("Anyframe Java Test")="Anyframe Java Test"   |  |
| <b>split</b> (String str, char separator)                                    |   |   | <b>deprecated</b><br>since core 1.5.0<br><br>tokenizeToStringArray(String str, char delimiter) 사용        |

| Method   | Description | Example | Remarks   |
|--|-------------|---------|---|
| <b>splitHead</b> (String str, int size)            |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>left(String str, int size) 사용                |
| <b>splitHeadWithEllipsis</b> (String str, int len) |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>abbreviateFromLeft(String str, int size) 사용  |
| <b>splitTail</b> (String str, int size)            |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>right(String str, int size) 사용               |
| <b>splitTailWithEllipsis</b> (String str, int len) |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>abbreviateFromRight(String str, int size) 사용 |
| <b>string2integer</b> (String str)                 |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.stringToInt(String str) 사용        |
| <b>stringToBigDecimal</b> (String str)             |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.stringToBigDecimal(str) 사용        |



| Method  | Description  | Example  | Remarks  |
|---|--|--|--|
| <b>stringToBigDecimal</b><br>(String str, int pos, int len)   |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.substringToBigDecimal(str, int pos, int len) 사용            |
| <b>stringToHex</b> (String str)   | 문자열을 받아 해당하는 hex 코드로 만들어 반환한다.                               | StringUtil.stringToHex("123")="003100390003"   | core<br>1.0.4  |
| <b>stringToNumn</b><br>(String str)   |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.stringToInt(String number) 사용                              |
| <b>stringToNumn</b><br>(String str, int pos, int len)   |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>NumberUtil.substringToInt(String number, int beginIndex, int size) 사용 |
| <b>swapFirstLetterCase</b><br>(String str)  | 첫번째 문자를 대문자로 변경  | StringUtil.swapFirstLetterCase("java")="Java"  |  |
| <b>toBusinessNoPattern</b><br>(String str)  | 주어진 String이 10자리가 아닐 경우, ""를 return한다.                       | StringUtil.toBusinessNoPattern("1111111111")="111-11-11111"  | core<br>1.0.4  |
| <b>tokenizeToStringArray</b><br>(String str, String delimiter, boolean trimTokens, boolean ignoreEmptyTokens) | 주어진 String에 대해서 delimiter를 이용하여 tokenize 한 후 String[]로 뽑아낸다. | String[] test; test = StringUtil.tokenizeToStringArray("aaa.bbbccc.ddd", ".", true, true) => test[0]="aaa", test[1]="bbb"... | since<br>core<br>1.0.4   |
| <b>toLowercase</b><br>(String str)  |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>toLowerCaseFirstLetter(String str) 사용                                 |

| Method   | Description  | Example  | Remarks  |
|--|--|--|--|
| <b>toSocialSecuNoPattern</b><br>(String str)                   | 주어진 13자리 숫자 String을 "111111-1111111" 형태의 주민등록번호 포맷으로 변환한다.                       | StringUtil.toSocialSecuNoPattern("1111111111111")="111111-1111111"       | <b>deprecated</b><br>since<br>core<br>1.0.4  |
| <b>toTelephoneNumberFormat</b><br>(String str)                 | 주어진 문자열로 부터 숫자만 추출하여 '-'가 포함된 전화번호 형태의 문자열로 포맷팅하여 리턴한다.                          | StringUtil.toTelephoneNumberFormat("1234567")="23-4567"                  | <b>deprecated</b><br>since<br>core<br>1.0.4  |
| <b>toUpperCase</b><br>(String inputString)                     |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>toUpperCaseFirstLetter(String inputString)<br>사용        |
| <b>toZipCodePattern</b><br>(String str)                        | 주어진 6자리 숫자 String을 "111-111" 형태의 우편번호 포맷으로 변환한다.                                 | StringUtil.toZipCodePattern("111111")="111-111"                          | <b>deprecated</b><br>since<br>core<br>1.0.4  |
| <b>trim</b> (String origString, String trimString)             |  |  | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>deleteFirstMatches(String str, String deletedStr)<br>사용 |
| <b>trimEquals</b> (String str1, String str2)                   | 주어진 두 개의 String 객체에 대해서, trim()후 같은지를 비교한다.                                      | StringUtil.trimEquals("test", "test")=true                               | since<br>core<br>1.0.4   |
| <b>unqualify</b> (String qualifiedName)                        | Qualified Name으로 표현된 String을 받아서 Unqualified Name 형태의 String으로 변환한다. 기본 구분자는 '.' | StringUtil.unqualify("A-11")="A-11"<br>StringUtil.unqualify("B.10")="11" | , since<br>core<br>1.0.4   |
| <b>String unqualify</b> (String qualifiedName, char separator) | Qualified Name으로 표현된 String을 받아서 Unqualified Name 형태의 String으로 변환한다.             | StringUtil.unqualify("A-11", '-')="11"                                   | since<br>core<br>1.0.4   |
| <b>decodeString</b><br>(String str)                            |  |  | <b>deprecated</b><br>since<br>core<br>1.0.4  |
| <b>encodePassword</b><br>(String password, String algorithm)   |  |  | <b>deprecated</b><br>since<br>core<br>1.0.4  |

| Method   | Description   | Example   | Remarks   |
|--|---|---|---|
| <b>encodeString</b><br>(String str)                            |   |   | <b>deprecated</b><br>since<br>core<br>1.0.4   |
| <b>isNumeric</b> (String<br>str)                               |   |   | <b>deprecated</b><br>since<br>core<br>1.0.4   |
| <b>isPatternMatching</b><br>(String str, String<br>pattern)    | 문자열이 입력된 정규식 패턴<br>에 맞는지 체크한다. *는 전체<br>문자를 표현한다.                               | StringUtil.isPatternMatching("abc", "*-<br>*")=true   | at core<br>1.0.4<br>deprecated,<br>since<br>core<br>1.5.0 변<br>경된 명<br>명규칙<br>적용에<br>따른<br>deprecated<br>해제 |
| <b>countMatches</b><br>(String str, char[]<br>chars)           | 주어진 문자열에 대해서 해당<br>하는 캐릭터배열이 포함되어<br>있는 숫자 반환                                   | StringUtil.countMatches("Anyframe Java<br>Test", new char[] {'a'})=3                            | since<br>core<br>1.5.0  |
| <b>countMatches</b><br>(String str, String<br>sub)             | 주어진 문자열에 대해서 해당<br>하는 문자열이 포함되어 있는<br>숫자 반환                                     | StringUtil.countMatches("Anyframe Java<br>Test", "a")=3   | since<br>core<br>1.5.0  |
| <b>countMatchesIgnoreCase</b><br>(String str, char[]<br>chars) | 주어진 문자 구분없이 주어진 문<br>자열에 대해서 해당하는 캐릭<br>터 배열이 포함되어 있는 숫자<br>를 반환                | StringUtil.getContainsCountIgnoreCase("Anyframe<br>Java Test", new char[] { 't', 'e' })=1       | Anyframe<br>core<br>1.5.0   |
| <b>countMatchesIgnoreCase</b><br>(String str, String<br>sub)   | 주어진 문자 구분없이 주어진 문<br>자열에 대해서 해당하는 문자<br>열이 포함되어 있는 갯수를 반<br>환                   | StringUtil.getContainsCountIgnoreCase("Anyframe<br>Java Test", "test") = 1                      | Anyframe<br>core<br>1.5.0   |
| <b>nullToObject</b><br>(Object obj, Object<br>defaultObj)      | 주어진 Object가 null이 아닐<br>경우 그 Object를 반환하고,<br>null일 경우 default Object를<br>반환한다. | String test = null;<br>System.out.println(StringUtil.nvl(test,<br>"NULL TEST"))="NULL TEST"     | since<br>core<br>1.5.0  |
| <b>toLowerCaseFirstLetter</b><br>(String str)                  | 주어진 String의 첫번째 글자<br>를 소문자로 변환한다.  | StringUtil.toUpperCase("Abcd")="bbcd"   | since<br>core<br>1.5.0  |
| <b>toUpperCaseFirstLetter</b><br>(String str)                  | 주어진 String의 첫번째 글자<br>를 대문자로 변환한다.  | StringUtil.toUpperCase("abcd")="Abcd"   | since<br>core<br>1.5.0  |
| <b>arrayToDelimitedString</b><br>(Object[] objects)            | Object[]를 입력받아 각<br>element를 ","로 구분하여<br>String을 생성한다.                         | String[] test = {"aaa", "bbb", "ccc"};<br>StringUtil.arrayToDelimitedString(test)="aaa,bbb,ccc" | since<br>core<br>1.5.0  |

| Method  | Description  | Example  | Remarks                |
|---|--|--|------------------------|
| <b>collectionToDelimitedString</b><br>(Collection<String> collection) | String을 입력으로 받아<br>, "(delimiter)로 각 element를<br>연결하여 String을 생성한다.      | String[] test = {"aaa", "bbb"};<br>StringUtil.collectionToCommaDelimitedString(Arrays.asList(test), ",")=          | since<br>core<br>1.5.0 |
| <b>containsAny</b><br>(String str, char[] chars)                      | char[]에 존재하는 문자중에<br>하나라도 String에 포함되어<br>있는지 검사한다.                      | StringUtil.containsAny("Anyframe", new<br>char[]{'a', 'x'}) = true   | since<br>core<br>1.5.0 |
| <b>containsAny</b><br>(String str, String chars)                      | chars에 존재하는 문자중에<br>하나라도 str에 존재하는지 여<br>부를검사한다.                         | StringUtil.containsAny("Anyframe",<br>"ax") = true   | since<br>core<br>1.5.0 |
| <b>deleteAny</b> (String str,<br>String charsToDelete)                | 하나의 String 객체 안에서 삭<br>제할 문자열 안에 포함된 모든<br>character들을 제거하여 반환<br>한다.    | StringUtil.deleteAny("AbzzzB", "bz") ==<br>"AB"  | since<br>core<br>1.5.0 |
| <b>deleteAny</b> (String str,<br>char[] charsToDelete)                | 하나의 String 객체 안에서 삭<br>제할 char[] 안에 포함된 모든<br>character들을 제거하여 반환<br>한다. | StringUtil.deleteAny("AbzzzB", new<br>char[]{'b', 'z'}) == "AB"  | since<br>core<br>1.5.0 |
| <b>deleteAny</b> (String str,<br>char charToDelete)                   | 하나의 String 객체 안에서 입<br>력된 문자를 모두 제거하여 반<br>환한다.                          | StringUtil.deleteAny("ABBBBBC", 'B')<br>=> "AC"  | since<br>core<br>1.5.0 |
| <b>deleteMatches</b><br>(String str, String subStr)                   | 하나의 문자열에서 입력된 문<br>자열을 모두 제거하여 반환한<br>다.                                 | StringUtil.deletePattern("zzABCcc",<br>"ABC") = "zzcc"   | since<br>core<br>1.5.0 |
| <b>deleteFirstMatches</b><br>(String, String)                         | 하나의 문자열에서 입력된 문<br>자열 중 첫번째를 제거하여 반<br>환한다.                              | StringUtil.deleteFirstMatches("aaaaxxxx",<br>"xx") = "aaaaxccc"  | since<br>core<br>1.5.0 |
| <b>abbreviateFromLeft</b><br>(String str, int len)                    | 주어진 String 객체에대해 왼<br>쪽에서부터 len만큼 길이를 얻<br>어내고 문자열 끝에 ...를 붙여<br>서 반환한다. | StringUtil.abbreviateFromLeft("anyframe",<br>8) = "anyframe..."  | since<br>core<br>1.5.0 |
| <b>abbreviateFromRight</b><br>(String str, int len)                   | 주어진 String 객체에대해 왼<br>쪽에서부터 len만큼 길이를 얻<br>어내고 문자열 앞에...를 붙여<br>서 반환한다.  | StringUtil.abbreviateFromRight("anyframe",<br>4) = "...java"   | since<br>core<br>1.5.0 |
| <b>tokenizeToStringArray</b><br>(String str, char<br>delimiter)       | 문자열을 입력받은 구분자로<br>분리하여 String[]로 반환한다.                                   | StringUtil.tokenizeToStringArray("aaa,bbb",<br>,',')   | since<br>core<br>1.5.0 |
| <b>tokenizeToStringArray</b><br>(String str, String<br>delimiter)     | 문자열을 입력받은 구분자로<br>분리하여 String[]로 반환한다.                                   | StringUtil.tokenizeToStringArray("aaa.bbb",<br>".");   | since<br>core<br>1.5.0 |
| <b>tokenizeToStringArray</b><br>(String str)                          | 문자열을 ', '로 분리하여<br>String[]로 반환한다.                                       | StringUtil.tokenizeToStringArray("aaa,bbb",<br>,',')   | since<br>core<br>1.5.0 |
| <b>containsMaxOccurrences</b><br>(String str, String<br>maxSeqNumber) | Regex인자로 전달된 String에<br>maxSeqNumber만큼 반복하<br>는 문자가 있는지 검사한다.            | StringUtil.containsMaxOccurrences("my00000",<br>"3")=true,StringUtil.containsMaxOccurrences("55555",<br>"5")=false | since<br>core<br>1.5.0 |

| Method  | Description   | Example   | Remarks                |
|---|---|---|------------------------|
| <b>isWhiteSpaceOnly</b><br>(String str)                         | 주어진 String이 white space 만을 가지고 있는지를 검사한다                    | StringUtil.isWhiteSpaceOnly(" ")=true                     | since<br>core<br>1.5.0 |
| <b>repeat</b> (int size,<br>char ch)                            | 특정한 문자(char)와 일정한 길이 값을 입력으로 받아 해당 크기만큼 문자가 반복되는 문자열을 생성한다. | repeat(5, 'e') = "eeeee"                                  | since<br>core<br>1.5.0 |
| <b>htmlEscape</b> (String<br>input)                             | html 태그를 입력으로 받아 escape 문자열로 변경해준다.                         | StringUtil.htmlEscape("<html>") =<br>&lt;html&gt;         | since<br>core<br>1.5.0 |
| <b>htmlUnescape</b><br>(String input)                           | html에서 사용가능한 escape 문자열을 입력받아 html태그 형태의 문자열로 변환한다.         | StringUtil.htmlEscape("&lt;html&gt;") =<br>"<html>"       | since<br>core<br>1.5.0 |
| <b>isRegexPatternMatch</b><br>(String str, String<br>pattern)   | 주어진 문자열이 입력된 정규식 패턴에 맞는지 체크한다.                              | StringUtil.isRegexPatternInclude("cabbba",<br>"a*b")=true | since<br>core<br>1.5.0 |
| <b>isUserFormat</b><br>(String str, String<br>pattern)          | 사용자가 지정한 포맷에 맞는 글자가 들어왔는지 체크한다.<br>#은 숫자를 S는 문자를 표현한다.      | ValidationUtil.isUserFormat("123,456",<br>"###,###")=true | since<br>core<br>1.5.0 |
| <b>isPatternInclude</b><br>(String str, String<br>param)        | 입력된 문자열이 주어진 필터 패턴에 맞는 문자열인지 확인                             | StringUtil.isPatternInclude("asdf@5456",<br>"s")=true     | since<br>core<br>1.5.0 |
| <b>isRegexPatternInclude</b><br>(String str, String<br>pattern) | 주어진 문자열 중에 일부 문자열이 패턴에 맞는지 체크한다                             | StringUtil.isRegexPatternInclude("cabbba",<br>"a*b")=true | since<br>core<br>1.5.0 |

## 5.ValidationUtil의 활용

정규 표현식을 활용한 주민번호,법인번호,사업자,외국인 등록번호등의 유효성 체크와 전화,휴대전화, 이메일,카드번호 등의 포맷 유효성 체크 기능을 제공한다.

다음은 ValidationUtil이 가지고 있는 메소드에 대한 설명이다.

| Method  | Description                                    | Example   | Remarks  |
|---|--|---|--|
| <b>isResidentRegNumber</b> (String regno)               | 입력된 주민등록번호가 유효한 주민등록번호인지 검증한다.                 | ValidationUtil.isResidentRegNumber("871224-1237613")=true           |  |
| <b>isIncorpCertNumber</b> (String corpNumber)           | 입력된 법인등록번호가 유효한 법인등록번호인지 검증한다.                 | ValidationUtil.isIncorpCertNumber("110111-0398556")=true            |  |
| <b>isBizRegNumber</b> (String bizNumber)                | 입력된 사업자등록번호가 유효한 사업자등록번호인지 검증한다.               | ValidationUtil.isBizRegNumber("110-81-28774")=true                  |  |
| <b>isTelephoneNumber</b> (String telNumber)             | 입력된 전화번호가 유효한 전화번호인지 검증한다.                     | ValidationUtil.isTelephoneNumber("0505-123-1231")                   |  |
| <b>isCellphoneNumber</b> (String cellPhoneNumber)       | 입력된 핸드폰번호가 유효한 핸드폰번호인지 검증한다.                   | ValidationUtil.isCellphoneNumber("018-1231-0912")                   |  |
| <b>isEmailAddress</b> (String email)                    | 입력된 이메일주소가 유효한 이메일주소인지 검증한다.                   | ValidationUtil.isEmailAddress("anyframe@samsung.com")               |  |
| <b>isCardNumber</b> (String cardNumber)                 | 입력된 카드번호가 유효한 카드번호인지 검증한다.                     | ValidationUtil.isCardNumber("4009-1311-1234-4321")                  |  |
| <b>isRangeLength</b> (String str, int min, int max)     | 문자열의 길이가 최소, 최대 길이 사이에 존재하는지 체크                | ValidationUtil.isRangeLength("Anyframe Java Test", 10, 20)=true     |  |
| <b>isRangeByteLength</b> (String str, int min, int max) | 문자열의 길이가 byte 단위로 계산했을때 최소, 최대 길이 사이에 존재하는지 체크 | ValidationUtil.isRangeByteLength("Anyframe Java Test", 10, 20)=true |  |
| <b>isUserFormat</b> (String str, String pattern)        |  |   | <b>deprecated</b><br>since core 1.5.0<br><br>StringUtil.isUserFormat(String str, String pattern) 사용        |
| <b>isRegexPatternMatch</b> (String str, String pattern) |  |   | <b>deprecated</b><br>since core 1.5.0<br><br>StringUtil.isRegexPatternMatch(String str, String pattern) 사용 |

| Method  | Description | Example | Remarks   |
|---|-------------|---------|---|
| <b>isPatternMatching</b> (String str, String pattern)     |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>StringUtil.isPatternMatching(str, String pattern)<br>사용  |
| <b>isPatternInclude</b> (String str, String param)        |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>StringUtil.isPatternInclude(S str, String param)<br>사용   |
| <b>isRegexPatternInclude</b> (String str, String pattern) |             |         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>StringUtil.isRegexPatternIncl str, String pattern)<br>사용 |

## 6.ThreadLocalUtil의 활용

ThreadLocal을 Map과 유사한 인터페이스로 사용할 수 있는 기능을 제공한다.

다음은 ThreadLocal이 가지고 있는 메소드에 대한 설명이다.

| Method  | Description   | Example                                   | Remarks   |
|---|---|---|---|
| <b>getThreadLocal()</b>                                     | ThreadLocal에 저장된 Map객체를 반환한다.   | ThreadLocalUtil.getThreadLocal()          | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>getAll()<br>사용 |
| <b>getAll()</b>   | ThreadLocal에 저장된 Map객체를 반환한다.   | ThreadLocalUtil.getAll()                  | since<br>core<br>1.5.0  |
| <b>getAll(boolean isInheritable)</b>                        | ThreadLocal에 저장된 Map객체를 반환한다. isInheritable 값이 true일 경우, InheritableThreadLocal을 이용한다.            | ThreadLocalUtil.getAll(true)              | since<br>core<br>1.5.0  |
| <b>add(Object key, Object value)</b>                        | ThreadLocal에 해당 key와 value를 가진 항목을 추가한다.  | ThreadLocalUtil.add("key", "value")       |   |
| <b>add(Object key, Object value, boolean isInheritable)</b> | ThreadLocal에 해당 key와 value를 가진 항목을 추가한다. isInheritable 값이 true일 경우, InheritableThreadLocal을 이용한다. | ThreadLocalUtil.add("key", "value", true) | since<br>core<br>1.5.0  |
| <b>get(Object key)</b>                                      | 해당 키를 가진 항목의 값을 반환한다.   | ThreadLocalUtil.get("key")                |   |
| <b>get(Object key, boolean isInheritable)</b>               | 해당 키를 가진 항목의 값을 반환한다. isInheritable 값이 true일 경우, InheritableThreadLocal을 이용한다.                    | ThreadLocalUtil.get("key", true)          | since<br>core<br>1.5.0  |
| <b>isExist(Object key)</b>                                  | 해당 키를 가진 항목이 존재하는지 여부를 판단한다.  | ThreadLocalUtil.isExist("key")            |   |
| <b>isExist(Object key, boolean isInheritable)</b>           | 해당 키를 가진 항목이 존재하는지 여부를 판단한다. isInheritable 값이 true일 경우, InheritableThreadLocal을 이용한다.             | ThreadLocalUtil.isExist("key", true)      | since<br>core<br>1.5.0  |
| <b>clearSharedInfo()</b>                                    | ThreadLocal에 저장된 모든 정보를 제거한다.   | ThreadLocalUtil.clearSharedInfo()         | <b>deprecated</b><br>since<br>core<br>1.5.0<br><br>clear() 사용     |



| Method   | Description  | Example                                  | Remarks  |
|--|--|--|--|
| <b>clear()</b>                                     | ThreadLocal에 저장된 모든 정보를 제거한다. InheritableThreadLocal을 이용한 경우, 부모 Thread에서 clear() 함수를 호출해도 자식 Thread의 값은 지워지지 않는다. 그 반대의 상황도 마찬가지로 동작한다. | ThreadLocalUtil.clear()                  | since core 1.5.0   |
| <b>getThreadLocalKeys()</b>                        | ThreadLocal에 저장된 모든 항목의 키값을 배열로 반환한다.  | ThreadLocalUtil.getThreadLocalKeys()     |  |
| <b>getThreadLocalKeys(boolean isInheritable)</b>   | ThreadLocal에 저장된 모든 항목의 키값을 배열로 반환한다. isInheritable값이 true일 경우, InheritableThreadLocal을 이용한다.  | ThreadLocalUtil.getThreadLocalKeys(true) | since core 1.5.0   |
| <b>size()</b>                                      | ThreadLocal에 저장된 항목의 개수를 반환한다.   | ThreadLocalUtil.size()                   |  |
| <b>size(boolean isInheritable)</b>                 | ThreadLocal에 저장된 항목의 개수를 반환한다. isInheritable값이 true일 경우, InheritableThreadLocal을 이용한다.   | ThreadLocalUtil.size()                   | since core 1.5.0   |
| <b>getThreadLocalValues()</b>                      | ThreadLocal에 저장된 모든 항목의 값을 배열로 반환한다.   | ThreadLocalUtil.getThreadLocalValues()   |  |
| <b>getThreadLocalValues(boolean isInheritable)</b> | ThreadLocal에 저장된 모든 항목의 값을 배열로 반환한다. isInheritable값이 true일 경우, InheritableThreadLocal을 이용한다.   | ThreadLocalUtil.getThreadLocalValues()   | since core 1.5.0   |
| <b>toPrintString()</b>                             |  |  | <b>deprecated</b><br>since core 1.5.0                          |
| <b>printThreadLocal()</b>                          | ThreadLocal에 있는 항목을 콘솔로 출력한다.  | ThreadLocalUtil.printThreadLocal()       | <b>deprecated</b><br>since core 1.5.0<br>getValueToString() 사용 |
| <b>getValuesToString()</b>                         | ThreadLocal에 있는 항목을 String으로 반환한다.   | ThreadLocalUtil.getValuesToString()      | since core 1.5.0   |
| <b>getValuesToString(boolean isInheritable)</b>    | ThreadLocal에 있는 항목을 String으로 반환한다. isInheritable 값이 true일 경우, InheritableThreadLocal을 이용한다.  | ThreadLocalUtil.getValuesToString()      | since core 1.5.0   |



## InheritableThreadLocal 사용 시 유의 사항

InheritableThreadLocal [<http://download.java.net/jdk7/archive/b123/docs/api/java/lang/InheritableThreadLocal.html>]은 부모 Thread로부터 값을 제공받아 자식 Thread에서 그 값을 사용하기 위해 제공되는 클래스이다. 현재 Thread에서 새로운 Thread를 생성했을 때 InheritableThreadLocal 영역에 저장된 값을 새로 생성된 Thread로 복사해주는 것이다. Anyframe에서 제공하는 ThreadLocalUtil을 사용할 경우 내부적으로 Map객체를 ThreadLocal 영역에 생성해서 사용하도록 구현되어 있다. 이 경우 **InheritableThreadLocal**을 이용하여 자식 Thread에서 값을 변경하게 되면, 다른 Thread에도 영향을 미치게 된다. 이는 자식 Thread가 생성될 때, 부모 Thread의 InheritableThreadLocal에 저장되어 있는 Map객체의 주소값(call by reference)을 자식 Thread로 복사하기 때문이다.