Package 'Ulysses'

January 15, 2026

Title Automate OHDSI Study Setup

Version 1.0.3

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:

addGitRemoteToUlysses   *Function to add a Remote to Ulysses directory*

Description

This function adds a git remote to the Ulysses repo. If user adds a commit message it will add and commit files prior to adding and pushing to remote. This function will check to see if there are untracked files that need to be commited prior to adding remote.

Usage

```
addGitRemoteToUlysses(
  gitRemoteUrl,
  gitRemoteName = "origin",
  commitMessage = NULL
)
```

Arguments

gitRemoteUrl	a character string of a git remote url
gitRemoteName	The name of the remote, defaults to origin
commitMessage	a character string of a commit Message to use. if null then skips commit

---

buildStudyHub *Build Study Hub*

Description

Build Study Hub

Usage

```
buildStudyHub(projectPath = here::here(), previewHub = TRUE)
```

Arguments

projectPath   the path to the Ulysses repo, by default takes the path of the active R project

previewHub   toggle to preview the hub after it builds. Default is TRUE

Value

invisible return. Creates _site folder with html files to preview site

---

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

---

createExecutionSettings

Create an ExecutionSettings object and set its attributes

Description

Create an ExecutionSettings object and set its attributes

Usage

```
createExecutionSettings(
  connectionDetails,
  connection = NULL,
  cdmDatabaseSchema,
  workDatabaseSchema,
  tempEmulationSchema,
  cohortTable,
  cdmSourceName
)
```

Arguments

connectionDetails

A DatabaseConnector connectionDetails object (optional if connection is specified)

connection   A DatabaseConnector connection object (optional if connectionDetails is specified)

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

Value

An ExecutionSettings object

---

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

---

execStudyPipeline *Function to execute all study task in analysis folder on set of config-Block*

Description

Function to execute all study task in analysis folder on set of configBlock

Usage

```
execStudyPipeline(configBlock, env = rlang::caller_env())
```

Arguments

configBlock	name of one or multiple configBlock to use in the execution
env	the execution environment

---

execStudyTask   *Function to execute a study task in Ulysses*

Description

Function to execute a study task in Ulysses

Usage

```
execStudyTask(taskFile, configBlock, env = rlang::caller_env())
```

Arguments

taskFile	the name of the taskFile. Only use the base name
configBlock	the name of the configBlock to use in the execution
env	the execution environment

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,
  repoPath = here::here("inputs")
)
```

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
repoPath	the location of the repo specifying where to save the manifest files

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

---

makeTaskFile   *Function initializing an R file for an analysis task*

Description

Function initializing an R file for an analysis task

Usage

```
makeTaskFile(
  nameOfTask,
  author = NULL,
  description = NULL,
  projectPath = here::here(),
  openFile = TRUE
)
```

Arguments

nameOfTask	The name of the analysis task script
author	the name of the person authoring the file. Defaults to template text if NULL
description	a description of the analysis task. Defaults to template text if NULL
projectPath	the path to the project
openFile	toggle on whether the file should be opened

---

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,
  repoPath = here::here("inputs")
)
```

Arguments

manifestType	the type of manifest to initialize, either conceptSet or cohort
importFromAtlas	
	toggle whether to import content from atlas. Default is TRUE
repoPath	the location of the repo specifying where to save the manifest files

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,
  cohortTable,
  databaseName = NULL,
  databaseLabel = NULL
)
```

Arguments

configBlockName

        the name of the config block

cdmDatabaseSchema

        the cdmDatabaseSchema specified as a character string

cohortTable   a character string specifying the way you want to name your cohort table

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

---

syncUlyssesWork   *Function to sync local work in Ulysses to remote*

Description

Function to sync local work in Ulysses to remote

Usage

```
syncUlyssesWork(
  commitMessage,
  branch = gert::git_branch(),
  gitRemoteName = "origin"
)
```

Arguments

`commitMessage`	the commit message describing the work done since the last commit
`branch`	the name of the branch you want to commit changes. Defaults to the active branch
`gitRemoteName`	the name of the remote. Defaults to origin.

`templateAtlasCredentials`

Template for setting Atlas Credentials

Description

Template for setting Atlas Credentials

Usage

```
templateAtlasCredentials()
```

Value

no return; prints info to console

`updateStudyVersion`   *Function to update the study version*

Description

Function to update the study version

Usage

```
updateStudyVersion(versionNumber, projectPath = here::here())
```

Arguments

`versionNumber`	the semantive version number to set as the new project version: 1.0.0
`projectPath`	the path of the project, defaults to the directory of the active Ulysses project

| zipAndArchive | *Zip and Archive results from a study execution* |

Description

Zip and Archive results from a study execution

Usage

```
zipAndArchive(input)
```

Arguments

input   the type of files to zip and archive. There are three options exportMerge, ex-
               portPretty and site. exportMerge is the merged results in long format. The
               exportPretty are xlsx files with formatted output from the study. The site is the
               html files of the studyHub

Value

invisible return. Stores the input as a zip file in the exec/archive folder

Index