

Package ‘rGroovy’

July 23, 2025

Type Package

Title Groovy Language Integration

Version 1.3

Date 2018-04-01

Author Thomas P. Fuller <thomas.fuller@coherentlogic.com>

Maintainer Thomas P. Fuller <thomas.fuller@coherentlogic.com>

Description Integrates the Groovy scripting language with the R Project for Statistical Computing.

SystemRequirements Java (>= 7)

Depends rJava

Suggests testthat

LazyLoad yes

License LGPL-3

Copyright (C) 2018 Coherent Logic Limited; ALL RIGHTS RESERVED.

URL <http://groovy-lang.org/>, <http://www.groovy-lang.org/indy.html>,
<http://docs.groovy-lang.org/latest/html/api/index.html>,
<https://coherentlogic.com/wordpress/middleware-development/rgroovy?source=cran>

Repository CRAN

Archs i386, x64

RoxygenNote 6.0.1

NeedsCompilation no

Date/Publication 2018-04-01 12:29:43 UTC

Contents

About	2
CheckJRERuntimeVersion	2
Evaluate	2
Execute	3
Initialize	4
rGroovy	5

Index**6**

About*Function prints some information about this package.*

Description

Function prints some information about this package.

Usage

About()

CheckJRERuntimeVersion*Function verifies the existing JRE version is greater than or equal to 1.7 and, if it is not, invokes the stop function with a message.*

Description

Function verifies the existing JRE version is greater than or equal to 1.7 and, if it is not, invokes the stop function with a message.

Usage

CheckJRERuntimeVersion(runtime_version)

Arguments

runtime_version

For example, "1.7".

Evaluate*Function evaluates (executes) the groovy script and returns the result.*

Description

Function evaluates (executes) the groovy script and returns the result.

Usage

Evaluate(groovyShell = NULL, groovyScript)

Arguments

- `groovyShell` The `groovyShell` with which to execute the specified groovy script. Note that the `groovyShell` can be `NULL`, however if this is `NULL` then the `Initialize` function must have been called so that a global `groovyShell` instance will be available in the environment otherwise an exception is raised.
- `groovyScript` The groovy script being executed.

Value

The result of the script execution.

Examples

```
## Not run:
Initialize ()
Evaluate (groovyScript="print 'Hello world!'")

## End(Not run)
```

Execute

Function executes the groovy script and returns the result. Execute differs from Evaluate in that references to Groovy objects are not required. The call to Initialize is not required in order to call this function either however keep in mind that a new instance of <http://docs.groovy-lang.org/latest/html/api/groovy/lang/GroovyShell.html> will be used every time this function is called.

Description

Function executes the groovy script and returns the result. Execute differs from Evaluate in that references to Groovy objects are not required. The call to Initialize is not required in order to call this function either however keep in mind that a new instance of `groovy.lang.GroovyShell` will be used every time this function is called.

Usage

```
Execute(groovyScript, variables = list())
```

Arguments

- `groovyScript` The groovy script being executed.
- `variables` The variables that will be passed to the binding that is used when the `groovyScript` is executed.

Examples

```
## Not run:
variables <- list ()

variables["name"] = "Tom"
variables["day"]  = "Wednesday"

groovyScript <- "return \"Hello ${name}, how are you doing? Today is ${day}.\""

result <- Execute (groovyScript=groovyScript, variables=variables)
result

## End(Not run)
```

Initialize

Function sets the global instance of GroovyShell that will be used by the Evaluate function whenever it is called with a NULL GroovyShell parameter.

Description

Function sets the global instance of GroovyShell that will be used by the Evaluate function whenever it is called with a NULL GroovyShell parameter.

Usage

```
Initialize(binding = NULL)
```

Arguments

binding An instance of `groovy.lang.Binding`.

Examples

```
## Not run:
Initialize ()

## End(Not run)
```

Description

Functions that integrate the Groovy scripting language with the R Project for Statistical Computing.

Details

From [Wikipedia](#):

"Groovy is an object-oriented programming language for the Java platform. It is a dynamic language with features similar to those of Python, Ruby, Perl, and Smalltalk. It can be used as a scripting language for the Java Platform, is dynamically compiled to Java Virtual Machine (JVM) bytecode, and interoperates with other Java code and libraries."

One powerful feature this package delivers is that it allows the developer to enhance their R script with Java and Groovy code without necessarily being required to ship jars (see Grape, below). A simple example is included here and advanced examples can be found at the project's homepage.

See Also

[rGroovy](#)

[Groovy \(programming language\)](#)

[Groovy](#)

[Grape](#)

[Invoke Dynamic](#)

Examples

```
## Not run:
#
# Installation Example
#
# Since this package does not ship with Groovy the user needs to specify the Groovy jars prior
# to using the package -- here's an example how this is accomplished:
#
groovyJars <- list (
  "C:/Temp/groovy.jars/groovy-2.4.5-indy.jar",
  # OTHER JAR FILES...
)

options(GROOVY_JARS=groovyJars)

library(rGroovy)

Execute (groovyScript="print 'Hello world!'")

## End(Not run)
```

Index

About, [2](#)

CheckJRERuntimeVersion, [2](#)

Evaluate, [2](#)

Execute, [3](#)

Initialize, [4](#)

rGroovy, [5](#)

rGroovy-package (rGroovy), [5](#)