Business Rule Engine

Detailed Listing of Actions

Uwe Geercken – last update: 04.04.16

Table of Contents

Overview

This document contains a detailed listing of the available actions for the Business Rule Engine „Jare". It lists the available data types, their combinations and possible optional or required parameters.

Actions are defined on the rulegroup level. When a rulegroup fails or passes then none, one or multiple actions can be executed. Actions are executed in the sequence that they have been defined. It is also possible to define actions that are execute when the rulegroup passes and actions when the rulegroup fails at the same time.

A condition must be specified when to execute the action – when the rulegroup fails or passes. Additionally a field - and it's type - must be specified that will be updated by the action.

Sometimes an action will require additional values to be passed to it. This can either be the value of another field or a defined fixed value.

The values selected in the action from "Field to retrieve data", "Parameter 1", "Parameter 2" and "Parameter 3" are passed to the action in this sequence. The "Field to retrieve data" is optional and indicates from which other field a value should be retrieved. It can be specified but must not be.

If for example the action requires two values (arguments) then either "Field to retrieve data" and "Parameter 1" have to be specified or alternatively "Parameter 1" and "Parameter 2" (and "Field to retrieve data" is left blank). Then in turn the "Field to update" will be updated with the resulting value.

Prerequisites

The default date format used is: yyyy-MM-dd. If not otherwise specified for String to Date conversions, it is assumed that the value is provided in this format. Date formats need to be according to the Java SimpleDateFormat Class format definitions.

All data types listed here are Java related data types.

Add Leading Spaces

Purpose:	Add leading spaces to a value until a given length of the value is reached
Java Class:	StringAction
Interface Name:	add leading spaces

Applicable to:

Value 1	**Value 2**
String	Integer
Note:	
Optional Value(s)	**Explanation**

Add Leading Zeros

Purpose:	Add leading zeros to a value until a given length of the value is reached
Java Class:	StringAction
Interface Name:	add leading zeros

Applicable to:

Value 1	**Value 2**
String	Integer
Note:	
Optional Value(s)	**Explanation**

Append Value

Purpose:	Append a given value to the end of another value
Java Class:	StringAction
Interface Name:	append

Applicable to:

Value 1	**Value 2**
String	String
Note:	
Optional Value(s)	**Explanation**
Value 3: String	Separator to be used between the values

Prepend Value

Purpose:	Prepend a given value to the start of another value
Java Class:	StringAction
Interface Name:	prepend

Applicable to:

Value 1	**Value 2**
String	String
Note:	
Optional Value(s)	**Explanation**
Value 3: String	Separator to be used between the values

Trim Value

Purpose:	Remove all leading and trailing space characters from a String
Java Class:	StringAction
Interface Name:	trim

Applicable to:

Value 1	**Value 2**
String	
Note:	
Optional Value(s)	**Explanation**

Concat Values

Purpose:	Concatenate two values
Java Class:	StringAction
Interface Name:	append value

Applicable to:

Value 1	**Value 2**
String	Double
String	Float
String	Long
String	Integer
String	String
Note:	
Optional Value(s)	**Explanation**
Value 3: String	Separator to be used between the value

Lowercase Value

Purpose:	Change a value to it's lowercase representation
Java Class:	StringAction
Interface Name:	lowercase

Applicable to:

Value 1	**Value 2**
String	
Note:	
Optional Value(s)	**Explanation**

Uppercase Value

Purpose:	Change a value to it's uppercase representation
Java Class:	StringAction
Interface Name:	uppercase

Applicable to:

Value 1	**Value 2**
String	
Note:	
Optional Value(s)	**Explanation**

Replace Value

Purpose:	Replace a given value – or part of it - with another value
Java Class:	StringAction
Interface Name:	replace

Applicable to:

Value 1	**Value 2**
String	String
Note: Value 2 has to contain a regular expression. All found occurrences of the regular expression will be replaced with value 3.	
Required Value(s)	**Explanation**
Value 3: String	the replacement for the original value or parts of it

Set Value (String)

Purpose:	Sets a given value of a string
Java Class:	StringAction
Interface Name:	set value (string)

Applicable to:

Value 1	**Value 2**
String	
Note:	
Optional Value(s)	**Explanation**

Set Value (Date)

Purpose:	Sets a given value of a date
Java Class:	DateAction
Interface Name:	set value (date)

Applicable to:

Value 1	**Value 2**
String	
Note:	
Optional Value(s)	**Explanation**

Set Value (Number)

Purpose:	Sets a given value of a number
Java Class:	MathAction
Interface Name:	absolute

Applicable to:

Value 1	**Value 2**
Double	
Float	
Long	
Integer	
Boolean	
Note:	
Optional Value(s)	**Explanation**

Substring Value

Purpose:	Sets the value to a given substring
Java Class:	StringAction
Interface Name:	set value (String)

Applicable to:

Value 1	**Value 2**
String	Integer
Note: Value 2 defines from which position of the value the substring will start	
Optional Value(s)	**Explanation**
Value 3: Integer	Defines at which position the substring will end

Set Todays Date

Purpose:	Set the value to the current date
Java Class:	DateAction
Interface Name:	

Applicable to:

Value 1	**Value 2**
String	
Note: Value 1 contains the date format to use	
Optional Value(s)	**Explanation**
Value 2: Integer	Define an offset of days – positive or negative – that will be added to the current date

Absolute Value

Purpose:	Calculates the absolute value of a given value
Java Class:	MathAction
Interface Name:	absolute

Applicable to:

Value 1	**Value 2**
Double	
Float	
Long	
Integer	
Note:	
Optional Value(s)	**Explanation**

Acosinus Value

Purpose:	Calculates the acosinus of a given value
Java Class:	MathAction
Interface Name:	acosinus

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Cosinus Value

Purpose:	Calculates the cosinus of a given value
Java Class:	MathAction
Interface Name:	cosinus

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Cosinush Value

Purpose:	Calculates the cosinush of a given value
Java Class:	MathAction
Interface Name:	cosinush

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Asinus Value

Purpose:	Calculates the asinus of a given value
Java Class:	MathAction
Interface Name:	asinus

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Sinus Value

Purpose:	Calculates the sinus of a given value
Java Class:	MathAction
Interface Name:	sinus

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Sinush Value

Purpose:	Calculates the sinush of a given value
Java Class:	MathAction
Interface Name:	sinush

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Atangens Value

Purpose:	Calculates the atangens of a given value
Java Class:	MathAction
Interface Name:	atangens

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Tangens Value

Purpose:	Calculates the tangens of a given value
Java Class:	MathAction
Interface Name:	tangens

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Tangensh Value

Purpose:	Calculates the tangensh of a given value
Java Class:	MathAction
Interface Name:	tangensh

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Square Root Value

Purpose:	Calculates the square root of a given value
Java Class:	MathAction
Interface Name:	square root

Applicable to:

Value 1	**Value 2**
Double	
Note:	
Optional Value(s)	**Explanation**

Square Value

Purpose:	Calculates the square of a given value
Java Class:	MathAction
Interface Name:	square

Applicable to:

Value 1	**Value 2**
Double	
Float	
Long	
Integer	
Note:	
Optional Value(s)	**Explanation**

Add Values

Purpose:	Calculates the sum of two values
Java Class:	MathAction
Interface Name:	sum

Applicable to:

Value 1	**Value 2**
Double	Double
Float	Float
Integer	Integer
Integer	Long
Long	Integer
Long	Long
Note:	
Optional Value(s)	**Explanation**

Subtract Values

Purpose:	Subtracts two values
Java Class:	MathAction
Interface Name:	subtract

Applicable to:

Value 1	**Value 2**
Double	Double
Float	Float
Integer	Integer
Integer	Long
Long	Integer
Long	Long
Note:	
Optional Value(s)	**Explanation**

Devide Values

Purpose:	Calculates the devision of two values
Java Class:	MathAction
Interface Name:	devide

Applicable to:

Value 1	**Value 2**
BigDecimal	BigDecimal
Double	Double
Double	Integer
Float	Float
Integer	Integer
Integer	Long
Long	Integer
Long	Double
Long	Long
Note:	
Optional Value(s)	**Explanation**

Multiply Values

Purpose:	Multiplies two values
Java Class:	MathAction
Interface Name:	multiply

Applicable to:

Value 1	**Value 2**
Double	Double
Double	Integer
Float	Float
Integer	Integer
Integer	Long
Long	Double
Long	Integer
Long	Long
Note:	
Optional Value(s)	**Explanation**

Round Value

Purpose:	Rounds a given value
Java Class:	MathAction
Interface Name:	round

Applicable to:

Value 1	**Value 2**
Double	
Float	
Note:	
Optional Value(s)	**Explanation**