Package 'Ulysses'

December 31, 2025

Title Automate OHDSI Study Setup

Version 1.0.0

Description
Automates setup of OHDSI study and provides functions to assist on improving organization and communication of a study.

License Apache License (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports cli,
crayon,
fs,
gert,
gh,
glue,
keyring,
lifecycle,
lubridate,
magrittr,
purrr,
rlang,
rstudioapi,
scales,
snakecase,
usethis,
whisker,
withr,
yaml,
RJSONIO

Suggests knitr,
rmarkdown,
here,
testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

R topics documented:

`cloneUlysses` *Function to clone a Ulysses Repo into local*

Description

Function to clone a Ulysses Repo into local

Usage

```
cloneUlysses(gitRemoteUrl, repoFolder)
```

Arguments

`gitRemoteUrl`	the url of the git remote used to clone
`repoFolder`	the location where you wish to place the git repo

Value

invisible return. Clones the git remote into local file structure and infuses default folders in clone

defineLoadTable *Define the load table*

Description

Define the load table

Usage

```
defineLoadTable(atlasId, label, category, subCategory)
```

Arguments

label	a vector of naming labels to identify the atlas Ids, must be in order of atlas ids
category	a vector of categories to identify the atlas ids, must be in order of atlas ids
subCategory	a vector of sub-categories to identify the atlas ids, must be in order of atlas ids
atlas	a vector of atlas Ids to load

Value

a tibble where each row is an atlas asset with meta data

ExecutionSettings *ExecutionSettings*

Description

An R6 class to define an ExecutionSettings object

Active bindings

`cdmDatabaseSchema` the schema containing the OMOP CDM

`workDatabaseSchema` the schema containing the cohort table

`tempEmulationSchema` the schema needed for temp tables

`cohortTable` the table containing the cohorts

`cdmSourceName` the name of the source data of the cdm

Methods

Public methods:

- `ExecutionSettings$new()`
- `ExecutionSettings$getDbms()`
- `ExecutionSettings$connect()`
- `ExecutionSettings$disconnect()`
- `ExecutionSettings$getConnection()`
- `ExecutionSettings$clone()`

Method `new()`:

Usage:

```
ExecutionSettings$new(
  connectionDetails = NULL,
  connection = NULL,
  cdmDatabaseSchema = NULL,
  workDatabaseSchema = NULL,
  tempEmulationSchema = NULL,
  cohortTable = NULL,
  cdmSourceName = NULL
)
```

Arguments:

`connectionDetails` a connectionDetails object

`connection` a connection to a dbms

`cdmDatabaseSchema` The schema of the OMOP CDM database

`workDatabaseSchema` The schema to which results will be written

`tempEmulationSchema` Some database platforms like Oracle and Snowflake do not truly support temp tables. To emulate temp tables, provide a schema with write privileges where temp tables can be created.

`cohortTable` The name of the table where the cohort(s) are stored

`cdmSourceName` A human-readable name for the OMOP CDM source

Method `getDbms()`: extract the dbms dialect

Usage:

```
ExecutionSettings$getDbms()
```

Method `connect()`: connect to dbms

Usage:

```
ExecutionSettings$connect()
```

Method `disconnect()`: disconnect from dbms

Usage:

```
ExecutionSettings$disconnect()
```

Method `getConnection()`: retrieve the connection object

Usage:

```
ExecutionSettings$getConnection()
```

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
ExecutionSettings$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`importAtlasCohortsFromManifest` *Import Atlas Cohorts from the manifest*

Description

Import Atlas Cohorts from the manifest

Usage

```
importAtlasCohortsFromManifest(
  cohortManifest,
  atlasConnection,
  outputFolder = here::here("inputs/cohorts/json")
)
```

Arguments

`atlasConnection`	an WebApiConnection R6 class that holds the creds to connect to webapi
`outputFolder`	the location where the cohorts should be written
`conceptSetManifest`	the location of the cohort manifest

Value

invisible return but stores the cohort files into Ulysses

`importAtlasConceptSetsFromManifest` *Import Atlas Concept sets from the manifest*

Description

Import Atlas Concept sets from the manifest

Usage

```
importAtlasConceptSetsFromManifest(
  conceptSetManifest,
  atlasConnection,
  outputFolder = here::here("inputs/conceptSets/json")
)
```

Arguments

`conceptSetManifest`	the location of the concept set manifest
`atlasConnection`	an WebApiConnection R6 class that holds the creds to connect to webapi
`outputFolder`	the location where the concept sets should be written

Value

invisible return but stores the concept set files into Ulysses

---

`initializeManifest` *Initialize Manifests*

Description

Initialize Manifests

Usage

```
initializeManifest(manifestType, loadTable = NULL, overwrite = FALSE)
```

Arguments

`manifestType`	the type of manifest to initialize, either conceptSet or cohort
`loadTable`	a tibble made using defineLoadTable which prespecifies atlas ids to initialize the manifest
`overwrite`	toggle whether to overwrite existing manifests. Default is FALSE

Value

invisible return but initializes the manifest in its appropriate folder location

---

`launchUlyssesRemoteWithBitBucketDC`
Launch Ulysses Remote using Bitbucket Data Center

Description

Launch Ulysses Remote using Bitbucket Data Center

Usage

```
launchUlyssesRemoteWithBitBucketDC(repoName, hostUrl, httpToken, projectName)
```

Arguments

`repoName`	the name of the repository in project
`hostUrl`	the url hosting bitbucket data center
`httpToken`	the http access token configured to your bitbucket profile. To find the httpToken go to Manage Account > HTTP access tokens. It is recommended that you store the httpToken as an environment variable fo ease of use.
`projectName`	the name of the project in Bitbucket Data Center storing the repo

Value

invisible return but initializes the remote on BitbucketDC

launchUlyssesRepo *Function to Launch new Ulysses Repo*

Description

Function to Launch new Ulysses Repo

Usage

```
launchUlyssesRepo(ulyssesStudySettings, verbose = TRUE, openProject = FALSE)
```

Arguments

ulyssesStudySettings	UlyssesStudy R6 class with the ulysses study details to make
verbose	a toggle whether to print details of launch in console
openProject	a toggle whether to open the repo as a new R project

Value

invisible return. Creates the ulysses repo in the local file structure

makeExecOptions *Make ExecOptions for Ulysses*

Description

Make ExecOptions for Ulysses

Usage

```
makeExecOptions(
  dbms,
  workDatabaseSchema,
  tempEmulationSchema = NULL,
  dbConnectionBlocks
)
```

Arguments

dbms	specify the dbms used in the exec options
workDatabaseSchema	the name of the workDatabaseSchema as a character string, location in DB where user has write access
tempEmulationSchema	he name of the tempEmulationSchema as a character strings
dbConnectionBlocks	a list of DbConfigBlock R6 classes specifying the dbs to connect

Value

A ExecOptions R6 class with the execOptions

`makeStudyMeta` *Make Study Meta for Ulysses*

Description

Make Study Meta for Ulysses

Usage

```
makeStudyMeta(
  studyTitle,
  therapeuticArea,
  studyType,
  contributors,
  studyLinks = NULL,
  studyTags = NULL
)
```

Arguments

`studyTitle`	the title of the study as a character string
`therapeuticArea`	the TA as a character string
`studyType`	the study type (typically characterization)
`studyLinks`	a list of study links
`studyTags`	a list of study tags

Value

A StudyMeta R6 class with the study meta

`makeUlyssesStudySettings`
Make Ulysses Study Settings

Description

Make Ulysses Study Settings

Usage

```
makeUlyssesStudySettings(
  repoName,
  repoFolder,
  studyMeta,
  execOptions,
  gitRemote = NULL,
  renvLock = NULL
)
```

Arguments

repoName	the name of repo as a character string
repoFolder	the folder path where the repo is stored in local as a character string
studyMeta	a StudyMeta R6 class with the details describing the study
execOptions	a ExecOptions R6 class with the execution details needed for the study
gitRemote	a remote url used to clone and set remote git
renvLock	file path to a renvLock file

Value

A UlyssesStudy R6 class with the ulysses study details to make

---

populateManifest *Initialize Manifests*

Description

Initialize Manifests

Usage

```
populateManifest(manifestType, importFromAtlas = TRUE)
```

Arguments

manifestType	the type of manifest to initialize, either conceptSet or cohort
importFromAtlas	toggle whether to import content from atlas. Default is TRUE

Value

invisible return but populates the manifest in its appropriate folder location

setAtlasConnection *Set Atlas Connection*

Description

Set Atlas Connection

Usage

```
setAtlasConnection()
```

Value

an R6 class of WebApiConnection

setContributor *Set Ulysses Contributor*

Description

Set Ulysses Contributor

Usage

```
setContributor(name, email, role)
```

Arguments

name	the name of the contributor as a character string
email	the email of the contributor as a character string
role	the role of the contirbutor as a character string

Value

A ContributorLine R6 class with the contributor info

setDbConfigBlock *set the config block for a database*

Description

set the config block for a database

Usage

```
setDbConfigBlock(
  configBlockName,
  cdmDatabaseSchema,
  databaseName = NULL,
  databaseLabel = NULL
)
```

Arguments

configBlockName	the name of the config block
cdmDatabaseSchema	the cdmDatabaseSchema specified as a character string
databaseName	the name of the database, typically uses the db name and id. For example optum_dod_202501
databaseLabel	the labelling name of the database, typically a common name for a db. For example Optum DOD

Value

A StudyMeta R6 class with the study meta

templateAtlasCredentials *Template for setting Atlas Credentials*

Description

Template for setting Atlas Credentials

Usage

```
templateAtlasCredentials()
```

Value

no return; prints info to console

Index