

# Package ‘OhdsiRTools’

October 14, 2015

**Type** Package

**Title** Tools for Maintaining OHDSI R Packages

**Version** 0.0.2

**Date** 2015-06-23

**Author** Martijn J. Schuemie [aut, cre],  
Marc A. Suchard [aut],

**Maintainer** Martijn J. Schuemie <schuemie@ohdsi.org>

**Description**

Format and check syntax of R code and packages following the OHDSI R style guidelines.

**License** Apache License 2.0

**Depends** R (>= 3.1.0)

**Imports** lintr,  
codetools,  
formatR,  
snow,  
ff,  
RJSONIO,  
RCurl,  
XML

**Suggests** testthat

**NeedsCompilation** no

## R topics documented:

checkUsagePackage . . . . .	2
clusterApply . . . . .	3
clusterRequire . . . . .	3
convertArgsToList . . . . .	4
convertFakeDfToDataFrame . . . . .	4
createArgFunction . . . . .	5
excludeFromList . . . . .	5
fakeDfToFdf . . . . .	6
formatRFile . . . . .	6
formatRFolder . . . . .	7
formatRText . . . . .	7
insertCirceDefinitionInPackage . . . . .	8

makeCluster . . . . .	8
matchInList . . . . .	9
nrowOfFakeDf . . . . .	9
ohdsiLintrFile . . . . .	10
ohdsiLintrFolder . . . . .	10
OhdsiRTools . . . . .	10
prettyPrint . . . . .	11
rbindFakeDfs . . . . .	11
selectFromList . . . . .	12
stopCluster . . . . .	12
subsetFakeDf . . . . .	12
subsetFfdFToFakeDf . . . . .	13

<b>Index</b>	<b>14</b>
--------------	-----------

---

checkUsagePackage	<i>Check all code in a package</i>
-------------------	------------------------------------

---

## Description

Check all code in a package

## Usage

```
checkUsagePackage(package, ignoreHiddenFunctions = TRUE,
  suppressBindingKeywords = c("ggplot2", "ffwhich", "subset.ffdf", "glm"))
```

## Arguments

package	The name of the package to check.
ignoreHiddenFunctions	Ignore functions for which the definition cannot be retrieved?
suppressBindingKeywords	A set of keywords that are indicative of non-standard evaluation.

## Details

This function uses the codetools package to check the code from problems. Heuristics are used to eliminate false positives due to non-standard evaluation.

---

clusterApply	<i>Apply a function to a list using the cluster</i>
--------------	---

---

**Description**

Apply a function to a list using the cluster

**Usage**

```
clusterApply(cluster, x, fun, ..., stopOnError = FALSE, progressBar = TRUE,  
             divideFfMemory = TRUE, setFfTempDir = TRUE)
```

**Arguments**

cluster	The cluster of threads to run the function.
x	The list on which the function will be applied.
fun	The function to apply.
...	Additional parameters for the function.
stopOnError	Stop when one of the threads reports an error? If FALSE, all errors will be reported at the end.
progressBar	Show a progress bar?
divideFfMemory	When TRUE, the memory available for processing ff and ffd objects will be equally divided over the threads.
setFfTempDir	When TRUE, the ffTempDir option will be copied to each thread.

**Details**

The function will be executed on each element of x in the threads of the cluster. If there are more elements than threads, the elements will be queued. The progress bar will show the number of elements that have been completed.

**Value**

A list with the result of the function on each item in x.

---

clusterRequire	<i>Require a package in the cluster</i>
----------------	---

---

**Description**

Require a package in the cluster

**Usage**

```
clusterRequire(cluster, package)
```

**Arguments**

cluster	The cluster object.
package	The name of the package to load in all nodes.

---

convertArgsToList	<i>Convert arguments used in call to a list</i>
-------------------	---

---

**Description**

Convert arguments used in call to a list

**Usage**

```
convertArgsToList(matchCall, resultClass = "list")
```

**Arguments**

matchCall	The result of <code>match.call()</code> .
resultClass	The class of the resulting object.

**Details**

Takes the argument values (both default and user-specified) and store them in a list.

**Value**

An object of the class specified in `resultClass`.

**Examples**

```
myFun <- function(x = 1, y = 2) {  
  return(convertArgsToList(match.call()))  
}
```

---

convertFakeDfToDataFrame	<i>Convert a fakeDf object to a data.frame</i>
--------------------------	--

---

**Description**

Convert a fakeDf object to a data.frame

**Usage**

```
convertFakeDfToDataFrame(fakeDf)
```

**Arguments**

fakeDf	An object of type fakeDf.
--------	---------------------------

**Value**

A data.frame.

---

createArgFunction	<i>Create an argument function</i>
-------------------	------------------------------------

---

**Description**

Create an argument function

**Usage**

```
createArgFunction(functionName, excludeArgs = c(), includeArgs = NULL,  
  rCode = c(), newName)
```

**Arguments**

functionName	The name of the function for which we want to create an args function.
excludeArgs	Exclude these arguments from appearing in the args function.
includeArgs	Include these arguments in the args function.
rCode	A character vector representing the R code where the new function should be appended to.
newName	The name of the new function. If not specified, the new name will be automatically derived from the old name.

**Details**

This function can be used to create a function that has (almost) the same interface as the specified function, and the output of this function will be a list of argument values.

**Value**

A character vector with the R code including the new function.

---

excludeFromList	<i>Exclude variables from a list of objects of the same type</i>
-----------------	--

---

**Description**

Exclude variables from a list of objects of the same type

**Usage**

```
excludeFromList(x, exclude)
```

**Arguments**

x	A list of objects of the same type.
exclude	A character vector of names of variables to exclude.

---

fakeDfToFdf	<i>Convert a fakeDf object to fdf</i>
-------------	---------------------------------------

---

**Description**

Convert a fakeDf object to fdf

**Usage**

```
fakeDfToFdf(fakeDf)
```

**Arguments**

fakeDf	An object of type fakeDf.
--------	---------------------------

**Value**

An fdf object.

---

formatRFile	<i>Format an R file</i>
-------------	-------------------------

---

**Description**

Format an R file

**Usage**

```
formatRFile(file, width.cutoff = 100)
```

**Arguments**

file	The path to the file.
width.cutoff	Number of characters that each line should be limited to.

---

formatRFolder	<i>Format all R files in a folder</i>
---------------	---------------------------------------

---

**Description**

Format all R files in a folder

**Usage**

```
formatRFolder(path = ".", recursive = TRUE, skipAutogenerated = TRUE, ...)
```

**Arguments**

path	Path to the folder containing the files to format. Only files with the .R extension will be formatted.
recursive	Include all subfolders?
skipAutogenerated	Skip autogenerated files such as RcppExports.R?
...	Parameters to be passed on the the formatRFile function

**Examples**

```
## Not run:  
formatRFolder()  
  
## End(Not run)
```

---

formatRText	<i>Format R code</i>
-------------	----------------------

---

**Description**

Format R code

**Usage**

```
formatRText(text, width.cutoff = 100)
```

**Arguments**

text	A character vector with the R code to be formatted.
width.cutoff	Number of characters that each line should be limited to.

**Value**

A character vector with formatted R code.

---

```
insertCirceDefinitionInPackage
```

*Load a Circe definition and insert it into this package*

---

### Description

Load a Circe definition and insert it into this package

### Usage

```
insertCirceDefinitionInPackage(definitionId, name = NULL,
    baseUrl = "http://hixbeta.jnj.com:8081/WebAPI")
```

### Arguments

definitionId	The number indicating which Circe definition to fetch.
name	The name that will be used for the json and SQL files. If not provided, the name in Circe will be used, but this may not lead to valid file names.
baseUrl	The base URL for the WebApi instance.

### Details

Load a Circe definition from a WebApi instance and insert it into this package. This will fetch the json object and store it in the 'inst/circe' folder, and fetch the template SQL and store it in the 'inst/sql/sql\_server' folder. Both folders will be created if they don't exist.

### Examples

```
## Not run:
# This will create 'inst/circe/MyocardialInfarction.json' and
# 'inst/sql/sql_server/MyocardialInfarction.sql':

insertCirceDefinitionInPackage(280, "MyocardialInfarction")

## End(Not run)
```

---

```
makeCluster
```

*Create a cluster of nodes for parallel computation*

---

### Description

Create a cluster of nodes for parallel computation

### Usage

```
makeCluster(numberOfThreads, singleThreadToMain = TRUE)
```



**Arguments**

numberOfThreads

Number of parallel threads.

singleThreadToMain

If numberOfThreads is 1, should we fall back to running the process in the main thread?

**Value**

An object representing the cluster.

---

matchInList	<i>In a list of object of the same type, find those that match the input</i>
-------------	--

---

**Description**

In a list of object of the same type, find those that match the input

**Usage**

matchInList(x, toMatch)

**Arguments**

x

A list of objects of the same type.

toMatch

The object to match.

**Details**

Typically, toMatch will contain a subset of the variables that are in the objects in the list. Any object matching all variables in toMatch will be included in the result.

**Value**

A list of objects that match the toMatch object.

---

nrowOfFakeDf	<i>Return number of rows in fakeDf</i>
--------------	--

---

**Description**

Return number of rows in fakeDf

**Usage**

nrowOfFakeDf(fakeDf)

**Arguments**

fakeDf

The fakeDf object

**Value**

Number of rows.

---

ohdsiLintrFile	<i>Check R file for code errors</i>
----------------	-------------------------------------

---

**Description**

Check R file for code errors

**Usage**

```
ohdsiLintrFile(file)
```

**Arguments**

file	The R file to check
------	---------------------

**Details**

This function uses the lintr package to check the code.

---

ohdsiLintrFolder	<i>Check all R files in a folder</i>
------------------	--------------------------------------

---

**Description**

Check all R files in a folder

**Usage**

```
ohdsiLintrFolder(path = ".", recursive = TRUE)
```

**Arguments**

path	Path to the folder containing the files to check. Only files with the .R extension will be checked.
recursive	Include all subfolders?

**Details**

This function uses the lintr package to check the code.

---

OhdsiRTools	<i>OhdsiRTools</i>
-------------	--------------------

---

**Description**

OhdsiRTools

---

prettyPrint	<i>Print a list of objects</i>
-------------	--------------------------------

---

**Description**

Print a list of objects

**Usage**

```
prettyPrint(object)
```

**Arguments**

object	The list to print.
--------	--------------------

**Details**

Will print nested lists using indentation.

---

rbindFakeDfs	<i>Bind two fakeDf objects by rows</i>
--------------	--

---

**Description**

Bind two fakeDf objects by rows

**Usage**

```
rbindFakeDfs(fakeDf1, fakeDf2)
```

**Arguments**

fakeDf1	The first fakeDf object.
fakeDf2	The second fakeDf object.

**Value**

A fakeDf object with the rows of both fakeDf1 and fakeDf2.

---

selectFromList	<i>Select variables from a list of objects of the same type</i>
----------------	---

---

**Description**

Select variables from a list of objects of the same type

**Usage**

```
selectFromList(x, select)
```

**Arguments**

x	A list of objects of the same type.
select	A character vector of names of variables to select.

---

stopCluster	<i>Stop the cluster</i>
-------------	-------------------------

---

**Description**

Stop the cluster

**Usage**

```
stopCluster(cluster)
```

**Arguments**

cluster	The cluster to stop
---------	---------------------

---

subsetFakeDf	<i>Create a subset of fakeDf object</i>
--------------	---

---

**Description**

Create a subset of fakeDf object

**Usage**

```
subsetFakeDf(fakeDf, index)
```

**Arguments**

fakeDf	The fakeDf object to subset.
index	An integer vector identifying the rows to extract.

**Value**

An object of type fakeDf, which is a list of vectors.

---

subsetFfdFToFakeDf	<i>Create a subset of an ffdF object as a fakeDf object</i>
--------------------	---

---

**Description**

Create a subset of an ffdF object as a fakeDf object

**Usage**

```
subsetFfdFToFakeDf(ffdf, index)
```

**Arguments**

ffdf	The ffdF object to subset.
index	An integer vector or ri object identifying the rows to extract.

**Value**

An object of type fakeDf, which is a list of vectors.

# Index

checkUsagePackage, [2](#)  
clusterApply, [3](#)  
clusterRequire, [3](#)  
convertArgsToList, [4](#)  
convertFakeDfToDataFrame, [4](#)  
createArgFunction, [5](#)  
  
excludeFromList, [5](#)  
  
fakeDfToFdf, [6](#)  
formatRFile, [6](#)  
formatRFolder, [7](#)  
formatRText, [7](#)  
  
insertCirceDefinitionInPackage, [8](#)  
  
makeCluster, [8](#)  
matchInList, [9](#)  
  
nrowOfFakeDf, [9](#)  
  
ohdsiLintrFile, [10](#)  
ohdsiLintrFolder, [10](#)  
OhdsiRTools, [10](#)  
OhdsiRTools-package (OhdsiRTools), [10](#)  
  
prettyPrint, [11](#)  
  
rbindFakeDfs, [11](#)  
  
selectFromList, [12](#)  
stopCluster, [12](#)  
subsetFakeDf, [12](#)  
subsetFdfToFakeDf, [13](#)