Package ‘OhdsiRTools’

June 3, 2020

Type Package

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

Version 1.9.0

Date 2020-06-03

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. Support for parallel computation, logging, and function automation. Functionality
for interacting with instances of the open source 'OHDSI' WebApi, which can be found at <https://github.com/OHDSI/WebAPI>.

License Apache License 2.0

VignetteBuilder knitr

Depends R (>= 3.1.0),

Imports devtools,
codetools,
formatR,
snow,
RJSONIO,
httr (>= 1.3.1),
openxlsx (>= 4.0.17),
XML,
jsonlite,
methods,
utils,
mailR

Suggests testthat,
shiny,
DT,
knitr,
rmarkdown

URL https://github.com/OHDSI/OhdsiRTools

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

NeedsCompilation no

RoxygenNote 7.1.0

Encoding UTF-8

R topics documented:

checkUsagePackage 2
createConceptSetWorkbook 3
createRenvLockFile 4
findNonAsciiStringsInFolder 4
fixHadesLogo 5
formatRFile 5
formatRFolder 6
formatRText 6
getCohortDefinitionExpression 7
getCohortDefinitionName 8
getCohortDefinitionSql 8
getCohortGenerationStatuses 9
getConceptSetConceptIds 9
getConceptSetExpression 10
getConceptSetName 11
getConceptSetsAndConceptsFromCohort 11
getCorePackages 12
getOhdsiGitHubPackages 12
getPriorityVocabKey 13
getSetExpressionConceptIds 13
insertCohortDefinitionInPackage 14
insertCohortDefinitionSetInPackage 15
insertConceptSetConceptIdsInPackage 16
insertEnvironmentSnapshotInPackage 16
invokeCohortSetGeneration 17
restoreEnvironment 17
restoreEnvironmentFromPackage 18
restoreEnvironmentFromPackageOnGithub 19
runAndNotify 20
takeEnvironmentSnapshot 21
updateCopyrightYearFile 22
updateCopyrightYearFolder 22
updatePackageName 22
updatePackageNameFolder 23

Index 24

`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")
```

`createConceptSetWorkbook` *Save a set of concept sets expressions, included concepts, and mapped concepts into a workbook*

Description

Save a set of concept sets expressions, included concepts, and mapped concepts into a workbook

Usage

```
createConceptSetWorkbook(
  conceptSetIds,
  workFolder = NULL,
  baseUrl,
  included = FALSE,
  mapped = FALSE
)
```

Arguments

`conceptSetIds`	A vector of concept set IDs.
`workFolder`	Directory location where the workbook will be saved, defaults to working directory.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`included`	Should included concepts be included in the workbook?
`mapped`	Should mapped concepts be included in the workbook?

Value

A xlsx workbook (conceptSetExpressions.xlsx) that includes a list of all concept set IDs and names and a worksheet for the concepts in each set. Options to include an included concepts and mapped concepts worksheet for each concept set are available.

createRenvLockFile *Create a renv lock file*

Description

Create a renv lock file

Usage

```
createRenvLockFile(
  rootPackage,
  ohdsiGitHubPackages = getOhdsiGitHubPackages(),
  ohdsiStudiesGitHubPackages = rootPackage,
  fileName = "renv.lock"
)
```

Arguments

`rootPackage`	The name of the root package, the package that we'd like to be able to run in the end.
`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.

Details

Create a lock file that allows recontruction 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.

Value

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

findNonAsciiStringsInFolder *Find non-ASCII strings in R files*

Description

Find non-ASCII string in R files.

Usage

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

Arguments

`path`	Path to the folder containing the R files.
`recursive`	If TRUE, subfolders will also be searched for R files.

Value

A table listing the lines per R 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.

`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.

`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

`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.

Value

A character vector with formatted R code.

Examples

```
code <- "
#' Example functon
#'
#' @param x One argument.
#' @param fooBar Another argument.
#'
#' @examples
#' example(x=1,fooBar='abc')
#'
#'@export
example <- function(x,foobar){paste(x,foobar)}
"

formatted <- formatRText(code)
writeLines(formatted)
```

getCohortDefinitionExpression

Get a cohort definition expression

Description

Get a cohort definition expression

Usage

```
getCohortDefinitionExpression(definitionId, baseUrl)
```

Arguments

definitionId	The number indicating which cohort definition to fetch.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Obtain the JSON expression from WebAPI for a given cohort id

Value

A JSON list object representing the cohort definition

Examples

```
## Not run:
# This will obtain a cohort definition's JSON expression:

getCohortDefinitionExpression(definitionId = 282,
                              baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getCohortDefinitionName

Get a cohort definition's name from WebAPI

Description

Get a cohort definition's name from WebAPI

Usage

```
getCohortDefinitionName(baseUrl, definitionId, formatName = FALSE)
```

Arguments

baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
definitionId	The cohort definition id in Atlas.
formatName	Should the name be formatted to remove prefixes and underscores?

Details

Obtains the name of a cohort.

Value

The name of the cohort.

getCohortDefinitionSql

Get a cohort definition's SQL from WebAPI

Description

Get a cohort definition's SQL from WebAPI

Usage

```
getCohortDefinitionSql(baseUrl, definitionId)
```

Arguments

baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
definitionId	The cohort definition id in Atlas.

Details

Obtains the template SQL of a cohort.

Value

The templated SQL to generate the cohort

getCohortGenerationStatuses

Get Cohort Generation Statuses

Description

Get Cohort Generation Statuses

Usage

```
getCohortGenerationStatuses(baseUrl, definitionIds, sourceKeys)
```

Arguments

`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`definitionIds`	A list of cohort definition Ids
`sourceKeys`	A list of CDM source keys. These can be found in Atlas -> Configure.

Details

Obtains cohort generation statuses for a collection of cohort definition Ids and CDM sources. Useful if running multiple cohort generation jobs that are long-running.

Value

A data frame of cohort generation statuses, start times, and execution durations per definition id and source key.

getConceptSetConceptIds

Get Concept Set Concept Ids

Description

Get Concept Set Concept Ids

Usage

```
getConceptSetConceptIds(baseUrl, setId, vocabSourceKey = NULL)
```

Arguments

`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`setId`	The concept set id in Atlas.
`vocabSourceKey`	The source key of the Vocabulary. By default, the priority Vocabulary is used.

Details

Obtains the full list of concept Ids in a concept set.

Value

A list of concept Ids.

`getConceptSetExpression` *Get a concept set expression*

Description

Get a concept set expression

Usage

```
getConceptSetExpression(baseUrl, setId)
```

Arguments

`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`setId`	The concept set id in Atlas.

Details

Obtain the JSON expression from WebAPI for a given concept set

Value

A JSON list object representing the concept set

Examples

```
## Not run:
# This will obtain a concept set's JSON expression:

getConceptSetExpression(setId = 282,
                        baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getConceptSetName *Get a concept set's name from WebAPI*

Description

Get a concept set's name from WebAPI

Usage

```
getConceptSetName(baseUrl, setId, formatName = FALSE)
```

Arguments

baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
setId	The concept set id in Atlas.
formatName	Should the name be formatted to remove prefixes and underscores?

Details

Obtains the name of a concept set.

Value

The name of the concept set.

getConceptSetsAndConceptsFromCohort *Get a list of concept sets and concepts from a cohort definition*

Description

Get a list of concept sets and concepts from a cohort definition

Usage

```
getConceptSetsAndConceptsFromCohort(
  baseUrl,
  definitionId,
  vocabSourceKey = NULL
)
```

Arguments

baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
definitionId	The cohort id to fetch concept sets and concepts from
vocabSourceKey	A mysterious parameter.

Details

For a given cohort definition id, get all concept sets and resolve all concepts from each

Value

A list of concept sets, set names, and concepts

Examples

```
## Not run:
# This will obtain a list of concept sets and concepts from a cohort id:

getConceptsFromCohortId(baseUrl = "http://server.org:80/WebAPI",
                        definitionId = 123)

## End(Not run)
```

`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.

Value

A character vector.

`getPriorityVocabKey` *Get Priority Vocab Source Key*

Description

Get Priority Vocab Source Key

Usage

```
getPriorityVocabKey(baseUrl)
```

Arguments

`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Obtains the source key of the default OMOP Vocab in Atlas.

Value

A string with the source key of the default OMOP Vocab in Atlas.

`getSetExpressionConceptIds` *Get Concepts from a Concept Set Expression*

Description

Get Concepts from a Concept Set Expression

Usage

```
getSetExpressionConceptIds(baseUrl, expression, vocabSourceKey = NULL)
```

Arguments

`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`expression`	A JSON string that represents the concept set expression
`vocabSourceKey`	The source key of the Vocabulary. By default, the priority Vocabulary is used.

Value

A list of concept ids

Examples

```
## Not run:
# This will obtain the concept ids from a concept set expression:

getSetExpressionConceptIds(baseUrl = "http://server.org:80/WebAPI",
                           expression = someJsonExpression)

## End(Not run)
```

insertCohortDefinitionInPackage

Load a cohort definition and insert it into this package

Description

Load a cohort definition and insert it into this package

Usage

```
insertCohortDefinitionInPackage(
  definitionId,
  name = NULL,
  baseUrl,
  generateStats = FALSE
)
```

Arguments

definitionId	The number indicating which cohort definition to fetch.
name	The name that will be used for the json and SQL files. If not provided, the name in cohort will be used, but this may not lead to valid file names.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
generateStats	Should the SQL include the code for generating inclusion rule statistics? Note that if TRUE, several additional tables are expected to exists as described in the details.

Details

Load a cohort definition from a WebApi instance and insert it into this package. This will fetch the json object and store it in the 'inst/cohorts' 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. When using generateStats = TRUE, the following tables are required to exist when executing the SQL: cohort_inclusion, cohort_inclusion_result, cohort_inclusion_stats, and cohort_summary_stats. Also note that the cohort_inclusion table should be populated with the names of the rules prior to executing the cohort definition SQL.

Examples

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

insertCohortDefinitionInPackage(definitionId = 282,
                                name = "Angioedema",
                                baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

insertCohortDefinitionSetInPackage

Insert a set of cohort definitions into package

Description

Insert a set of cohort definitions into package

Usage

```
insertCohortDefinitionSetInPackage(
  fileName,
  baseUrl,
  insertTableSql = TRUE,
  insertCohortCreationR = TRUE,
  generateStats = FALSE,
  packageName
)
```

Arguments

`fileName`	Name of a CSV file in the inst/settings folder of the package specifying the cohorts to insert. See details for the expected file format.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`insertTableSql`	Should the SQL for creating the cohort table be inserted into the package as well? This file will be called CreateCohortTable.sql.
`insertCohortCreationR`	Insert R code that will create the cohort table and instantiate the cohorts? This will create a file called R/CreateCohorts.R containing a function called `.createCohorts`.
`generateStats`	Should cohort inclusion rule statistics be created?
`packageName`	The name of the package (only needed when inserting the R code as well).

Details

The CSV file should have at least the following fields:

atlasId The cohort ID in ATLAS.

cohortId The cohort ID that will be used when instantiating the cohort (can be different from atlasId).

name The name to be used for the cohort. This name will be used to generate file names, so please use letters and numbers only (no spaces).

insertConceptSetConceptIdsInPackage

Insert a set of concept sets' concept ids into package

Description

Insert a set of concept sets' concept ids into package

Usage

```
insertConceptSetConceptIdsInPackage(fileName, baseUrl)
```

Arguments

fileName	Name of a CSV file in the inst/settings folder of the package specifying the concept sets to insert. See details for the expected file format.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

The CSV file should have:

atlasId The concept set Id in ATLAS.

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)
```

invokeCohortSetGeneration
Invoke the generation of a set of cohort definitions

Description

Invoke the generation of a set of cohort definitions

Usage

```
invokeCohortSetGeneration(baseUrl, sourceKeys, definitionIds)
```

Arguments

baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
sourceKeys	A list of CDM source keys. These can be found in Atlas -> Configure.
definitionIds	A list of cohort definition Ids

Details

Invokes the generation of a set of cohort definitions across a set of CDMs set up in WebAPI. Use getCohortGenerationStatuses to check the progress of the set.

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)
```

`runAndNotify` *Run code and send e-mail notification on error, warning, or completion*

Description

Run code and send e-mail notification on error, warning, or completion

Usage

```
runAndNotify(expression, mailSettings, label = "R", stopOnWarning = FALSE)
```

Arguments

`expression` The expression to run.

`mailSettings` Arguments to be passed to the send.mail function in the mailR package (except subject and body).

`label` A label to be used in the subject to identify a run.

`stopOnWarning` Stop expression on warning and send notification?

Value

The output of `expression`.

Examples

```
## Not run:
mailSettings <- list(from = "someone@gmail.com",
                     to = c("someone_else@gmail.com"),
                     smtp = list(host.name = "smtp.gmail.com",
                                 port = 465,
                                 user.name = "someone@gmail.com",
                                 passwd = "super_secret!",
                                 ssl = TRUE),
                     authenticate = TRUE,
                     send = TRUE)
runAndNotify({
  a <- 1 + 2 + 3
}, mailSettings = mailSettings, label = "My fancy R code")

## 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

checkUsagePackage, 2
createConceptSetWorkbook, 3
createRenvLockFile, 4

findNonAsciiStringsInFolder, 4
fixHadesLogo, 5
formatRFile, 5
formatRFolder, 6
formatRText, 6

getCohortDefinitionExpression, 7
getCohortDefinitionName, 8
getCohortDefinitionSql, 8
getCohortGenerationStatuses, 9
getConceptSetConceptIds, 9
getConceptSetExpression, 10
getConceptSetName, 11
getConceptSetsAndConceptsFromCohort, 11
getCorePackages, 12
getOhdsiGitHubPackages, 12
getPriorityVocabKey, 13
getSetExpressionConceptIds, 13

insertCohortDefinitionInPackage, 14
insertCohortDefinitionSetInPackage, 15
insertConceptSetConceptIdsInPackage, 16
insertEnvironmentSnapshotInPackage, 16
invokeCohortSetGeneration, 17

restoreEnvironment, *16*, 17, *21*
restoreEnvironmentFromPackage, 18
restoreEnvironmentFromPackageOnGithub, 19
runAndNotify, 20

takeEnvironmentSnapshot, *18*, 21

updateCopyrightYearFile, 22
updateCopyrightYearFolder, 22
updatePackageName, 22
updatePackageNameFolder, 23