Package 'OhdsiRTools'

June 3, 2022

Type Package

Title Tools Used by Observational Health Data Science and Informatics (OHDSI)

Version 2.0.2

Date 2022-06-03

Maintainer Martijn Schuemie <schuemie@ohdsi.org>

Description
Includes functions to format and check syntax of R code and packages following the 'OHDSI' R style guidelines. Generate renv lock files for OHDSI study packages.

License Apache License 2.0

Imports remotes,
codetools,
formatR,
RJSONIO,
httr (>= 1.3.1),
methods,
utils

Suggests renv,
testthat

URL http://ohdsi.github.io/OhdsiRTools/, https://github.com/OHDSI/OhdsiRTools

BugReports https://github.com/OHDSI/OhdsiRTools/issues

NeedsCompilation no

RoxygenNote 7.1.2

Encoding UTF-8

R topics documented:

`checkUsagePackage` *Check all code in a package*

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.

Examples

```
checkUsagePackage("OhdsiRTools")
```

`createRenvLockFile` *Create a renv lock file*

Description

Create a renv lock file

Usage

```
createRenvLockFile(
  rootPackage,
  mode = "auto",
  includeRootPackage = TRUE,
  additionalRequiredPackages = NULL,
  ohdsiGitHubPackages = getOhdsiGitHubPackages(),
  ohdsiStudiesGitHubPackages = rootPackage,
  fileName = "renv.lock",
  restart = (mode == "auto")
)
```

Arguments

`rootPackage`	The name of the root package, the package that we'd like to be able to run in the end.
`mode`	Can be "auto" or "description". See details.
`includeRootPackage`	Include the root package in the renv file?
`additionalRequiredPackages`	Additional packages we want to have installed (with their dependencies), such as 'keyring'. Ignored if `mode = "auto"`.
`ohdsiGitHubPackages`	Names of R packages that need to be installed from the OHDSI GitHub.
`ohdsiStudiesGitHubPackages`	Names of R packages that need to be installed from the OHDSI-Studies GitHub.
`fileName`	Name of the lock file to be generated. Ignored if `mode = "auto"`.
`restart`	Boolean; attempt to restart the R session after initializing the project?

Details

Create a lock file that allows reconstruction of the R environment using the `renv` package. This function will include the root file and all of its dependencies in the lock file, requiring the same package versions as currently installed on this computer.

If `mode = "auto"`, this function will invoke `renv::init()`, which in turn will scan the project folders for any dependencies that are referenced. Afterwards, references to OHDSI packages will be altered so the correct GitHub tags are used for the installed versions.

If `mode = "description"`, this function will assume the project is a full-fledged R package with up-to-date DESCRIPTION, and will only install the dependencies listed in the DESCRIPION.

The second option tends to lead to smaller lock files, but requires all dependencies are accurately listed in the DESCRIPTION file of the study package.

Value

Does not return a value. Is executed for the side-effect of creating the lock file.

---

findNonAsciiStringsInFolder

Find non-ASCII strings in files

Description

Find non-ASCII string in files

Usage

```
findNonAsciiStringsInFolder(path = ".", recursive = TRUE, pattern = "*.R$")
```

Arguments

path	Path to the folder containing the files matching the pattern parameter.
recursive	If TRUE, subfolders will also be searched for files matching the pattern parameter.
pattern	The regular expression to use for selecting files. The default is .R files.

Value

A table listing the lines per file containing non-ASCII characters.

---

fixHadesLogo

Fix HADES logo in pkgdown output

Description

In all HTML files in the docs folder, each occurrence of 'hadesLogo' is replaced with an HTML image tag referring to the HADES logo.

Usage

```
fixHadesLogo(path = ".")
```

Arguments

path	Path to the root of the package for which the pkgdown output needs to be fixed.

Value

This function returns nothing.

`fixHadesPackagesInLockFile` *Fix HADES packages in lock file*

Description

For all HADES packages that must be installed from GitHub, ensures the lock file points to the git tag corresponding to the installed version. If no corresponding tag exists, a warning is raised.

Usage

```
fixHadesPackagesInLockFile(
  lockFile = "renv.lock",
  ohdsiGitHubPackages = getOhdsiGitHubPackages()
)
```

Arguments

`lockFile`	The name of the lock file to fix.
`ohdsiGitHubPackages`	Names of R packages that need to be installed from the OHDSI GitHub.

Value

Returns nothing. Fixes the lock file.

`formatRFile` *Format an R file*

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.

Details

DEPRECRATED. Please use `styler::style_file` instead.

`formatRFolder` *Format all R files in a folder*

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 auto-generated files such as RcppExports.R?
`...`	Parameters to be passed on the the formatRFile function

Details

DEPRECRATED. Please use `styler::style_dir` instead.

`formatRText` *Format R code*

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.

Details

DEPRECRATED. Please use `styler::style_text` instead.

Value

A character vector with formatted R code.

`getCorePackages` *Get a list of R core packages*

Description

Get a list of R core packages

Usage

```
getCorePackages()
```

Details

Returns names of packages that are part of the R code, and can therefore not be installed.

Value

A character vector.

`getOhdsiGitHubPackages` *Get a list of packages in the OHDSI GitHub.*

Description

Get a list of packages in the OHDSI GitHub.

Usage

```
getOhdsiGitHubPackages()
```

Details

Returns names of packages that need to be installed from https://github.com/ohdsi. Connects to GitHub to get the latest list.

Value

A character vector.

insertEnvironmentSnapshotInPackage *Store snapshot of the R environment in the package*

Description

Store snapshot of the R environment in the package

Usage

```
insertEnvironmentSnapshotInPackage(
  rootPackage,
  pathToCsv = "inst/settings/rEnvironmentSnapshot.csv"
)
```

Arguments

`rootPackage`	The name of the root package
`pathToCsv`	The path for saving the snapshot (as CSV file).

Details

This function records all versions used in the R environment that are used by one root package, and stores them in a CSV file in the R package that is currently being developed. The default location is `inst/settings/rEnvironmentSnapshot.csv`.This can be used for example to restore the environment to the state it was when a particular study package was run using the `restoreEnvironment` function.

Examples

```
## Not run:
insertEnvironmentSnapshotInPackage("OhdsiRTools")

## End(Not run)
```

restoreEnvironment *Restore the R environment to a snapshot*

Description

Restore the R environment to a snapshot

Usage

```
restoreEnvironment(
  snapshot,
  stopOnWrongRVersion = FALSE,
  strict = FALSE,
  skipLast = TRUE
)
```

Arguments

`snapshot`	The snapshot data frame as generated using the `takeEnvironmentSnapshot` function.
`stopOnWrongRVersion`	Should the function stop when the wrong version of R is installed? Else just a warning will be thrown when the version doesn't match.
`strict`	If TRUE, the exact version of each package will installed. If FALSE, a package will only be installed if (a) a newer version is required than currently installed, or (b) the major version number is different.
`skipLast`	Skip last entry in snapshot? This is usually the study package that needs to be installed manually.

Details

This function restores the R environment to a previous snapshot, meaning all the packages will be restored to the versions they were at at the time of the snapshot. Note: on Windows you will very likely need to have RTools installed to build the various packages.

Examples

```
## Not run:
snapshot <- takeEnvironmentSnapshot("OhdsiRTools")
write.csv(snapshot, "snapshot.csv")

# 5 years later

snapshot <- read.csv("snapshot.csv")
restoreEnvironment(snapshot)

## End(Not run)
```

`restoreEnvironmentFromPackage` *Restore environment stored in package*

Description

Restore environment stored in package

Usage

```
restoreEnvironmentFromPackage(
  pathToCsv = "inst/settings/rEnvironmentSnapshot.csv",
  stopOnWrongRVersion = FALSE,
  strict = FALSE,
  skipLast = TRUE
)
```

Arguments

`pathToCsv`	The path for saving the snapshot (as CSV file).
`stopOnWrongRVersion`	Should the function stop when the wrong version of R is installed? Else just a warning will be thrown when the version doesn't match.
`strict`	If TRUE, the exact version of each package will installed. If FALSE, a package will only be installed if (a) a newer version is required than currently installed, or (b) the major version number is different.
`skipLast`	Skip last entry in snapshot? This is usually the study package that needs to be installed manually.

Details

This function restores all packages (and package versions) described in the environment snapshot stored in the package currently being developed. The default location is `inst/settings/rEnvironmentSnapshot.csv`.

Examples

```
## Not run:
restoreEnvironmentFromPackage()

## End(Not run)
```

`restoreEnvironmentFromPackageOnGithub` *Restore environment stored in package*

Description

Restore environment stored in package

Usage

```
restoreEnvironmentFromPackageOnGithub(
  githubPath,
  pathToCsv = "inst/settings/rEnvironmentSnapshot.csv",
  stopOnWrongRVersion = FALSE,
  strict = FALSE,
  skipLast = TRUE
)
```

Arguments

`githubPath`	The path for the GitHub repo containing the package (e.g. 'OHDSI/StudyProtocols/AlendronateVsRa
`pathToCsv`	The path for the snapshot inside the package.
`stopOnWrongRVersion`	Should the function stop when the wrong version of R is installed? Else just a warning will be thrown when the version doesn't match.

`strict`	If TRUE, the exact version of each package will installed. If FALSE, a package will only be installed if (a) a newer version is required than currently installed, or (b) the major version number is different.
`skipLast`	Skip last entry in snapshot? This is usually the study package that needs to be installed manually.

Details

This function restores all packages (and package versions) described in the environment snapshot stored in the package currently being developed. The default location is `inst/settings/rEnvironmentSnapshot.csv`.

Examples

```
## Not run:
restoreEnvironmentFromPackageOnGithub("OHDSI/StudyProtocols/AlendronateVsRaloxifene")

## End(Not run)
```

`takeEnvironmentSnapshot` *Take a snapshot of the R environment*

Description

Take a snapshot of the R environment

Usage

```
takeEnvironmentSnapshot(rootPackage)
```

Arguments

`rootPackage`	The name of the root package

Details

This function records all versions used in the R environment that are used by one root package. This can be used for example to restore the environment to the state it was when a particular study package was run using the `restoreEnvironment` function.

Value

A data frame listing all the dependencies of the root package and their version numbers, in the order in which they should be installed.

Examples

```
snapshot <- takeEnvironmentSnapshot("OhdsiRTools")
snapshot
```

updateCopyrightYearFile *Update the copyright year in a R or SQL file*

Description

Update the copyright year in a R or SQL file

Usage

```
updateCopyrightYearFile(file)
```

Arguments

`file`	The path to the file.

updateCopyrightYearFolder *Update the copyright year in all R and SQL files in a folder*

Description

Update the copyright year in all R and SQL files in a folder

Usage

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

Arguments

`path`	Path to the folder containing the files to update. Only files with the .R and .SQL extension will be updated.
`recursive`	Include all subfolders?

updatePackageName *Update the package name in a R or SQL file*

Description

Update the package name in a R or SQL file

Usage

```
updatePackageName(file, packageName)
```

Arguments

`file`	The path to the file.
`packageName`	The replacement package name

`updatePackageNameFolder`
Update the package name in all R and SQL files in a folder

Description

Update the package name in all R and SQL files in a folder

Usage

```
updatePackageNameFolder(path = ".", packageName, recursive = TRUE)
```

Arguments

`path`	Path to the folder containing the files to update. Only files with the .R and .SQL extension will be updated.
`packageName`	The replacement package name
`recursive`	Include all subfolders?

Index