Package ‘ROhdsiWebApi’

March 14, 2023

Type Package

Title Interacting With an OHDSI WebApi Instance

Version 1.3.3

Date 2023-03-14

Maintainer Gowtham Rao <rao@ohdsi.org>

Description ROhdsiWebApi is an R based interface to OHDSI WebAPI services, and performs GET/PULL/POST/DELETE calls via the WebApi.
All objects starting from R or output to R - are analysis ready R-objects like list and data.frame.
The package handles the intermediary steps by converting R-objects to JSON and vice versa.

License Apache License 2.0

Depends R (>= 3.1.0),

Imports checkmate,
dplyr,
httr (>= 1.3.1),
lubridate,
openxlsx (>= 4.0.17),
purrr,
readr,
RJSONIO,
scales,
SqlRender,
stringr,
tibble,
tidyr,
rlang

Suggests testthat,
knitr,
rmarkdown,
httpuv,
jsonlite,
withr,
httptest,
stringi

URL https://ohdsi.github.io/ROhdsiWebApi/, https://github.com/OHDSI/ROhdsiWebApi

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

NeedsCompilation no

RoxygenNote 7.2.3

Encoding UTF-8

VignetteBuilder knitr

Language en-US

R topics documented:

`authorizeWebApi` *Authorize ROhdsiWebApi to access a protected instance of WebAPI Authorize the ROhdsiWebApi package to access WebApi on behalf of the user. This can be done with any of the auth methods described below. authorizeWebApi will use attempt to retrieve, cache, and update a token which will grant access to webAPI by all subsequent requests made by the package.*

Description

Authorize ROhdsiWebApi to access a protected instance of WebAPI Authorize the ROhdsiWebApi package to access WebApi on behalf of the user. This can be done with any of the auth methods described below. authorizeWebApi will use attempt to retrieve, cache, and update a token which will grant access to webAPI by all subsequent requests made by the package.

Usage

```
authorizeWebApi(
  baseUrl,
  authMethod,
  webApiUsername = NULL,
  webApiPassword = NULL
)
```

Arguments

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

`authMethod` The method used for authentication to WebAPI. Options are

- "db"Database authentication using Atlas/WebAPI built in auth
- "ad"Active Directory
- "windows"Windows NT authentication

The auth method must be enabled in the instance of WebAPI pointed to by baseUrl.

`webApiUsername` A character string containing the WebApi username passed on to authentication methods

`webApiPassword` An character string containing a WebApi password passed on to authentication methods. By default the user will be prompted for their password when needed.

`cancelCharacterizationGeneration` *Cancel generation of Characterization id.*

Description

Cancel generation of Characterization id.

Usage

```
cancelCharacterizationGeneration(characterizationId, baseUrl, sourceKey)
```

Arguments

`characterizationId`	An integer id representing the id that uniquely identifies a Characterization definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`sourceKey`	The source key for a CDM instance in WebAPI, as defined in the Configuration page

Details

Cancel the generation of Characterization id in the WebApi.

Value

A tibble with job status information.

Examples

```
## Not run:
cancelCharacterizationGeneration(characterizationId = 13242,
                                 baseUrl = "http://server.org:80/WebAPI",
                                 sourceKey = "HCUP")

## End(Not run)
```

`cancelCohortGeneration` *Cancel generation of Cohort id.*

Description

Cancel generation of Cohort id.

Usage

```
cancelCohortGeneration(cohortId, baseUrl, sourceKey)
```

Arguments

`cohortId`	An integer id representing the id that uniquely identifies a Cohort definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`sourceKey`	The source key for a CDM instance in WebAPI, as defined in the Configuration page

Details

Cancel the generation of Cohort id in the WebApi.

Value

A tibble with job status information.

Examples

```
## Not run:
cancelCohortGeneration(cohortId = 13242,
                        baseUrl = "http://server.org:80/WebAPI",
                        sourceKey = "HCUP")

## End(Not run)
```

`cancelGeneration` *Invoke generation.*

Description

Invoke generation.

Usage

```
cancelGeneration(id, baseUrl, sourceKey, category)
```

Arguments

`id`	An integer id representing the id that uniquely identifies a definition for the category in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`sourceKey`	The source key for a CDM instance in WebAPI, as defined in the Configuration page
`category`	These are the categories in WebApi. The valid string options are 'conceptSet', 'cohort', 'characterization', 'pathway, 'incidenceRate', 'estimation', 'prediction'.

Details

Invoke generation (execution) information.

Value

Error message if invoke failed.

Examples

```
## Not run:
cancelGeneration(id = 13242, category = "cohort", baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

`cancelIncidenceRateGeneration` *Cancel generation of IncidenceRate id.*

Description

Cancel generation of IncidenceRate id.

Usage

```
cancelIncidenceRateGeneration(incidenceRateId, baseUrl, sourceKey)
```

Arguments

`incidenceRateId`	An integer id representing the id that uniquely identifies a IncidenceRate definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`sourceKey`	The source key for a CDM instance in WebAPI, as defined in the Configuration page

Details

Cancel the generation of IncidenceRate id in the WebApi.

Value

A tibble with job status information.

Examples

```
## Not run:
cancelIncidenceRateGeneration(incidenceRateId = 13242,
                              baseUrl = "http://server.org:80/WebAPI",
                              sourceKey = "HCUP")

## End(Not run)
```

cancelPathwayGeneration

Cancel generation of Pathway id.

Description

Cancel generation of Pathway id.

Usage

```
cancelPathwayGeneration(pathwayId, baseUrl, sourceKey)
```

Arguments

pathwayId	An integer id representing the id that uniquely identifies a Pathway definition in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
sourceKey	The source key for a CDM instance in WebAPI, as defined in the Configuration page

Details

Cancel the generation of Pathway id in the WebApi.

Value

A tibble with job status information.

Examples

```
## Not run:
cancelPathwayGeneration(pathwayId = 13242,
                        baseUrl = "http://server.org:80/WebAPI",
                        sourceKey = "HCUP")

## End(Not run)
```

checkInputFileEncoding

Check character encoding of input file

Description

For its input files, CohortDiagnostics only accepts UTF-8 or ASCII character encoding. This function can be used to check whether a file meets these criteria.

Usage

```
checkInputFileEncoding(fileName)
```

Arguments

`fileName`	The path to the file to check

Value

Throws an error if the input file does not have the correct encoding.

`convertConceptSetDefinitionToTable`

Convert a concept set definition to a table

Description

Convert a concept set definition to a table

Usage

```
convertConceptSetDefinitionToTable(conceptSetDefinition)
```

Arguments

`conceptSetDefinition`	A concept set definition, for example as obtained through the `getConceptSetDefinition` function, or taken from a cohort definition.

Value

Takes a R (list) representation of the Concept Set expression and returns a table (dataframe) representing the concept set expression. This is useful to create publication friendly output of the concept set expression.

Examples

```
## Not run:
conceptSetDefinition <- getConceptSetDefinition(conceptSetId = 282,
                                                baseUrl = "http://server.org:80/WebAPI")
convertConceptSetDefinitionToTable(conceptSetDefinition = conceptSetDefinition)

## End(Not run)
```

`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,
  fileName,
  baseUrl,
  included = FALSE,
  mapped = FALSE
)
```

Arguments

`conceptSetIds`	A vector of concept set IDs.
`fileName`	The name of the XLSX workbook file.
`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 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.

`deleteCharacterizationDefinition`
Delete Characterization id definition.

Description

Delete Characterization id definition.

Usage

```
deleteCharacterizationDefinition(characterizationId, baseUrl)
```

Arguments

`characterizationId`
An integer id representing the id that uniquely identifies a Characterization definition in a WebApi instance.

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

Details

Delete the Characterization definition from WebAPI for a given Characterization id

Value

None, unless error.

Examples

```
## Not run:
deleteCharacterizationDefinition(characterizationId = 13242,
                                 baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

deleteCohortDefinition *Delete Cohort id definition.*

Description

Delete Cohort id definition.

Usage

```
deleteCohortDefinition(cohortId, baseUrl)
```

Arguments

`cohortId` An integer id representing the id that uniquely identifies a Cohort definition in a WebApi instance.

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

Details

Delete the Cohort definition from WebAPI for a given Cohort id

Value

None, unless error.

Examples

```
## Not run:
deleteCohortDefinition(cohortId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

deleteConceptSetDefinition

Delete ConceptSet id definition.

Description

Delete ConceptSet id definition.

Usage

```
deleteConceptSetDefinition(conceptSetId, baseUrl)
```

Arguments

conceptSetId	An integer id representing the id that uniquely identifies a ConceptSet definition in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Delete the ConceptSet definition from WebAPI for a given ConceptSet id

Value

None, unless error.

Examples

```
## Not run:
deleteConceptSetDefinition(conceptSetId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

deleteDefinition

Delete a definition id of a chosen category.

Description

Delete a definition id of a chosen category.

Usage

```
deleteDefinition(id, baseUrl, category)
```

Arguments

id	An integer id representing the id that uniquely identifies a definition for the category in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
category	These are the categories in WebApi. The valid string options are 'conceptSet', 'cohort', 'characterization', 'pathway, 'incidenceRate', 'estimation', 'prediction'.

Details

Delete the definition for an id of chosen category in WebApi.

Value

None, unless error.

Examples

```
## Not run:
deleteDefinition(id = 13242, baseUrl = "http://server.org:80/WebAPI", category = "cohort")

## End(Not run)
```

`deleteEstimationDefinition` *Delete Estimation id definition.*

Description

Delete Estimation id definition.

Usage

```
deleteEstimationDefinition(estimationId, baseUrl)
```

Arguments

`estimationId`	An integer id representing the id that uniquely identifies a Estimation definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Delete the Estimation definition from WebAPI for a given Estimation id

Value

None, unless error.

Examples

```
## Not run:
deleteEstimationDefinition(estimationId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

deleteIncidenceRateDefinition

Delete IncidenceRate id definition.

Description

Delete IncidenceRate id definition.

Usage

```
deleteIncidenceRateDefinition(incidenceRateId, baseUrl)
```

Arguments

`incidenceRateId`	An integer id representing the id that uniquely identifies a IncidenceRate definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Delete the IncidenceRate definition from WebAPI for a given IncidenceRate id

Value

None, unless error.

Examples

```
## Not run:
deleteIncidenceRateDefinition(incidenceRateId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

deletePathwayDefinition

Delete Pathway id definition.

Description

Delete Pathway id definition.

Usage

```
deletePathwayDefinition(pathwayId, baseUrl)
```

Arguments

`pathwayId`	An integer id representing the id that uniquely identifies a Pathway definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Delete the Pathway definition from WebAPI for a given Pathway id

Value

None, unless error.

Examples

```
## Not run:
deletePathwayDefinition(pathwayId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

deletePredictionDefinition   *Delete Prediction id definition.*

Description

Delete Prediction id definition.

Usage

```
deletePredictionDefinition(predictionId, baseUrl)
```

Arguments

`predictionId`	An integer id representing the id that uniquely identifies a Prediction definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Delete the Prediction definition from WebAPI for a given Prediction id

Value

None, unless error.

Examples

```
## Not run:
deletePredictionDefinition(predictionId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

detectCharacterizationsByName *Detect the presence of string matched Characterization definitions.*

Description

Detect the presence of string matched Characterization definitions.

Usage

```
detectCharacterizationsByName(pattern, negate = FALSE, baseUrl)
```

Arguments

pattern	A pattern to look for. See str_detect for details.
negate	If TRUE, return non-matching elements. See str_detect for details.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Detect string matched Characterization definition names from the WebApi, and retrieve metadata definitions.

Value

FALSE if no matches. If matched - output from getCharacterizationDefinitionsMetaData

Examples

```
## Not run:
detectCharacterizations(pattern = "this text string to search in pattern",
                        baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

detectCohortsByName *Detect the presence of string matched Cohort definitions.*

Description

Detect the presence of string matched Cohort definitions.

Usage

```
detectCohortsByName(pattern, negate = FALSE, baseUrl)
```

Arguments

pattern	A pattern to look for. See str_detect for details.
negate	If TRUE, return non-matching elements. See str_detect for details.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Detect string matched Cohort definition names from the WebApi, and retrieve metadata definitions.

Value

FALSE if no matches. If matched - output from getCohortDefinitionsMetaData

Examples

```
## Not run:
detectCohorts(pattern = "this text string to search in pattern",
              baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

detectConceptSetsByName

Detect the presence of string matched ConceptSet definitions.

Description

Detect the presence of string matched ConceptSet definitions.

Usage

```
detectConceptSetsByName(pattern, negate = FALSE, baseUrl)
```

Arguments

`pattern`	A pattern to look for. See str_detect for details.
`negate`	If TRUE, return non-matching elements. See str_detect for details.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Detect string matched ConceptSet definition names from the WebApi, and retrieve metadata definitions.

Value

FALSE if no matches. If matched - output from getConceptSetDefinitionsMetaData

Examples

```
## Not run:
detectConceptSets(pattern = "this text string to search in pattern",
                  baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

detectEstimationsByName

Detect the presence of string matched Estimation definitions.

Description

Detect the presence of string matched Estimation definitions.

Usage

```
detectEstimationsByName(pattern, negate = FALSE, baseUrl)
```

Arguments

`pattern`	A pattern to look for. See str_detect for details.
`negate`	If TRUE, return non-matching elements. See str_detect for details.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Detect string matched Estimation definition names from the WebApi, and retrieve metadata definitions.

Value

FALSE if no matches. If matched - output from getEstimationDefinitionsMetaData

Examples

```
## Not run:
detectEstimations(pattern = "this text string to search in pattern",
                  baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

detectIncidenceRatesByName

Detect the presence of string matched IncidenceRate definitions.

Description

Detect the presence of string matched IncidenceRate definitions.

Usage

```
detectIncidenceRatesByName(pattern, negate = FALSE, baseUrl)
```

Arguments

`pattern`	A pattern to look for. See str_detect for details.
`negate`	If TRUE, return non-matching elements. See str_detect for details.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Detect string matched IncidenceRate definition names from the WebApi, and retrieve metadata definitions.

Value

FALSE if no matches. If matched - output from getIncidenceRateDefinitionsMetaData

Examples

```
## Not run:
detectIncidenceRates(pattern = "this text string to search in pattern",
                     baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

`detectPathwaysByName` *Detect the presence of string matched Pathway definitions.*

Description

Detect the presence of string matched Pathway definitions.

Usage

```
detectPathwaysByName(pattern, negate = FALSE, baseUrl)
```

Arguments

`pattern`	A pattern to look for. See str_detect for details.
`negate`	If TRUE, return non-matching elements. See str_detect for details.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Detect string matched Pathway definition names from the WebApi, and retrieve metadata definitions.

Value

FALSE if no matches. If matched - output from getPathwayDefinitionsMetaData

Examples

```
## Not run:
detectPathways(pattern = "this text string to search in pattern",
               baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

detectPredictionsByName

Detect the presence of string matched Prediction definitions.

Description

Detect the presence of string matched Prediction definitions.

Usage

```
detectPredictionsByName(pattern, negate = FALSE, baseUrl)
```

Arguments

pattern	A pattern to look for. See str_detect for details.
negate	If TRUE, return non-matching elements. See str_detect for details.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Detect string matched Prediction definition names from the WebApi, and retrieve metadata definitions.

Value

FALSE if no matches. If matched - output from getPredictionDefinitionsMetaData

Examples

```
## Not run:
detectPredictions(pattern = "this text string to search in pattern",
                  baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

existsCharacterizationName *Check if Characterization definition name exists.*

Description

Check if Characterization definition name exists.

Usage

```
existsCharacterizationName(characterizationName, baseUrl)
```

Arguments

`characterizationName`	A string name for the Characterization to be checked.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Check if a string name already exists in the WebApi as a Characterization definition name.

Value

If found, the function will return a tibble with details of the specification. If not found, FALSE will be returned.

Examples

```
## Not run:
existsCharacterizationName(characterizationName = "this text string needs to be checked",
                           baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

existsCohortName *Check if Cohort definition name exists.*

Description

Check if Cohort definition name exists.

Usage

```
existsCohortName(cohortName, baseUrl)
```

Arguments

`cohortName`	A string name for the Cohort to be checked.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Check if a string name already exists in the WebApi as a Cohort definition name.

Value

If found, the function will return a tibble with details of the specification. If not found, FALSE will be returned.

Examples

```
## Not run:
existsCohortName(cohortName = "this text string needs to be checked",
                 baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

existsConceptSetName *Check if ConceptSet definition name exists.*

Description

Check if ConceptSet definition name exists.

Usage

```
existsConceptSetName(conceptSetName, baseUrl)
```

Arguments

conceptSetName	A string name for the ConceptSet to be checked.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Check if a string name already exists in the WebApi as a ConceptSet definition name.

Value

If found, the function will return a tibble with details of the specification. If not found, FALSE will be returned.

Examples

```
## Not run:
existsConceptSetName(conceptSetName = "this text string needs to be checked",
                     baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

existsEstimationName *Check if Estimation definition name exists.*

Description

Check if Estimation definition name exists.

Usage

```
existsEstimationName(estimationName, baseUrl)
```

Arguments

`estimationName`	A string name for the Estimation to be checked.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Check if a string name already exists in the WebApi as a Estimation definition name.

Value

If found, the function will return a tibble with details of the specification. If not found, FALSE will be returned.

Examples

```
## Not run:
existsEstimationName(estimationName = "this text string needs to be checked",
                     baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

existsIncidenceRateName *Check if IncidenceRate definition name exists.*

Description

Check if IncidenceRate definition name exists.

Usage

```
existsIncidenceRateName(incidenceRateName, baseUrl)
```

Arguments

`incidenceRateName`	A string name for the IncidenceRate to be checked.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Check if a string name already exists in the WebApi as a IncidenceRate definition name.

Value

If found, the function will return a tibble with details of the specification. If not found, FALSE will be returned.

Examples

```
## Not run:
existsIncidenceRateName(incidenceRateName = "this text string needs to be checked",
                        baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

existsPathwayName *Check if Pathway definition name exists.*

Description

Check if Pathway definition name exists.

Usage

```
existsPathwayName(pathwayName, baseUrl)
```

Arguments

pathwayName	A string name for the Pathway to be checked.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Check if a string name already exists in the WebApi as a Pathway definition name.

Value

If found, the function will return a tibble with details of the specification. If not found, FALSE will be returned.

Examples

```
## Not run:
existsPathwayName(pathwayName = "this text string needs to be checked",
                  baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

existsPredictionName *Check if Prediction definition name exists.*

Description

Check if Prediction definition name exists.

Usage

```
existsPredictionName(predictionName, baseUrl)
```

Arguments

predictionName	A string name for the Prediction to be checked.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Check if a string name already exists in the WebApi as a Prediction definition name.

Value

If found, the function will return a tibble with details of the specification. If not found, FALSE will be returned.

Examples

```
## Not run:
existsPredictionName(predictionName = "this text string needs to be checked",
                     baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

exportCohortDefinitionSet *Export cohort definition set from WebAPI*

Description

Export cohort definition set from WebAPI

Usage

```
exportCohortDefinitionSet(baseUrl, cohortIds, generateStats = FALSE)
```

Arguments

baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
cohortIds	A set of cohortIds to fetch from WebApi.
generateStats	Should cohort inclusion rule statistics be generated?

Details

Constructs a CohortDefinition set containing the following fields:

atlasId The cohort ID in ATLAS.

cohortId a copy of the value in atlasId.

cohortName The name of the cohort.

sql The cohort generation sql.

json The cohort definition JSON.

logicDescription The cohort description.

`getCdmSources` *Get the data sources in the WebAPI instance*

Description

Get the data sources in the WebAPI instance

Usage

```
getCdmSources(baseUrl)
```

Arguments

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

Details

Obtains the data sources configured in the WebAPI instance.

Value

A data frame.

`getCharacterizationDefinition` *Get Characterization id definition.*

Description

Get Characterization id definition.

Usage

```
getCharacterizationDefinition(characterizationId, baseUrl)
```

Arguments

characterizationId	An integer id representing the id that uniquely identifies a Characterization definition in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Obtain the Characterization definition from WebAPI for a given Characterization id

Value

An R object representing the Characterization definition

Examples

```
## Not run:
getCharacterizationDefinition(characterizationId = 13242,
                              baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getCharacterizationDefinitionsMetaData

Get the meta data for Characterization definitions.

Description

Get the meta data for Characterization definitions.

Usage

```
getCharacterizationDefinitionsMetaData(baseUrl)
```

Arguments

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

Details

Get the meta data of WebApi specifications such as id, name, created/modified details, hash object, etc. from WebApi for Characterization. This function is useful to retrieve the current Characterization specifications.

Value

A tibble of specification metadata for Characterization. Note: modifiedDate and createdDate are returned as text/character.

Examples

```
## Not run:
getCharacterizationDefinitionsMetaData(baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getCharacterizationGenerationInformation

Get generation information for Characterization id.

Description

Get generation information for Characterization id.

Usage

```
getCharacterizationGenerationInformation(characterizationId, baseUrl)
```

Arguments

`characterizationId`
An integer id representing the id that uniquely identifies a Characterization definition in a WebApi instance.

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

Details

Get generation (execution) information about Characterization for a characterizationId.

Value

An R object representing the Characterization definition

Examples

```
## Not run:
getCharacterizationGenerationInformation(characterizationId = 13242,
                                         baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getCharacterizationResults *Get results for a Characterization Id.*

Description

Get results for a Characterization Id.

Usage

```
getCharacterizationResults(characterizationId, baseUrl)
```

Arguments

`characterizationId`
: An integer id representing the id that uniquely identifies a characterization analysis definition in a WebApi instance.

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

Details

Get the results for Characterization id.

Value

An R object with results.

Examples

```
## Not run:
getCharacterizationResults(characterizationId = 342, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getCohortDefinition *Get Cohort id definition.*

Description

Get Cohort id definition.

Usage

```
getCohortDefinition(cohortId, baseUrl)
```

Arguments

`cohortId`
: An integer id representing the id that uniquely identifies a Cohort definition in a WebApi instance.

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

Details

Obtain the Cohort definition from WebAPI for a given Cohort id

Value

An R object representing the Cohort definition

Examples

```
## Not run:
getCohortDefinition(cohortId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getCohortDefinitionExpression *(Deprecated) Get a cohort definition expression*

Description

(Deprecated) Get a cohort definition expression

Usage

```
getCohortDefinitionExpression(cohortId, baseUrl)
```

Arguments

`cohortId`	An integer id representing the id that uniquely identifies a cohort definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

(Deprecated) Obtain the JSON expression from WebAPI for a given cohort id

Value

A JSON list object representing the cohort definition This function has been deprecated. As an alternative please use the following steps as in the example below: 1) cohortDefinition <- getCohortDefinition(baseUrl = baseUrl, cohortId = 15873) 2) validJsonExpression <- RJSONIO::toJSON(cohortDefinition$expression)
3) save validJsonExpression object as .txt"

Examples

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

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

## End(Not run)
```

getCohortDefinitionName *(Deprecated) Get a cohort definition's name from WebAPI*

Description

(Deprecated) Get a cohort definition's name from WebAPI

Usage

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

Arguments

`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`cohortId`	An integer id representing the id that uniquely identifies a cohort definition in a WebApi instance.
`formatName`	Should the name be formatted to remove prefixes and underscores?

Details

(Deprecated) Obtains the name of a cohort. This function has been deprecated. As an alternative please use getCohortDefinition

Value

The name of the cohort.

getCohortDefinitionsMetaData *Get the meta data for Cohort definitions.*

Description

Get the meta data for Cohort definitions.

Usage

```
getCohortDefinitionsMetaData(baseUrl)
```

Arguments

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

Details

Get the meta data of WebApi specifications such as id, name, created/modified details, hash object, etc. from WebApi for Cohort. This function is useful to retrieve the current Cohort specifications.

Value

A tibble of specification metadata for Cohort. Note: modifiedDate and createdDate are returned as text/character.

Examples

```
## Not run:
getCohortDefinitionsMetaData(baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

getCohortDefinitionSql *Get a cohort definition's SQL from WebAPI*

Description

Get a cohort definition's SQL from WebAPI

Usage

```
getCohortDefinitionSql(cohortId, baseUrl, generateStats = TRUE)
```

Arguments

`cohortId`	An integer id representing the id that uniquely identifies a cohort definition in a WebApi instance.
`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. By default this is TRUE.

Details

Obtains the template SQL of a cohort. 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.

Value

The templated SQL to generate the cohort

getCohortGenerationInformation

Get generation information for Cohort id.

Description

Get generation information for Cohort id.

Usage

```
getCohortGenerationInformation(cohortId, baseUrl)
```

Arguments

`cohortId`	An integer id representing the id that uniquely identifies a Cohort definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Get generation (execution) information about Cohort for a cohortId.

Value

An R object representing the Cohort definition

Examples

```
## Not run:
getCohortGenerationInformation(cohortId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getCohortInclusionRulesAndCounts

Get cohort inclusion rules and person counts

Description

Get cohort inclusion rules and person counts

Usage

```
getCohortInclusionRulesAndCounts(baseUrl, cohortId, sourceKey)
```

Arguments

`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`cohortId`	An integer id representing the id that uniquely identifies a cohort definition in a WebApi instance.
`sourceKey`	The source key for a CDM instance in WebAPI, as defined in the Configuration page

Details

Obtains the inclusion rules from a cohort definition and summarizes the person counts per rule

---

getCohortResults *Get results for a Cohort Id.*

Description

Get results for a Cohort Id.

Usage

```
getCohortResults(cohortId, baseUrl)
```

Arguments

cohortId	An integer id representing the id that uniquely identifies a cohort definition in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Get the results for Cohort id.

Value

An R object with results.

Examples

```
## Not run:
getCohortResults(cohortId = 342, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

getCohortSql *Get SQL query for Cohort definition.*

Description

Get SQL query for Cohort definition.

Usage

```
getCohortSql(cohortDefinition, baseUrl, generateStats = TRUE)
```

Arguments

`cohortDefinition`
: An R list object (not JSON) representing the Cohort definition. It is the output R expression object of list object from `CohortDefinition`

`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. By default this is TRUE.

Details

Given a valid Cohort definition R-object (not JSON) this function will return the parameterized SQL in OHDSI SQL dialect. This SQL may be used along with OHDSI R-package 'SQLRender' to render/translate to target SQL dialect and parameters rendered.

Value

An R object containing the SQL for Cohort definition.

Examples

```
## Not run:
getCohortSql(CohortDefinition = (getCohortDefinition(cohortId = 13242, baseUrl = baseUrl)),
            baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getConcepts *Get concepts*

Description

Get concepts

Usage

```
getConcepts(
  conceptIds,
  baseUrl,
  vocabularySourceKey = NULL,
  snakeCaseToCamelCase = TRUE
)
```

Arguments

`conceptIds`
: A vector of concept IDs.

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

`vocabularySourceKey`
: The source key of the Vocabulary. By default, the priority Vocabulary is used.

`snakeCaseToCamelCase`
: Should the column names of the result be converted to camelCase?

Value

A tibble containing information on the concepts.

Examples

```
## Not run:
conceptSet <- getConceptSet(conceptSetId = 282, baseUrl = "http://server.org:80/WebAPI")
conceptIds <- resolveConceptSet(conceptSet = conceptSet, baseUrl = "http://server.org:80/WebAPI")
concepts <- getConcepts(conceptIds = conceptIds, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

getConceptSetDefinition

Get ConceptSet id definition.

Description

Get ConceptSet id definition.

Usage

```
getConceptSetDefinition(conceptSetId, baseUrl)
```

Arguments

`conceptSetId`	An integer id representing the id that uniquely identifies a ConceptSet definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Obtain the ConceptSet definition from WebAPI for a given ConceptSet id

Value

An R object representing the ConceptSet definition

Examples

```
## Not run:
getConceptSetDefinition(conceptSetId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getConceptSetDefinitionBySourceKey

Fetch concept set definition from WebAPI by SourceKey

Description

Fetch concept set definition from WebAPI by SourceKey

Usage

```
getConceptSetDefinitionBySourceKey(
  conceptSetId,
  baseUrl,
  vocabularySourceKey = NULL
)
```

Arguments

conceptSetId	the id of the concept set to retrieve.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
vocabularySourceKey	The source key of the Vocabulary. By default, the priority Vocabulary is used.

Details

Fetches a concept set definition from WebAPI by SourceKey. If SourceKey is not specified, the priority vocabulary will be used.

Value

An R object representing the ConceptSet definition

Examples

```
## Not run:
conceptSetDefinition <- getConceptSetDefinitionBySourceKey(conceptSetId = 282,
                                                baseUrl = "http://server.org:80/WebAPI",
                                                vocabularySourceKey = "MY_VOCAB")
conceptIds <- resolveConceptSet(conceptSetDefinition = conceptSetDefinition,
                                baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getConceptSetDefinitionsMetaData

Get the meta data for ConceptSet definitions.

Description

Get the meta data for ConceptSet definitions.

Usage

```
getConceptSetDefinitionsMetaData(baseUrl)
```

Arguments

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

Details

Get the meta data of WebApi specifications such as id, name, created/modified details, hash object, etc. from WebApi for ConceptSet. This function is useful to retrieve the current ConceptSet specifications.

Value

A tibble of specification metadata for ConceptSet. Note: modifiedDate and createdDate are returned as text/character.

Examples

```
## Not run:
getConceptSetDefinitionsMetaData(baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getDefinition

Get the definition for an id of chosen category in WebApi.

Description

Get the definition for an id of chosen category in WebApi.

Usage

```
getDefinition(id, baseUrl, category)
```

Arguments

`id`	An integer id representing the id that uniquely identifies a definition for the category in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`category`	These are the categories in WebApi. The valid string options are 'conceptSet', 'cohort', 'characterization', 'pathway, 'incidenceRate', 'estimation', 'prediction'.

Details

Get the definition for an id of chosen category in WebApi. The return object will be a R representation of the definition, that may be reconverted to JSON.

Value

An R object representing the definition

Examples

```
## Not run:
getDefinition(id = 13242, category = "cohort", baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

`getDefinitionsMetadata` *Retrieve the meta data for WebApi definitions of a certain category*

Description

Retrieve the meta data for WebApi definitions of a certain category

Usage

```
getDefinitionsMetadata(baseUrl, category)
```

Arguments

`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`category`	These are the categories in WebApi. The valid string options are 'conceptSet', 'cohort', 'characterization', 'pathway, 'incidenceRate', 'estimation', 'prediction'.

Details

Obtains the meta data of WebApi specifications such as id, name, created/modified details, hash object, etc for a certain category. The following function categories are supported. Concept-set, Cohort-definition, Cohort-characterization, Pathway-analysis, Incidence rate (ir), estimation and prediction. This function is useful to retrieve the current specifications.

Value

A tibble of specification metadata.

Examples

```
## Not run:
getDefinitionsMetadata(baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

getEstimationDefinition *Get Estimation id definition.*

Description

Get Estimation id definition.

Usage

```
getEstimationDefinition(estimationId, baseUrl)
```

Arguments

estimationId	An integer id representing the id that uniquely identifies a Estimation definition in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Obtain the Estimation definition from WebAPI for a given Estimation id

Value

An R object representing the Estimation definition

Examples

```
## Not run:
getEstimationDefinition(estimationId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getEstimationDefinitionsMetaData

Get the meta data for Estimation definitions.

Description

Get the meta data for Estimation definitions.

Usage

```
getEstimationDefinitionsMetaData(baseUrl)
```

Arguments

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

Details

Get the meta data of WebApi specifications such as id, name, created/modified details, hash object, etc. from WebApi for Estimation. This function is useful to retrieve the current Estimation specifications.

Value

A tibble of specification metadata for Estimation. Note: modifiedDate and createdDate are returned as text/character.

Examples

```
## Not run:
getEstimationDefinitionsMetaData(baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getGenerationInformation

Get generation information.

Description

Get generation information.

Usage

```
getGenerationInformation(id, category, baseUrl)
```

Arguments

`id`	An integer id representing the id that uniquely identifies a definition for the category in a WebApi instance.
`category`	The category of expression in WebApi. Only the following strings are accepted: 'cohort', 'characterization', 'pathway', 'incidenceRate'.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Get generation (execution) information.

Value

An R object with the generation information.

Examples

```
## Not run:
getGenerationInformation(id = 13242,
                         category = "cohort",
                         baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

`getIncidenceRateDefinition` *Get IncidenceRate id definition.*

Description

Get IncidenceRate id definition.

Usage

```
getIncidenceRateDefinition(incidenceRateId, baseUrl)
```

Arguments

`incidenceRateId`	An integer id representing the id that uniquely identifies a IncidenceRate definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Obtain the IncidenceRate definition from WebAPI for a given IncidenceRate id

Value

An R object representing the IncidenceRate definition

Examples

```
## Not run:
getIncidenceRateDefinition(incidenceRateId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getIncidenceRateDefinitionsMetaData

Get the meta data for IncidenceRate definitions.

Description

Get the meta data for IncidenceRate definitions.

Usage

```
getIncidenceRateDefinitionsMetaData(baseUrl)
```

Arguments

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

Details

Get the meta data of WebApi specifications such as id, name, created/modified details, hash object, etc. from WebApi for IncidenceRate. This function is useful to retrieve the current IncidenceRate specifications.

Value

A tibble of specification metadata for IncidenceRate. Note: modifiedDate and createdDate are returned as text/character.

Examples

```
## Not run:
getIncidenceRateDefinitionsMetaData(baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getIncidenceRateGenerationInformation

Get generation information for IncidenceRate id.

Description

Get generation information for IncidenceRate id.

Usage

```
getIncidenceRateGenerationInformation(incidenceRateId, baseUrl)
```

Arguments

`incidenceRateId`	An integer id representing the id that uniquely identifies a IncidenceRate definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Get generation (execution) information about IncidenceRate for a incidenceRateId.

Value

An R object representing the IncidenceRate definition

Examples

```
## Not run:
getIncidenceRateGenerationInformation(incidenceRateId = 13242,
                                      baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getIncidenceRateResults

Get results for a IncidenceRate Id.

Description

Get results for a IncidenceRate Id.

Usage

```
getIncidenceRateResults(incidenceRateId, baseUrl)
```

Arguments

incidenceRateId
: An integer id representing the id that uniquely identifies a incidence rate analysis definition in a WebApi instance.

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

Details

Get the results for IncidenceRate id.

Value

An R object with results.

Examples

```
## Not run:
getIncidenceRateResults(incidenceRateId = 342, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getPathwayDefinition *Get Pathway id definition.*

Description

Get Pathway id definition.

Usage

```
getPathwayDefinition(pathwayId, baseUrl)
```

Arguments

pathwayId
: An integer id representing the id that uniquely identifies a Pathway definition in a WebApi instance.

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

Details

Obtain the Pathway definition from WebAPI for a given Pathway id

Value

An R object representing the Pathway definition

Examples

```
## Not run:
getPathwayDefinition(pathwayId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getPathwayDefinitionsMetaData

Get the meta data for Pathway definitions.

Description

Get the meta data for Pathway definitions.

Usage

```
getPathwayDefinitionsMetaData(baseUrl)
```

Arguments

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

Details

Get the meta data of WebApi specifications such as id, name, created/modified details, hash object, etc. from WebApi for Pathway. This function is useful to retrieve the current Pathway specifications.

Value

A tibble of specification metadata for Pathway. Note: modifiedDate and createdDate are returned as text/character.

Examples

```
## Not run:
getPathwayDefinitionsMetaData(baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getPathwayGenerationInformation

Get generation information for Pathway id.

Description

Get generation information for Pathway id.

Usage

```
getPathwayGenerationInformation(pathwayId, baseUrl)
```

Arguments

pathwayId	An integer id representing the id that uniquely identifies a Pathway definition in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Get generation (execution) information about Pathway for a pathwayId.

Value

An R object representing the Pathway definition

Examples

```
## Not run:
getPathwayGenerationInformation(pathwayId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getPathwayResults *Get results for a Pathway Id.*

Description

Get results for a Pathway Id.

Usage

```
getPathwayResults(pathwayId, baseUrl)
```

Arguments

pathwayId	An integer id representing the id that uniquely identifies a pathway analysis definition in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Get the results for Pathway id.

Value

An R object with results.

Examples

```
## Not run:
getPathwayResults(pathwayId = 342, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getPersonProfile *Get person profile data*

Description

Get person profile data

Usage

```
getPersonProfile(baseUrl, sourceKey, personId, indexCohortId = NULL)
```

Arguments

baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
sourceKey	The source key for a CDM instance in WebAPI, as defined in the Configuration page
personId	The personId of the person whose profile is being reviewed
indexCohortId	<OPTIONAL> Do you want to use a particular cohortId as the index cohort? If left NULL, the WebApi will identify the earliest cohort for the person by cohort start date and use it as the index cohort. WebApi uses the cohort start date of the index cohort to calculate the person's index age (ageAtIndex). WebApi will also return the relative position, in days, for each event compared to the index cohorts start date. These relative positions are useful to study the relationship of various events with respect to the index cohort start date.

Details

Get a R object with person profile data. This function may be used for visualizing a patients profile in tables or visualization.

Value

A list of tibble data frame objects corresponding to cohorts, observationPeriod, records and person.

Examples

```
## Not run:
getPersonProfile(baseUrl = "http://server.org:80/WebAPI",
                 sourceKey = "sourceKey",
                 personId = 342342)

## End(Not run)
```

getPredictionDefinition

Get Prediction id definition.

Description

Get Prediction id definition.

Usage

```
getPredictionDefinition(predictionId, baseUrl)
```

Arguments

`predictionId`	An integer id representing the id that uniquely identifies a Prediction definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Obtain the Prediction definition from WebAPI for a given Prediction id

Value

An R object representing the Prediction definition

Examples

```
## Not run:
getPredictionDefinition(predictionId = 13242, baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getPredictionDefinitionsMetaData

Get the meta data for Prediction definitions.

Description

Get the meta data for Prediction definitions.

Usage

```
getPredictionDefinitionsMetaData(baseUrl)
```

Arguments

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

Details

Get the meta data of WebApi specifications such as id, name, created/modified details, hash object, etc. from WebApi for Prediction. This function is useful to retrieve the current Prediction specifications.

Value

A tibble of specification metadata for Prediction. Note: modifiedDate and createdDate are returned as text/character.

Examples

```
## Not run:
getPredictionDefinitionsMetaData(baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getPriorityVocabularyKey

Get Priority Vocabulary Source Key

Description

Get Priority Vocabulary Source Key

Usage

```
getPriorityVocabularyKey(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 Vocabulary in WebApi.

Value

A string.

getResults *Get generation results*

Description

Get generation results

Usage

```
getResults(id, baseUrl, category)
```

Arguments

id	An integer id representing the id that uniquely identifies a definition for the category in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
category	These are the categories in WebApi. The valid string options are ’conceptSet’, ’cohort’, ’characterization’, ’pathway, ’incidenceRate’, ’estimation’, ’prediction’.

Details

Get the results objects from WebApi for a definition of a certain category in WebApi.

Value

Returns the result objects for a given id and category from the WebApi.

Examples

```
## Not run:
getResults(id = 282, category = "cohort", baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

getSourceConcepts *Get source concepts that map to standard concepts*

Description

Get source concepts that map to standard concepts

Usage

```
getSourceConcepts(
  conceptIds,
  baseUrl,
  vocabularySourceKey = NULL,
  snakeCaseToCamelCase = TRUE
)
```

Arguments

conceptIds — A list of concept IDs referring to standard concepts.

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

vocabularySourceKey — The source key of the Vocabulary. By default, the priority Vocabulary is used.

snakeCaseToCamelCase — Should the column names of the result be converted to camelCase?

Value

A tibble containing information on the source concepts.

Examples

```
## Not run:
conceptSet <- getConceptSetDefinition(conceptSetId = 282,
                                      baseUrl = "http://server.org:80/WebAPI")
conceptIds <- resolveConceptSet(conceptSet = conceptSet, baseUrl = "http://server.org:80/WebAPI")
sourceConcepts <- getSourceConcepts(conceptIds = conceptIds,
                                    baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

getWebApiVersion *Get the WebAPI version number*

Description

Get the WebAPI version number

Usage

```
getWebApiVersion(baseUrl)
```

Arguments

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

Details

Obtains the WebAPI version number. This function is used to check that WebAPI baseUrl can be accessed and is a good first check to make sure you can access a WebAPI endpoint.

Value

The WebApi versions as a string.

Examples

```
## Not run:
getWebApiVersion("http://server.org:80/WebAPI")

## 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(
  cohortId,
  name = NULL,
  jsonFolder = "inst/cohorts",
  sqlFolder = "inst/sql/sql_server",
  baseUrl,
  generateStats = FALSE
)
```

Arguments

cohortId	An integer id representing the id that uniquely identifies a cohort definition in a WebApi instance.
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.
jsonFolder	Path to the folder where the JSON representation will be saved.
sqlFolder	Path to the folder where the SQL representation will be saved.
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 a folder (defaults to 'the inst/cohorts' folder), and fetch the template SQL and store it in another folder (defaults to 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. Note: generate inclusion statistics are created for all by default.

Examples

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

insertCohortDefinitionInPackage(cohortId = 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 = "inst/settings/CohortsToCreate.csv",
  baseUrl,
  jsonFolder = "inst/cohorts",
  sqlFolder = "inst/sql/sql_server",
  rFileName = "R/CreateCohorts.R",
  insertTableSql = TRUE,
  insertCohortCreationR = TRUE,
  generateStats = FALSE,
  packageName
)
```

Arguments

fileName Name of a CSV file 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".

jsonFolder Path to the folder where the JSON representations will be saved.

sqlFolder Path to the folder where the SQL representations will be saved.

rFileName Name of R file to generate when `insertCohortCreationR = TRUE`.

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).

`invokeCharacterizationGeneration`

Invoke generation of Characterization id.

Description

Invoke generation of Characterization id.

Usage

```
invokeCharacterizationGeneration(characterizationId, baseUrl, sourceKey)
```

Arguments

`characterizationId`	An integer id representing the id that uniquely identifies a Characterization definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`sourceKey`	The source key for a CDM instance in WebAPI, as defined in the Configuration page

Details

Invoke the generation of Characterization id in the WebApi.

Value

A tibble with job status information.

Examples

```
## Not run:
invokeCharacterizationGeneration(characterizationId = 13242,
                                 baseUrl = "http://server.org:80/WebAPI",
                                 sourceKey = "HCUP")

## End(Not run)
```

invokeCohortGeneration *Invoke generation of Cohort id.*

Description

Invoke generation of Cohort id.

Usage

```
invokeCohortGeneration(cohortId, baseUrl, sourceKey)
```

Arguments

cohortId	An integer id representing the id that uniquely identifies a Cohort definition in a WebApi instance.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
sourceKey	The source key for a CDM instance in WebAPI, as defined in the Configuration page

Details

Invoke the generation of Cohort id in the WebApi.

Value

A tibble with job status information.

Examples

```
## Not run:
invokeCohortGeneration(cohortId = 13242,
                       baseUrl = "http://server.org:80/WebAPI",
                       sourceKey = "HCUP")

## End(Not run)
```

invokeGeneration *Invoke generation.*

Description

Invoke generation.

Usage

```
invokeGeneration(id, baseUrl, sourceKey, category)
```

Arguments

`id`	An integer id representing the id that uniquely identifies a definition for the category in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`sourceKey`	The source key for a CDM instance in WebAPI, as defined in the Configuration page
`category`	These are the categories in WebApi. The valid string options are 'conceptSet', 'cohort', 'characterization', 'pathway, 'incidenceRate', 'estimation', 'prediction'.

Details

Invoke generation (execution) information.

Value

A dataframe with generation information such as status, jobName, and time.

Examples

```
## Not run:
invokeGeneration(id = 13242, category = "cohort", baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

`invokeIncidenceRateGeneration`

Invoke generation of IncidenceRate id.

Description

Invoke generation of IncidenceRate id.

Usage

```
invokeIncidenceRateGeneration(incidenceRateId, baseUrl, sourceKey)
```

Arguments

`incidenceRateId`	An integer id representing the id that uniquely identifies a IncidenceRate definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`sourceKey`	The source key for a CDM instance in WebAPI, as defined in the Configuration page

Details

Invoke the generation of IncidenceRate id in the WebApi.

Value

A tibble with job status information.

Examples

```
## Not run:
invokeIncidenceRateGeneration(incidenceRateId = 13242,
                              baseUrl = "http://server.org:80/WebAPI",
                              sourceKey = "HCUP")

## End(Not run)
```

`invokePathwayGeneration`

Invoke generation of Pathway id.

Description

Invoke generation of Pathway id.

Usage

```
invokePathwayGeneration(pathwayId, baseUrl, sourceKey)
```

Arguments

`pathwayId`	An integer id representing the id that uniquely identifies a Pathway definition in a WebApi instance.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`sourceKey`	The source key for a CDM instance in WebAPI, as defined in the Configuration page

Details

Invoke the generation of Pathway id in the WebApi.

Value

A tibble with job status information.

Examples

```
## Not run:
invokePathwayGeneration(pathwayId = 13242,
                        baseUrl = "http://server.org:80/WebAPI",
                        sourceKey = "HCUP")

## End(Not run)
```

isValidCharacterizationId *is Characterization id a valid definition in the WebApi.*

Description

is Characterization id a valid definition in the WebApi.

Usage

```
isValidCharacterizationId(characterizationIds, baseUrl)
```

Arguments

characterizationIds	A list of integer id(s) of the Characterization to be tested for validity.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Checks if a set of id for a Characterization is valid. The following checks are performed. 1) checks if all the ids exists in the WebApi i.e. valid.

Value

A logical vector indicating if an ID is valid.

Examples

```
## Not run:
isValidCharacterizationId(characterizationIds = c(13242, 3423, 34),
                          baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

isValidCohortId *is Cohort id a valid definition in the WebApi.*

Description

is Cohort id a valid definition in the WebApi.

Usage

```
isValidCohortId(cohortIds, baseUrl)
```

Arguments

cohortIds	A list of integer id(s) of the Cohort to be tested for validity.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Checks if a set of id for a Cohort is valid. The following checks are performed. 1) checks if all the ids exists in the WebApi i.e. valid.

Value

A logical vector indicating if an ID is valid.

Examples

```
## Not run:
isValidCohortId(cohortIds = c(13242, 3423, 34), baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

isValidConceptSetId *is ConceptSet id a valid definition in the WebApi.*

Description

is ConceptSet id a valid definition in the WebApi.

Usage

```
isValidConceptSetId(conceptSetIds, baseUrl)
```

Arguments

conceptSetIds	A list of integer id(s) of the ConceptSet to be tested for validity.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Checks if a set of id for a ConceptSet is valid. The following checks are performed. 1) checks if all the ids exists in the WebApi i.e. valid.

Value

A logical vector indicating if an ID is valid.

Examples

```
## Not run:
isValidConceptSetId(conceptSetIds = c(13242, 3423, 34), baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

isValidEstimationId *is Estimation id a valid definition in the WebApi.*

Description

is Estimation id a valid definition in the WebApi.

Usage

```
isValidEstimationId(estimationIds, baseUrl)
```

Arguments

`estimationIds`	A list of integer id(s) of the Estimation to be tested for validity.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Checks if a set of id for a Estimation is valid. The following checks are performed. 1) checks if all the ids exists in the WebApi i.e. valid.

Value

A logical vector indicating if an ID is valid.

Examples

```
## Not run:
isValidEstimationId(estimationIds = c(13242, 3423, 34), baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

isValidId *Check if an id is valid.*

Description

Check if an id is valid.

Usage

```
isValidId(ids, baseUrl, category)
```

Arguments

`ids`	A list of integer id(s) of the category to be tested for validity.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`category`	These are the categories in WebApi. The valid string options are 'conceptSet', 'cohort', 'characterization', 'pathway, 'incidenceRate', 'estimation', 'prediction'.

Details

Checks if a set of id for a category is valid, i.e. checks if all the ids exists in the WebApi i.e. valid.

Value

A logical vector indicating if an ID is valid.

Examples

```
## Not run:
isValidId(ids = c(13242, 3423, 34), baseUrl = "http://server.org:80/WebAPI", category = "cohort")

## End(Not run)
```

isValidIncidenceRateId

is IncidenceRate id a valid definition in the WebApi.

Description

is IncidenceRate id a valid definition in the WebApi.

Usage

```
isValidIncidenceRateId(incidenceRateIds, baseUrl)
```

Arguments

incidenceRateIds	A list of integer id(s) of the IncidenceRate to be tested for validity.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Checks if a set of id for a IncidenceRate is valid. The following checks are performed. 1) checks if all the ids exists in the WebApi i.e. valid.

Value

A logical vector indicating if an ID is valid.

Examples

```
## Not run:
isValidIncidenceRateId(incidenceRateIds = c(13242, 3423, 34),
                       baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

isValidPathwayId *is Pathway id a valid definition in the WebApi.*

Description

is Pathway id a valid definition in the WebApi.

Usage

```
isValidPathwayId(pathwayIds, baseUrl)
```

Arguments

pathwayIds	A list of integer id(s) of the Pathway to be tested for validity.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Checks if a set of id for a Pathway is valid. The following checks are performed. 1) checks if all the ids exists in the WebApi i.e. valid.

Value

A logical vector indicating if an ID is valid.

Examples

```
## Not run:
isValidPathwayId(pathwayIds = c(13242, 3423, 34), baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

isValidPredictionId *is Prediction id a valid definition in the WebApi.*

Description

is Prediction id a valid definition in the WebApi.

Usage

```
isValidPredictionId(predictionIds, baseUrl)
```

Arguments

predictionIds	A list of integer id(s) of the Prediction to be tested for validity.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Checks if a set of id for a Prediction is valid. The following checks are performed. 1) checks if all the ids exists in the WebApi i.e. valid.

Value

A logical vector indicating if an ID is valid.

Examples

```
## Not run:
isValidPredictionId(predictionIds = c(13242, 3423, 34), baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

isValidSourceKey *Check if source key is valid.*

Description

Check if source key is valid.

Usage

```
isValidSourceKey(sourceKeys, baseUrl)
```

Arguments

sourceKeys	The source key(s) for a CDM instance in WebAPI, as defined in the Configuration page.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Checks if a set of sourceKey(s) are valid, i.e. checks if all the sourceKey(s) exists in the WebApi i.e. valid.

Value

A logical vector indicating if an ID is valid.

Examples

```
## Not run:
isValidSourceKey(sourceKeys = c("HCUP", "CCA"),
                 baseUrl = "http://server.org:80/WebAPI",
                 category = "cohort")

## End(Not run)
```

postCharacterizationDefinition

Post Characterization definition.

Description

Post Characterization definition.

Usage

```
postCharacterizationDefinition(name, characterizationDefinition, baseUrl)
```

Arguments

name	A valid name for the definition. WebApi will use this name (if valid) as the name of the definition. WebApi checks for validity, such as uniqueness, absence of unacceptable character etc. An error might be thrown.
characterizationDefinition	An R list object containing the expression for the specification. This will be converted to JSON expression by function and posted into the WebApi. Note: only limited checks are performed in R to check the validity of this expression.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Post Characterization definition to WebAPI

Value

This function will return a dataframe object with one row describing the posted WebApi expression and its details. If unsuccessful a STOP message will be shown.

Examples

```
## Not run:
postCharacterizationDefinition(name = "new valid name",
                               characterizationDefinition = definition,
                               baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

postCohortDefinition *Post Cohort definition.*

Description

Post Cohort definition.

Usage

```
postCohortDefinition(name, cohortDefinition, baseUrl)
```

Arguments

name	A valid name for the definition. WebApi will use this name (if valid) as the name of the definition. WebApi checks for validity, such as uniqueness, absence of unacceptable character etc. An error might be thrown.
cohortDefinition	An R list object containing the expression for the specification. This will be converted to JSON expression by function and posted into the WebApi. Note: only limited checks are performed in R to check the validity of this expression.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Post Cohort definition to WebAPI

Value

This function will return a dataframe object with one row describing the posted WebApi expression and its details. If unsuccessful a STOP message will be shown.

Examples

```
## Not run:
postCohortDefinition(name = "new valid name",
                     cohortDefinition = definition,
                     baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

postConceptSetDefinition *Post ConceptSet definition.*

Description

Post ConceptSet definition.

Usage

```
postConceptSetDefinition(name, conceptSetDefinition, baseUrl)
```

Arguments

`name` A valid name for the definition. WebApi will use this name (if valid) as the name of the definition. WebApi checks for validity, such as uniqueness, absence of unacceptable character etc. An error might be thrown.

`conceptSetDefinition` An R list object containing the expression for the specification. This will be converted to JSON expression by function and posted into the WebApi. Note: only limited checks are performed in R to check the validity of this expression.

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

Details

Post ConceptSet definition to WebAPI

Value

This function will return a dataframe object with one row describing the posted WebApi expression and its details. If unsuccessful a STOP message will be shown.

Examples

```
## Not run:
postConceptSetDefinition(name = "new valid name",
                         conceptSetDefinition = definition,
                         baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

`postDefinition` *Post a definition into WebApi*

Description

Post a definition into WebApi

Usage

```
postDefinition(baseUrl, name, category, definition)
```

Arguments

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

`name` A valid name for the definition. WebApi will use this name (if valid) as the name of the definition. WebApi checks for validity, such as uniqueness, absence of unacceptable character etc. An error might be thrown.

`category` These are the categories in WebApi. The valid string options are 'conceptSet', 'cohort', 'characterization', 'pathway, 'incidenceRate', 'estimation', 'prediction'.

`definition` An R list object containing the expression for the specification. This will be converted to JSON expression by function and posted into the WebApi. Note: only limited checks are performed in R to check the validity of this expression.

Details

Post a definition into WebAPI. Currently only cohort and concept-set are supported.

Value

This function will return a dataframe object with one row describing the posted WebApi expression and its details. If unsuccessful a STOP message will be shown.

Examples

```
## Not run:
definition <- getCohortDefinition(baseUrl = baseUrl, cohortId = 15873)
postDefinition(name = "new name for expression in sdaddaddd",
               baseUrl = "http://server.org:80/WebAPI",
               expression = definition,
               category = "cohort")

## End(Not run)
```

---

postEstimationDefinition

Post Estimation definition.

Description

Post Estimation definition.

Usage

```
postEstimationDefinition(name, estimationDefinition, baseUrl)
```

Arguments

`name` A valid name for the definition. WebApi will use this name (if valid) as the name of the definition. WebApi checks for validity, such as uniqueness, absence of unacceptable character etc. An error might be thrown.

`estimationDefinition` An R list object containing the expression for the specification. This will be converted to JSON expression by function and posted into the WebApi. Note: only limited checks are performed in R to check the validity of this expression.

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

Details

Post Estimation definition to WebAPI

Value

This function will return a dataframe object with one row describing the posted WebApi expression and its details. If unsuccessful a STOP message will be shown.

Examples

```
## Not run:
postEstimationDefinition(name = "new valid name",
                         estimationDefinition = definition,
                         baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

---

postIncidenceRateDefinition

Post IncidenceRate definition.

Description

Post IncidenceRate definition.

Usage

```
postIncidenceRateDefinition(name, incidenceRateDefinition, baseUrl)
```

Arguments

`name` A valid name for the definition. WebApi will use this name (if valid) as the name of the definition. WebApi checks for validity, such as uniqueness, absence of unacceptable character etc. An error might be thrown.

`incidenceRateDefinition` An R list object containing the expression for the specification. This will be converted to JSON expression by function and posted into the WebApi. Note: only limited checks are performed in R to check the validity of this expression.

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

Details

Post IncidenceRate definition to WebAPI

Value

This function will return a dataframe object with one row describing the posted WebApi expression and its details. If unsuccessful a STOP message will be shown.

Examples

```
## Not run:
postIncidenceRateDefinition(name = "new valid name",
                            incidenceRateDefinition = definition,
                            baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

postPathwayDefinition *Post Pathway definition.*

Description

Post Pathway definition.

Usage

```
postPathwayDefinition(name, pathwayDefinition, baseUrl)
```

Arguments

`name`	A valid name for the definition. WebApi will use this name (if valid) as the name of the definition. WebApi checks for validity, such as uniqueness, absence of unacceptable character etc. An error might be thrown.
`pathwayDefinition`	An R list object containing the expression for the specification. This will be converted to JSON expression by function and posted into the WebApi. Note: only limited checks are performed in R to check the validity of this expression.
`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".

Details

Post Pathway definition to WebAPI

Value

This function will return a dataframe object with one row describing the posted WebApi expression and its details. If unsuccessful a STOP message will be shown.

Examples

```
## Not run:
postPathwayDefinition(name = "new valid name",
                      pathwayDefinition = definition,
                      baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

postPredictionDefinition *Post Prediction definition.*

Description

Post Prediction definition.

Usage

```
postPredictionDefinition(name, predictionDefinition, baseUrl)
```

Arguments

`name` A valid name for the definition. WebApi will use this name (if valid) as the name of the definition. WebApi checks for validity, such as uniqueness, absence of unacceptable character etc. An error might be thrown.

`predictionDefinition`
An R list object containing the expression for the specification. This will be converted to JSON expression by function and posted into the WebApi. Note: only limited checks are performed in R to check the validity of this expression.

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

Details

Post Prediction definition to WebAPI

Value

This function will return a dataframe object with one row describing the posted WebApi expression and its details. If unsuccessful a STOP message will be shown.

Examples

```
## Not run:
postPredictionDefinition(name = "new valid name",
                         predictionDefinition = definition,
                         baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

`resolveConceptSet` *Resolve a concept set to the included standard concept IDs*

Description

Resolve a concept set to the included standard concept IDs

Usage

```
resolveConceptSet(conceptSetDefinition, baseUrl, vocabularySourceKey = NULL)
```

Arguments

`conceptSetDefinition`
A concept set definition, for example as obtained through the `getConceptSetDefinition` function, or taken from a cohort definition.

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

`vocabularySourceKey`
The source key of the Vocabulary. By default, the priority Vocabulary is used.

Details

Resolve a concept set to the included standard concept IDs

Value

A vector of standard concept ids.

Examples

```
## Not run:
conceptSetDefinition <- getConceptSetDefinition(conceptSetId = 282,
                                                baseUrl = "http://server.org:80/WebAPI")
conceptIds <- resolveConceptSet(conceptSetDefinition = conceptSetDefinition,
                                baseUrl = "http://server.org:80/WebAPI")

## End(Not run)
```

setAuthHeader

Manually set the authorization http header for a WebAPI baseUrl In some cases the user may want to manually set the authorization header. An authHeader is associated with a particular baseUrl and added to to the header of all http requests sent to that url by ROhdsiWebApi.

Description

Manually set the authorization http header for a WebAPI baseUrl In some cases the user may want to manually set the authorization header. An authHeader is associated with a particular baseUrl and added to to the header of all http requests sent to that url by ROhdsiWebApi.

Usage

```
setAuthHeader(baseUrl, authHeader)
```

Arguments

`baseUrl`	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
`authHeader`	A character string containing a Bearer token that will be added to the header of all http requests sent to baseUrl. (e.g. "Bearer lxd9n2nsdsd2329km23mexjop02m23m23mmmsioxiis(

updateCohortDefinition

Update a Cohort definition.

Description

Update a Cohort definition.

Usage

```
updateCohortDefinition(cohortDefinition, baseUrl)
```

Arguments

`cohortDefinition`
: An R list object containing the expression for the specification. Must include id, name and expression. This will be converted to JSON expression by function and posted into the WebApi. The definition will be checked against the WebApi instance for errors

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

Details

Update a Cohort definition.

Examples

```
## Not run:
definition <- getCohortDefinition(id = 13242,
                                  baseUrl = "http://server.org:80/WebAPI",
                                  category = cohort)
definition$name <- "My new name for this"
updateCohort(cohortDefinition, baseUrl, category = "cohort")

## End(Not run)
```

`updateConceptSetDefinition` *Update a ConceptSet definition.*

Description

Update a ConceptSet definition.

Usage

```
updateConceptSetDefinition(conceptSetDefinition, baseUrl)
```

Arguments

`conceptSetDefinition`
: An R list object containing the expression for the specification. Must include id, name and expression. This will be converted to JSON expression by function and posted into the WebApi. The definition will be checked against the WebApi instance for errors

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

Details

Update a ConceptSet definition.

Examples

```
## Not run:
definition <- getConceptSetDefinition(id = 13242,
                                      baseUrl = "http://server.org:80/WebAPI",
                                      category = conceptSet)
definition$name <- "My new name for this"
updateConceptSet(conceptSetDefinition, baseUrl, category = "cohort")

## End(Not run)
```

updateDefinition *Update definition*

Description

Update definition

Usage

```
updateDefinition(definition, baseUrl, category)
```

Arguments

definition	An R list object containing the expression for the specification. This will be converted to JSON expression by function and posted into the WebApi.
baseUrl	The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
category	These are the categories in WebApi. The valid string options are 'conceptSet', 'cohort', 'characterization', 'pathway, 'incidenceRate', 'estimation', 'prediction'.

Details

Update a definition in WebAPI. Currently only cohorts are supported. Takes the definition as a parameter and converts it to json. This is the full definition (i.e. including name and id fields)

Examples

```
## Not run:
definition <- getDefinition(
  id = 13242,
  baseUrl = "http://server.org:80/WebAPI",
  category = "cohort"
)
definition$name <- "My new name for this"
updateDefinition(definition, baseUrl, category = "cohort")

## End(Not run)
```

Index