

# **Business Rule Engine**

## **Detailed Listing of Checks**

## Table of Contents

Overview.....	4
Prerequisites.....	5
Check Is Equal.....	6
Check Is Not Equal.....	7
Check Contains.....	8
Check Not Contains.....	9
Check Is Between.....	10
Check Is Not Between.....	13
Check Is Empty.....	16
Check Is Not Empty.....	17
Check Is Even.....	18
Check Is Not Even.....	19
Check Is Greater.....	20
Check Is Greater or Equal.....	22
Check Is Smaller.....	24
Check Is Smaller or Equal.....	26
Check Is In List.....	28
Check Is Not In List.....	29
Check Is Lowercase.....	30
Check Is Uppercase.....	31
Check Matches.....	32
Check Not Matches.....	33
Check Sounds Like.....	34
Check Not Sounds Like.....	35
Check Is Negative Number.....	36
Check Is Not Negative Number.....	37
Check Length.....	38
Check List Has Member.....	39
Check List Not Has Member.....	40
Check Starts With.....	41
Check Not Starts With.....	42
Check Ends With.....	43
Check Not Ends With.....	44
Check Is Null.....	45
Check Is Not Null.....	46
Check Is Numeric.....	47
Check Is Prime.....	48
Check Is Not Prime.....	49

Check Levenshtein Distance is equal.....	50
Check Levenshtein Distance is greater.....	51
Check Levenshtein Distance is greater or equal.....	52
Check Levenshtein Distance is smaller.....	53
Check Levenshtein Distance is smaller or equal.....	54

## Overview

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

In most cases two values are compared against each other. But in some cases there are additional parameters available, that can be specified (optionally) or that must be specified (required).

There are checks that are logical checks and only expect a single value. An example is the CheckIsUppercase or CheckIsPrime – they don't compare values, but make a logical evaluation.

The "compare" column is always the value that comes from the source data. It will always be passed first to the relevant Java method.

The "compare to" column is the value to check against. It can be another value coming from the source data or it can be a fixed, defined 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 format definitions.

## Check Is Equal

Purpose:	Check for equality of two values
Java Class:	CheckIsEqual
Interface Name:	is equal to

Possible comparisons:

Compare	Compare to
String	String
Boolean	Boolean
Date	Date
Date	String
Double	Double
Float	Float
Integer	Integer
Integer	Long
Long	Integer
Long	Long

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Optional additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

Compare	Compare to
Date	String
Note: The „compare to“ String value will be converted internally to a date, using the default date format, if no different date format is specified.	
Optional additional Parameter	Explanation
Parameter 1: String	Provide a date format if different from the default.

## Check Is Not Equal

Purpose:	Check if two values are not equal
Java Class:	CheckIsNotEqual
Interface Name:	is not equal to

Possible comparisons:

Compare	Compare to
String	String
Boolean	Boolean
Date	Date
Date	String
Double	Double
Float	Float
Integer	Integer
Integer	Long
Long	Integer
Long	Long

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Optional additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

Compare	Compare to
Date	String
Note: The „compare to“ String value will be converted internally to a date, using the default date format, if no different date format is specified.	
Optional additional Parameter	Explanation
Parameter 1: String	Provide a date format if different from the default.

## Check Contains

Purpose:	Check if a value contains a given other value
Java Class:	CheckContains
Interface Name	contains

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Optional additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.



## Check Not Contains

Purpose:	Check if a value does not contain a given other value
Java Class:	CheckNotContains
Interface Name	not contains

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Optional additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

## Check Is Between

Purpose:	Check if a value is between two given other values
Java Class:	CheckIsBetween
Interface Name:	is between

Possible comparisons:

Compare	Compare to
Double	Double
Float	Float
Float	String
Integer	Integer
Integer	String
Long	Long
Long	String

Special Notes:

Compare	Compare to
Double	Double
Note: The „compare to“ value defines the lower limit to check against.	
Required additional Parameter	Explanation
Parameter 1: Double	Specify the upper limit to check against

Compare	Compare to
Float	Float
Note: The „compare to“ value defines the lower limit to check against.	
Required additional Parameter	Explanation
Parameter 1: Float	Specify the upper limit to check against

Compare	Compare to
Float	String
Note: The „compare to“ value defines the lower and the upper limit – separated by a comma - to check against.	
Optional additional Parameter	Explanation

Compare	Compare to
Integer	Integer
Note: The „compare to“ value defines the lower limit to check against.	
Required additional Parameter	Explanation
Parameter 1: Integer	Specify the upper limit to check against

Compare	Compare to
Integer	String
Note: The „compare to“ value defines the lower and the upper limit – separated by a comma - to check against.	
Optional additional Parameter	Explanation

Compare	Compare to
Long	Long
Note: The „compare to“ value defines the lower limit to check against.	
Required additional Parameter	Explanation
Parameter 1: Long	Specify the upper limit to check against

Compare	Compare to
Long	String
Note: The „compare to“ value defines the lower and the upper limit – separated by a comma - to check against.	
Optional additional Parameter	Explanation

## Check Is Not Between

Purpose:	Check if a value is not between two given other values
Java Class:	CheckIsNotBetween
Interface Name:	is not between

Possible comparisons:

Compare	Compare to
Double	Double
Float	Float
Float	String
Integer	Integer
Integer	String
Long	Long
Long	String

Special Notes:

Compare	Compare to
Double	Double
Note: The „compare to“ value defines the lower limit to check against.	
Required additional Parameter	Explanation
Parameter 1: Double	Specify the upper limit to check against

Compare	Compare to
Float	Float
Note: The „compare to“ value defines the lower limit to check against.	
Required additional Parameter	Explanation
Parameter 1: Float	Specify the upper limit to check against

Compare	Compare to
Float	String
Note: The „compare to“ value defines the lower and the upper limit – separated by a comma - to check against.	
Optional additional Parameter	Explanation

Compare	Compare to
Integer	Integer
Note: The „compare to“ value defines the lower limit to check against.	
Required additional Parameter	Explanation
Parameter 1: Integer	Specify the upper limit to check against

Compare	Compare to
Integer	String
Note: The „compare to“ value defines the lower and the upper limit – separated by a comma - to check against.	
Optional additional Parameter	Explanation

Compare	Compare to
Long	Long
Note: The „compare to“ value defines the lower limit to check against.	
Required additional Parameter	Explanation
Parameter 1: Long	Specify the upper limit to check against

Compare	Compare to
Long	String
Note: The „compare to“ value defines the lower and the upper limit – separated by a comma - to check against.	
Optional additional Parameter	Explanation

## Check Is Empty

Purpose:	Check if a value is not null but is of zero length/is empty
Java Class:	CheckIsEmpty
Interface Name:	is empty

Possible comparisons:

Compare	Compare to
String	



## Check Is Not Empty

Purpose:	Check if a value is not null but is not of zero length/is not empty
Java Class:	CheckIsNotEmpty
Interface Name:	is not empty

Possible comparisons:

Compare	Compare to
String	

## Check Is Even

Purpose:	Check if a value is even
Java Class:	CheckIsEven
Interface Name:	is even

Possible comparisons:

Compare	Compare to
Integer	
Long	

## Check Is Not Even

Purpose:	Check if a value is not even
Java Class:	CheckIsNotEven
Interface Name:	is not even

Possible comparisons:

Compare	Compare to
Integer	
Long	

## Check Is Greater

Purpose:	Check if a value is greater than the other one
Java Class:	CheckIsGreater
Interface Name:	is greater than

Possible comparisons:

Compare	Compare to
Date	Date
Date	String
Double	Double
Double	Integer
Double	Long
Float	Float
Float	Integer
Integer	Integer
Long	Integer
Long	Long
String	Integer
String	Long
String	String

Special Notes:

Compare	Compare to
Date	String
Note: The „compare to“ value is a String format in the default date format (yyyy-MM-dd)	
Optional additional Parameter	Explanation
Parameter 1: String	Provide a date format if different from the default.

Compare	Compare to
String	Integer
Note: Checks if the string is greater than the given value in „compare to“.	
Optional additional Parameter	Explanation

Compare	Compare to
String	Long
Note: Checks if the string is greater than the given value in „compare to“.	
Optional additional Parameter	Explanation

Compare	Compare to
String	String
Note: Checks if the string is greater than the given value in „compare to“.	
Required additional Parameter	Explanation
Parameter 1: String	Provide a date format

## Check Is Greater or Equal

Purpose:	Check if a value is greater than or equal to the other one
Java Class:	CheckIsGreaterOrEqual
Interface Name:	is greater or equal than

Possible comparisons:

Compare	Compare to
Date	Date
Date	String
Double	Double
Double	Integer
Double	Long
Float	Float
Float	Integer
Integer	Integer
Long	Integer
Long	Long
String	Integer
String	Long
String	String

Special Notes:

Compare	Compare to
Date	String
Note: The „compare to“ value is a String format in the default date format (yyyy-MM-dd)	
Optional additional Parameter	Explanation
Parameter 1: String	Provide a date format if different from the default.

Compare	Compare to
String	Integer
Note: Checks if the string is greater than or equal to the given value in „compare to“.	
Optional additional Parameter	Explanation

Compare	Compare to
String	Long
Note: Checks if the string is greater than the given value in „compare to“.	
Optional additional Parameter	Explanation

Compare	Compare to
String	String
Note: Checks if the string is greater than or equal to the given value in „compare to“.	
Required additional Parameter	Explanation
Parameter 1: String	Provide a date format

## Check Is Smaller

Purpose:	Check if a value is smaller than the other one
Java Class:	CheckIsSmaller
Interface Name:	is smaller than

Possible comparisons:

Compare	Compare to
Date	Date
Date	String
Double	Double
Double	Integer
Double	Long
Float	Float
Float	Integer
Integer	Integer
Long	Integer
Long	Long
String	Integer
String	Long
String	String

Special Notes:

Compare	Compare to
Date	String
Note: The „compare to“ value is a String format in the default date format (yyyy-MM-dd)	
Optional additional Parameter	Explanation
Parameter 1: String	Provide a date format if different from the default.



Compare	Compare to
String	Integer
Note: Checks if the string is greater than the given value in „compare to“.	
Optional additional Parameter	Explanation

Compare	Compare to
String	Long
Note: Checks if the string is greater than the given value in „compare to“.	
Optional additional Parameter	Explanation

Compare	Compare to
String	String
Note: Checks if the string is greater than the given value in „compare to“.	
Required additional Parameter	Explanation
Parameter 1: String	Provide a date format

## Check Is Smaller or Equal

Purpose:	Check if a value is smaller than or equal to the other one
Java Class:	CheckIsSmallerOrEqual
Interface Name:	is smaller or equal than

Possible comparisons:

Compare	Compare to
Date	Date
Date	String
Double	Double
Double	Integer
Double	Long
Float	Float
Float	Integer
Integer	Integer
Long	Integer
Long	Long
String	Integer
String	Long
String	String

Special Notes:

Compare	Compare to
Date	String
Note: The „compare to“ value is a String format in the default date format (yyyy-MM-dd)	
Optional additional Parameter	Explanation
Parameter 1: String	Provide a date format if different from the default.

Compare	Compare to
String	Integer
Note: Checks if the string is greater than or equal to the given value in „compare to“.	
Optional additional Parameter	Explanation

Compare	Compare to
String	Long
Note: Checks if the string is greater than the given value in „compare to“.	
Optional additional Parameter	Explanation

Compare	Compare to
String	String
Note: Checks if the string is greater than or equal to the given value in „compare to“.	
Required additional Parameter	Explanation
Parameter 1: String	Provide a date format

## Check Is In List

Purpose:	Check if a value is in a given list of values
Java Class:	CheckIsInList
Interface Name:	is in list

Possible comparisons:

Compare	Compare to
Integer	String
Long	String
String	String

Special Notes:

Compare	Compare to
String	String
Note:	
Required additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

## Check Is Not In List

Purpose:	Check if a value is not in a given list of values
Java Class:	CheckIsNotInList
Interface Name:	is not in list

Possible comparisons:

Compare	Compare to
Integer	String
Long	String
String	String

Special Notes:

Compare	Compare to
String	String
Note:	
Required additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

## Check Is Lowercase

Purpose:	Check if a value contains lowercase characters only
Java Class:	CheckIsLowercase
Interface Name:	is lowercase

Possible comparisons:

Compare	Compare to
String	

## Check Is Uppercase

Purpose:	Check if a value contains uppercase characters only
Java Class:	CheckIsUppercase
Interface Name:	is uppercase

Possible comparisons:

Compare	Compare to
String	

## Check Matches

Purpose:	Check if a value matches a given regular expression
Java Class:	CheckMatches
Interface Name:	matches

Possible comparisons:

Compare	Compare to
String	String



## Check Not Matches

Purpose:	Check if a value does not match a given regular expression
Java Class:	CheckNotMatches
Interface Name:	not matches

Possible comparisons:

Compare	Compare to
String	String

## Check Sounds Like

Purpose:	Check if a value sounds like the other using the Soundex algorithm
Java Class:	CheckSoundsLike
Interface Name:	sounds like

Possible comparisons:

Compare	Compare to
String	String

## Check Not Sounds Like

Purpose:	Check if a value does not sound like the other using the Soundex algorithm
Java Class:	CheckNotSoundsLike
Interface Name:	not sounds like

Possible comparisons:

Compare	Compare to
String	String

## Check Is Negative Number

Purpose:	Check if a value is a negative number
Java Class:	CheckIsNegativeNumber
Interface Name:	is negative number

Possible comparisons:

Compare	Compare to
Double	
Integer	
Float	
Long	

## Check Is Not Negative Number

Purpose:	Check if a value is not a negative number
Java Class:	CheckIsNotNegativeNumber
Interface Name:	is not negative number

Possible comparisons:

Compare	Compare to
Double	
Integer	
Float	
Long	

## Check Length

Purpose:	Checks if a value contains a given number of digits/characters
Java Class:	CheckLength
Interface Name:	has length

Possible comparisons:

Compare	Compare to
Integer	Integer
Integer	Long
Long	Integer
String	Integer
String	Long

## Check List Has Member

Purpose:	Checks if a list of values contains a given member.
Java Class:	CheckListHasMember
Interface Name:	has list member

Possible comparisons:

Compare	Compare to
String	Integer
String	Long
String	String
String	Integer
String	Long

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Required additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

## Check List Not Has Member

Purpose:	Checks if a list of values does not contain a given member.
Java Class:	CheckListNotHasMember
Interface Name:	has not list member

Possible comparisons:

Compare	Compare to
String	Integer
String	Long
String	String
String	Integer
String	Long

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Required additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.



## Check Starts With

Purpose:	Checks if a value starts with a given other value
Java Class:	CheckStartsWith
Interface Name:	starts with

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Required additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

## Check Not Starts With

Purpose:	Checks if a value does not start with a given other value
Java Class:	CheckNotStartsWith
Interface Name:	not starts with

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Required additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

## Check Ends With

Purpose:	Check if a value ends with a given other value
Java Class:	CheckEndsWith
Interface Name:	ends with

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Optional additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

## Check Not Ends With

Purpose:	Check if a value does not end with a given other value
Java Class:	CheckNotEndsWith
Interface Name:	not ends with

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note: The default is to compare the values case sensitive.	
Optional additional Parameter	Explanation
Parameter 1: Boolean	Ignore case differences during comparison.

## Check Is Null

Purpose:	Check if a value is Null
Java Class:	CheckIsNull
Interface Name:	is null

Possible comparisons:

Compare	Compare to
Date	
Double	
Float	
Integer	
Long	
String	

Special Notes:

Compare	Compare to
String	
Note: String will be checked for being null. Additionally a check is made if the value of the String is equal to the String „null“	
Optional additional Parameter	Explanation

## Check Is Not Null

Purpose:	Check if a value is not Null
Java Class:	CheckIsNotNull
Interface Name:	is not null

Possible comparisons:

Compare	Compare to
Date	
Double	
Float	
Integer	
Long	
String	

Special Notes:

Compare	Compare to
String	
<p>Note:</p> <p>String will be checked for not being null. Additionally a check is made that the value of the String is not equal to the String „null“</p>	
Optional additional Parameter	Explanation

## Check Is Numeric

Purpose:	Check if all characters of a value are numeric
Java Class:	CheckIsNumeric
Interface Name:	is numeric

Possible comparisons:

Compare	Compare to
String	

## Check Is Prime

Purpose:	Check if a value is a prime number
Java Class:	CheckIsPrime
Interface Name:	is prime

Possible comparisons:

Compare	Compare to
Integer	



## Check Is Not Prime

Purpose:	Check if a value is not a prime number
Java Class:	CheckIsNotPrime
Interface Name:	not is prime

Possible comparisons:

Compare	Compare to
Integer	

## Check Levenshtein Distance is equal

Purpose:	Check if the Levenshtein Distance is equal to a given value
Java Class:	CheckDistancelsEqual
Interface Name:	is equal to (Levenshtein distance)

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note:	
Required additional Parameter	Explanation
Parameter 1: integer	The distance value to be checked against

## Check Levenshtein Distance is greater

Purpose:	Check if the Levenshtein Distance is greater than a given value
Java Class:	CheckDistancelsGreater
Interface Name:	is greater than (Levenshtein distance)

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note:	
Required additional Parameter	Explanation
Parameter 1: integer	The distance value to be checked against

## Check Levenshtein Distance is greater or equal

Purpose:	Check if the Levenshtein Distance is greater than or equal to a given value
Java Class:	CheckDistancelIsGreaterOrEqual
Interface Name:	is greater than or equal to (Levenshtein distance)

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note:	
Required additional Parameter	Explanation
Parameter 1: integer	The distance value to be checked against

## Check Levenshtein Distance is smaller

Purpose:	Check if the Levenshtein Distance is smaller than a given value
Java Class:	CheckDistancelIsSmaller
Interface Name:	is smaller than (Levenshtein distance)

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note:	
Required additional Parameter	Explanation
Parameter 1: integer	The distance value to be checked against

## Check Levenshtein Distance is smaller or equal

Purpose:	Check if the Levenshtein Distance is smaller than or equal to a given value
Java Class:	CheckDistanceIsSmallerOrEqual
Interface Name:	is smaller than or equal to (Levenshtein distance)

Possible comparisons:

Compare	Compare to
String	String

Special Notes:

Compare	Compare to
String	String
Note:	
Required additional Parameter	Explanation
Parameter 1: integer	The distance value to be checked against