Package ‘OhdsiShinyModules’

January 19, 2024

Type Package

Title Repository of Shiny Modules for OHDSI Result Viewers

Version 2.1.1

Author Jenna Reps

Maintainer Jenna Reps <reps@ohdsi.org>

Description Install this package to access useful shiny modules for building shiny apps to explore results using the OHDSI tools .

License Apache License 2.0

Encoding UTF-8

LazyData true

Depends R (>= 3.3.0)

VignetteBuilder knitr

Language EN-US

Imports checkmate,
CirceR,
cowplot,
DatabaseConnector,
dplyr,
DT,
ggplot2,
gridExtra,
htmltools,
lubridate,
methods,
ParallelLogger,
plotly,
purrr,
reactable,
readr,
RJSONIO,
rlang,
rmarkdown,
scales,
shiny,
shinycssloaders,
shinydashboard,

shinyWidgets,
SqlRender,
stringi,
stringr,
tibble,
tidyr,
tidyselect,
tippy,
RColorBrewer,
markdown

Suggests kableExtra,
knitr,
ResultModelManager,
RSQLite,
testthat,
withr

Remotes ohdsi/CirceR,
ohdsi/ResultModelManager

RoxygenNote 7.2.3

R topics documented:

`aboutHelperFile` *The location of the about module helper file*

Description

The location of the about module helper file

Usage

```
aboutHelperFile()
```

Details

Returns the location of the about helper file

Value

string location of the about helper file

aboutServer *The module server for the shiny app home*

Description

The module server for the shiny app home

Usage

```
aboutServer(
  id = "homepage",
  connectionHandler = NULL,
  resultDatabaseSettings = NULL
)
```

Arguments

`id` the unique reference id for the module

`connectionHandler`
a connection to the database with the results

`resultDatabaseSettings`
a list containing the characterization result schema, dbms, tablePrefix, databaseTable and cgTablePrefix

Details

The user specifies the id for the module

Value

The server for the shiny app home

aboutViewer *The module viewer for the shiny app home*

Description

The module viewer for the shiny app home

Usage

```
aboutViewer(id = "homepage")
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the home page module

characterizationAggregateFeaturesServer

The module server for exploring aggregate features results

Description

The module server for exploring aggregate features results

Usage

```
characterizationAggregateFeaturesServer(
  id,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`connectionHandler`	the connection to the prediction result database
`resultDatabaseSettings`	a list containing the characterization result schema, dbms, tablePrefix, databaseTable and cgTablePrefix

Details

The user specifies the id for the module

Value

The server to the description aggregate features module

characterizationAggregateFeaturesViewer

The module viewer for exploring aggregate feature results

Description

The module viewer for exploring aggregate feature results

Usage

```
characterizationAggregateFeaturesViewer(id)
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the description aggregate feature module

---

`characterizationDechallengeRechallengeServer`

The module server for exploring Dechallenge Rechallenge results

---

Description

The module server for exploring Dechallenge Rechallenge results

Usage

```
characterizationDechallengeRechallengeServer(
  id,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id` the unique reference id for the module

`connectionHandler` the connection to the prediction result database

`resultDatabaseSettings` a list containing the characterization result schema, dbms, tablePrefix, databaseTable and cgTablePrefix

Details

The user specifies the id for the module

Value

The server to the Dechallenge Rechallenge module

characterizationDechallengeRechallengeViewer
The module viewer for exploring Dechallenge Rechallenge results

Description

The module viewer for exploring Dechallenge Rechallenge results

Usage

```
characterizationDechallengeRechallengeViewer(id)
```

Arguments

id	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the description Dechallenge Rechallenge module

characterizationHelperFile
The location of the characterization module helper file

Description

The location of the characterization module helper file

Usage

```
characterizationHelperFile()
```

Details

Returns the location of the characterization helper file

Value

string location of the characterization helper file

`characterizationIncidenceServer` *The module server for exploring incidence results*

Description

The module server for exploring incidence results

Usage

```
characterizationIncidenceServer(id, connectionHandler, resultDatabaseSettings)
```

Arguments

`id` the unique reference id for the module

`connectionHandler`
the connection to the prediction result database

`resultDatabaseSettings`
a list containing the characterization result schema, dbms, tablePrefix, databaseTable and cgTablePrefix

Details

The user specifies the id for the module

Value

The server to the prediction incidence module

`characterizationIncidenceViewer` *The module viewer for exploring incidence results*

Description

The module viewer for exploring incidence results

Usage

```
characterizationIncidenceViewer(id)
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the description incidence module

characterizationServer *The module server for exploring characterization studies*

Description

The module server for exploring characterization studies

Usage

```
characterizationServer(
  id,
  connectionHandler,
  resultDatabaseSettings = list(port = 1)
)
```

Arguments

`id` the unique reference id for the module

`connectionHandler` a connection to the database with the results

`resultDatabaseSettings` a list containing the characterization result schema, dbms, tablePrefix, databaseTable and cgTablePrefix

Details

The user specifies the id for the module

Value

The server for the characterization module

characterizationTableServer *The module server for exploring 1 or more cohorts features*

Description

The module server for exploring 1 or more cohorts features

Usage

```
characterizationTableServer(id, connectionHandler, resultDatabaseSettings)
```

Arguments

`id` the unique reference id for the module

`connectionHandler`
the connection to the prediction result database

`resultDatabaseSettings`
a list containing the characterization result schema, dbms, tablePrefix, databaseTable and cgTablePrefix

Details

The user specifies the id for the module

Value

The server to the cohorts features server

`characterizationTableViewer` *The module viewer for exploring 1 or more cohorts features*

Description

The module viewer for exploring 1 or more cohorts features

Usage

```
characterizationTableViewer(id)
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the description cohorts features

characterizationTimeToEventServer

The module server for exploring time to event results

Description

The module server for exploring time to event results

Usage

```
characterizationTimeToEventServer(
  id,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id` the unique reference id for the module

`connectionHandler` the connection to the prediction result database

`resultDatabaseSettings` a list containing the characterization result schema, dbms, tablePrefix, databaseTable and cgTablePrefix

Details

The user specifies the id for the module

Value

The server to the prediction time to event module

characterizationTimeToEventViewer

The module viewer for exploring time to event results

Description

The module viewer for exploring time to event results

Usage

```
characterizationTimeToEventViewer(id)
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the characterization time to event module

---

`characterizationViewer` *The module viewer for exploring characterization studies*

Description

The module viewer for exploring characterization studies

Usage

```
characterizationViewer(id = 1)
```

Arguments

`id`	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the characterization viewer module

`cohortCountsModule` *Shiny module for cohort counts*

Description

Shiny module for cohort counts. Displays reactable table of cohort counts

Usage

```
cohortCountsModule(
  id,
  dataSource,
  cohortTable = dataSource$cohortTable,
  databaseTable = dataSource$dbTable,
  selectedCohorts,
  selectedDatabaseIds,
  cohortIds
)
```

Arguments

`id` namespace id

`dataSource` Backend Data source (DatabaseConnection)

`cohortTable` data.frame of all cohorts

`databaseTable` data.frame of all databases

`selectedCohorts` shiny::reactive - should return cohorts selected or NULL

`selectedDatabaseIds` shiny::reactive - should return cohorts selected or NULL

`cohortIds` shiny::reactive - should return cohorts selected integers or NULL

`cohortCountsView` *Cohort Counts View*

Description

Shiny view for cohort counts module

Usage

```
cohortCountsView(id)
```

Arguments

`id` Namespace id

`cohortDefinitionsModule` *Cohort Definition module*

Description

cohort defintion conceptsets, json etc

Usage

```
cohortDefinitionsModule(
  id,
  dataSource,
  cohortDefinitions,
  cohortTable = dataSource$cohortTable,
  cohortCountTable = dataSource$cohortCountTable,
  databaseTable = dataSource$dbTable
)
```

Arguments

`id` Namespace id

`dataSource` DatabaseConnection

`cohortDefinitions` reactive of cohort definitions to display

`cohortTable` data.frame of cohorts, cohortId, cohortName

`cohortCountTable` data.frame of cohortCounts, cohortId, subjects records

`databaseTable` data.frame of databasese, databaseId, name

`cohortDefinitionsView` *Cohort Definitions View*

Description

Outputs cohort definitions

Usage

```
cohortDefinitionsView(id)
```

Arguments

`id` Namespace id for module

`cohortDiagCharacterizationView` *characterization*

Description

Use for customizing UI

Usage

```
cohortDiagCharacterizationView(id)
```

Arguments

`id` Namespace Id - use namespaced id ns("characterization") inside diagnosticsExplorer module

cohortDiagnosticsHelperFile *The location of the description module helper file*

Description

The location of the description module helper file

Usage

```
cohortDiagnosticsHelperFile()
```

Details

Returns the location of the description helper file

Value

string location of the description helper file

cohortDiagnosticsServer *Cohort Diagnostics Explorer main module*

Description

Cohort Diagnostics Explorer main module

Usage

```
cohortDiagnosticsServer(
  id,
  connectionHandler,
  resultDatabaseSettings,
  dataSource = NULL
)
```

Arguments

id	module Id
connectionHandler	ResultModelManager ConnectionHander instance
resultDatabaseSettings	results database settings
dataSource	dataSource optionally created with createCdDatabaseDataSource

`cohortDiagnosticsView` *View for cohort diagnostics module*

Description

View for cohort diagnostics module

Usage

```
cohortDiagnosticsView(id = "DiagnosticsExplorer")
```

Arguments

`id`	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the cohort diagnostics viewer module

`cohortGeneratorHelperFile` *The location of the cohort-generator module helper file*

Description

The location of the cohort-generator module helper file

Usage

```
cohortGeneratorHelperFile()
```

Details

Returns the location of the cohort-generator helper file

Value

string location of the cohort-generator helper file

`cohortGeneratorServer` *The module server for the main cohort generator module*

Description

The module server for the main cohort generator module

Usage

```
cohortGeneratorServer(id, connectionHandler, resultDatabaseSettings)
```

Arguments

`id`	the unique reference id for the module
`connectionHandler`	a connection to the database with the results
`resultDatabaseSettings`	a named list containing the cohort generator results database details (schema, table prefix)

Value

the cohort generator results viewer main module server

`cohortGeneratorViewer` *The viewer of the main cohort generator module*

Description

The viewer of the main cohort generator module

Usage

```
cohortGeneratorViewer(id)
```

Arguments

`id`	the unique reference id for the module

Value

The user interface to the cohort generator results viewer

cohortMethodAttritionServer
The module server for rendering the PLE attrition results

Description

The module server for rendering the PLE attrition results

Usage

```
cohortMethodAttritionServer(
  id,
  selectedRow,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`selectedRow`	the selected row from the main results table
`connectionHandler`	the connection to the PLE results database
`resultDatabaseSettings`	a list containing the result schema and prefixes

Value

the PLE attrition results content server

cohortMethodAttritionViewer
The module viewer for rendering the PLE attrition results

Description

The module viewer for rendering the PLE attrition results

Usage

```
cohortMethodAttritionViewer(id)
```

Arguments

`id`	the unique reference id for the module

Value

The user interface to the cohort method attrition

`cohortMethodCovariateBalanceServer` *The module server for rendering the covariate balance plot*

Description

The module server for rendering the covariate balance plot

Usage

```
cohortMethodCovariateBalanceServer(
  id,
  selectedRow,
  connectionHandler,
  resultDatabaseSettings,
  metaAnalysisDbIds = NULL
)
```

Arguments

`id` the unique reference id for the module

`selectedRow` the selected row from the main results table

`connectionHandler` the connection to the PLE results database

`resultDatabaseSettings` a list containing the result schema and prefixes

`metaAnalysisDbIds` metaAnalysisDbIds

Value

the PLE covariate balance content server

`cohortMethodCovariateBalanceViewer` *The module viewer for rendering the PLE covariate balance analysis*

Description

The module viewer for rendering the PLE covariate balance analysis

Usage

```
cohortMethodCovariateBalanceViewer(id)
```

Arguments

`id` the unique reference id for the module

Value

The user interface to the cohort method covariate balance results

`cohortMethodDiagnosticsSummaryServer` *The module server for rendering the PLE diagnostics summary*

Description

The module server for rendering the PLE diagnostics summary

Usage

```
cohortMethodDiagnosticsSummaryServer(
  id,
  connectionHandler,
  resultDatabaseSettings,
  inputSelected
)
```

Arguments

`id` the unique reference id for the module

`connectionHandler` the connection to the PLE results database

`resultDatabaseSettings` a list containing the result schema and prefixes

`inputSelected` The target id, comparator id, outcome id and analysis id selected by the user

Value

the PLE diagnostics summary results

`cohortMethodDiagnosticsSummaryViewer` *The module viewer for rendering the PLE diagnostics results*

Description

The module viewer for rendering the PLE diagnostics results

Usage

```
cohortMethodDiagnosticsSummaryViewer(id)
```

Arguments

`id` the unique reference id for the module

Value

The user interface to the cohort method diagnostics viewer

`cohortMethodHelperFile` *The location of the cohort method module helper file*

Description

The location of the cohort method module helper file

Usage

```
cohortMethodHelperFile()
```

Details

Returns the location of the cohort method helper file

Value

string location of the cohort method helper file

`cohortMethodKaplanMeierServer` *The module server for rendering the Kaplan Meier curve*

Description

The module server for rendering the Kaplan Meier curve

Usage

```
cohortMethodKaplanMeierServer(
  id,
  selectedRow,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`selectedRow`	the selected row from the main results table
`connectionHandler`	the connection to the PLE results database
`resultDatabaseSettings`	a list containing the result schema and prefixes

Value

the PLE Kaplain Meier content server

cohortMethodKaplanMeierViewer *The module viewer for rendering the PLE Kaplan Meier curve*

Description

The module viewer for rendering the PLE Kaplan Meier curve

Usage

```
cohortMethodKaplanMeierViewer(id)
```

Arguments

`id`	the unique reference id for the module

Value

The module viewer for Kaplan Meier objects

cohortMethodPopulationCharacteristicsServer *The module server for rendering the population characteristics*

Description

The module server for rendering the population characteristics

Usage

```
cohortMethodPopulationCharacteristicsServer(
  id,
  selectedRow,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`selectedRow`	the selected row from the main results table
`connectionHandler`	the connection to the PLE results database
`resultDatabaseSettings`	a list containing the result schema and prefixes

Value

the PLE population characteristics content server

`cohortMethodPopulationCharacteristicsViewer` *The module viewer for rendering the PLE population characteristics*

Description

The module viewer for rendering the PLE population characteristics

Usage

```
cohortMethodPopulationCharacteristicsViewer(id)
```

Arguments

`id`	the unique reference id for the module

Value

The user interface to the cohort method population characteristics objects

`cohortMethodPowerServer` *The module server for rendering the PLE power analysis results*

Description

The module server for rendering the PLE power analysis results

Usage

```
cohortMethodPowerServer(
  id,
  selectedRow,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`selectedRow`	the selected row from the main results table
`connectionHandler`	the connection to the PLE results database
`resultDatabaseSettings`	a list containing the result schema and prefixes

Value

the PLE systematic error power server

`cohortMethodPowerViewer` *The module viewer for rendering the PLE power analysis*

Description

The module viewer for rendering the PLE power analysis

Usage

```
cohortMethodPowerViewer(id)
```

Arguments

`id`	the unique reference id for the module

Value

The user interface to the cohort method power calculation results

`cohortMethodPropensityModelServer` *The module server for rendering the propensity score model*

Description

The module server for rendering the propensity score model

Usage

```
cohortMethodPropensityModelServer(
  id,
  selectedRow,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`selectedRow`	the selected row from the main results table
`connectionHandler`	the connection to the PLE results database
`resultDatabaseSettings`	a list containing the result schema and prefixes

Value

the PLE propensity score model

cohortMethodPropensityModelViewer

The module viewer for rendering the PLE propensity score model covariates/coefficients

Description

The module viewer for rendering the PLE propensity score model covariates/coefficients

Usage

```
cohortMethodPropensityModelViewer(id)
```

Arguments

id	the unique reference id for the module

Value

The user interface to the cohort method propensity score model covariates/coefficients

cohortMethodPropensityScoreDistServer

The module server for rendering a PLE propensity score distribution

Description

The module server for rendering a PLE propensity score distribution

Usage

```
cohortMethodPropensityScoreDistServer(
  id,
  selectedRow,
  connectionHandler,
  resultDatabaseSettings,
  metaAnalysisDbIds = F
)
```

Arguments

id	the unique reference id for the module
selectedRow	the selected row from the main results table
connectionHandler	the connection to the PLE results database
resultDatabaseSettings	a list containing the result schema and prefixes
metaAnalysisDbIds	metaAnalysisDbIds

Value

the PLE propensity score distribution content server

`cohortMethodPropensityScoreDistViewer`

The module viewer for rendering the propensity score distribution

---

Description

The module viewer for rendering the propensity score distribution

Usage

```
cohortMethodPropensityScoreDistViewer(id)
```

Arguments

`id`	the unique reference id for the module

Value

The user interface to the cohort method propensity score distribution

---

`cohortMethodResultSummaryServer`

The module server for rendering the PLE diagnostics summary

Description

The module server for rendering the PLE diagnostics summary

Usage

```
cohortMethodResultSummaryServer(
  id,
  connectionHandler,
  resultDatabaseSettings,
  inputSelected
)
```

Arguments

`id`	the unique reference id for the module
`connectionHandler`	the connection to the PLE results database
`resultDatabaseSettings`	a list containing the result schema and prefixes
`inputSelected`	The target id, comparator id, outcome id and analysis id selected by the user

Value

the PLE diagnostics summary results

cohortMethodResultSummaryViewer *The module viewer for rendering the cohort method results*

Description

The module viewer for rendering the cohort method results

Usage

```
cohortMethodResultSummaryViewer(id)
```

Arguments

`id`	the unique reference id for the module

Value

The user interface to the cohort method diagnostics viewer

cohortMethodServer *The module server for the main cohort method module*

Description

The module server for the main cohort method module

Usage

```
cohortMethodServer(id, connectionHandler, resultDatabaseSettings)
```

Arguments

`id`	the unique reference id for the module
`connectionHandler`	a connection to the database with the results
`resultDatabaseSettings`	a named list containing the PLE results database connection details

Value

the PLE results viewer main module server

cohortMethodSystematicErrorServer *The module server for rendering the systematic error objects*

Description

The module server for rendering the systematic error objects

Usage

```
cohortMethodSystematicErrorServer(
  id,
  selectedRow,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id` the unique reference id for the module

`selectedRow` the selected row from the main results table

`connectionHandler` the connection handler to the result databases

`resultDatabaseSettings` a list containing the result schema and prefixes

Value

the PLE systematic error content server

cohortMethodSystematicErrorViewer *The module viewer for rendering the PLE systematic error objects*

Description

The module viewer for rendering the PLE systematic error objects

Usage

```
cohortMethodSystematicErrorViewer(id)
```

Arguments

`id` the unique reference id for the module

Value

The user interface to the cohort method systematic error module

cohortMethodViewer *The viewer of the main cohort method module*

Description

The viewer of the main cohort method module

Usage

```
cohortMethodViewer(id)
```

Arguments

id	the unique reference id for the module

Value

The user interface to the cohort method results viewer

cohortOverlapView *Cohort Overlap View*

Description

Use for customizing UI

Usage

```
cohortOverlapView(id)
```

Arguments

id	Namespace Id - use namespaced id ns("cohortOverlap") inside diagnosticsExplorer module

compareCohortCharacterizationView *compare characterization view*

Description

Use for customizing UI

Usage

```
compareCohortCharacterizationView(
  id,
  title = "Compare cohort characterization"
)
```

Arguments

`id`	Namespace Id - use namespaced id ns("compareCohortCharacterization") inside diagnosticsExplorer module
`title`	Optional string title field

conceptsInDataSourceView

concepts In DataSource View

Description

Use for customizing UI

Usage

```
conceptsInDataSourceView(id)
```

Arguments

`id`	Namespace Id - use namespaced id ns("conceptsInDataSource") inside diagnosticsExplorer module

createCdDatabaseDataSource

Create a CD data source from a database

Description

use this to create an interface to cohort diagnostics results data NOTE: I think this would make a good R6 class for other objects in this package so you could query them outside of a shiny app. E.g. if you wanted to make a custom R markdown template

Usage

```
createCdDatabaseDataSource(
  connectionHandler,
  resultDatabaseSettings,
  dataModelSpecificationsPath = system.file("cohort-diagnostics-ref",
    "resultsDataModelSpecification.csv", package = utils::packageName()),
 dataMigrationsRef = system.file("cohort-diagnostics-ref", "migrations.csv", package =
    utils::packageName()),
  displayProgress = FALSE
)
```

Arguments

connectionHandler
: An instance of a ResultModelManager::connectionHander - manages a connection to a database.

resultDatabaseSettings
: a list containing the result schema and prefixes

dataModelSpecificationsPath
: The path to a file containing specifications for the data model used by the database.

dataMigrationsRef
: The path to a file listing all migrations for the data model that should have been applied

displayProgress
: display a progress messaage (can only be used inside a shiny reactive context)

Value

An object of class ‘CdDataSource‘.

---

createCustomColDefList *Creating a list of custom column definitions for use in reactables*

Description

Creating a list of custom column definitions for use in reactables

Usage

```
createCustomColDefList(
  rawColNames,
  niceColNames = NULL,
  tooltipText = NULL,
  case = NULL,
  customColDefOptions = NULL
)
```

Arguments

rawColNames
: The raw column names taken directly from the source data table that are to be overwritten in the reactable

niceColNames
: The formatted column names that will appear as-specified in the reactable

tooltipText
: The text to be displayed in a toolTip when hovering over the column in the reactable

case
: Optional argument to convert raw column names to snake or camel case. Defaults to NULL and preserves whatever raw column names are passed in

customColDefOptions
: A list of lists, where the inner lists are any custom options from reactable::colDef for each column

Value

A named list of reactable::colDef objects

`createLargeSqlQueryDt` *Create Large Sql Query Data Table*

Description

Construct an instance of a LargeDataTable R6 instance for use inside largeTableServer

This should pass a parameterized sql query that can be used to iteratively return data from a table rather than returning the entire object.

Usage

```
createLargeSqlQueryDt(
  connectionHandler = NULL,
  connectionDetails = NULL,
  baseQuery,
  countQuery = NULL
)
```

Arguments

`connectionHandler`	ResultModelManager connectionHandler instance
`connectionDetails`	DatabaseConnector connectionDetails instance
`baseQuery`	base sql query
`countQuery`	count query string (should match query). Can be auto generated with sub query (default) but this will likely result in slow results

---

`databaseInformationView` *database Information View*

Description

Use for customizing UI

Usage

```
databaseInformationView(id)
```

Arguments

`id`	Namespace Id - use namespaced id ns("databaseInformation") inside diagnosticsExplorer module

`dataDiagnosticDrillServer` *The module server for exploring prediction summary results*

Description

The module server for exploring prediction summary results

Usage

```
dataDiagnosticDrillServer(id, connectionHandler, resultDatabaseSettings)
```

Arguments

`id`	the unique reference id for the module
`connectionHandler`	the connection to the prediction result database
`resultDatabaseSettings`	a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the summary module

`dataDiagnosticDrillViewer` *The module viewer for exploring data-diagnostic results in more detail*

Description

The module viewer for exploring data-diagnostic results in more detail

Usage

```
dataDiagnosticDrillViewer(id)
```

Arguments

`id`	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the summary module

`dataDiagnosticHelperFile` *The location of the data-diagnostic module helper file*

Description

The location of the data-diagnostic module helper file

Usage

```
dataDiagnosticHelperFile()
```

Details

Returns the location of the data-diagnostic helper file

Value

string location of the data-diagnostic helper file

`dataDiagnosticServer` *The module server for exploring data-diagnostic*

Description

The module server for exploring data-diagnostic

Usage

```
dataDiagnosticServer(
  id = "dataDiag",
  connectionHandler,
  resultDatabaseSettings = list(port = 1)
)
```

Arguments

`id` the unique reference id for the module

`connectionHandler` a connection to the database with the results

`resultDatabaseSettings` a list containing the data-diagnostic result schema

Details

The user specifies the id for the module

Value

The server for the data-diagnostic module

dataDiagnosticSummaryServer

The module server for exploring prediction summary results

Description

The module server for exploring prediction summary results

Usage

```
dataDiagnosticSummaryServer(id, connectionHandler, resultDatabaseSettings)
```

Arguments

`id` the unique reference id for the module

`connectionHandler`
the connection to the prediction result database

`resultDatabaseSettings`
a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the summary module

dataDiagnosticSummaryViewer

The module viewer for exploring data-diagnostic summary results

Description

The module viewer for exploring data-diagnostic summary results

Usage

```
dataDiagnosticSummaryViewer(id)
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the summary module

`dataDiagnosticViewer` *The module viewer for exploring data-diagnostic*

Description

The module viewer for exploring data-diagnostic

Usage

```
dataDiagnosticViewer(id = "dataDiag")
```

Arguments

`id`	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the data-diagnostic viewer module

`datasourcesHelperFile` *Define the helper file for the module*

Description

Define the helper file for the module

Usage

```
datasourcesHelperFile()
```

Value

The helper html file for the datasources module

datasourcesServer — *The server function for the datasources module*

Description

The server function for the datasources module

Usage

```
datasourcesServer(id, connectionHandler, resultDatabaseSettings)
```

Arguments

`id` The unique id for the datasources server namespace

`connectionHandler` A connection to the database with the results

`resultDatabaseSettings` A named list containing the cohort generator results database details (schema, table prefix)

Value

The server for the datasources module

datasourcesViewer — *The viewer function for hte datasources module*

Description

The viewer function for hte datasources module

Usage

```
datasourcesViewer(id)
```

Arguments

`id` The unique id for the datasources viewer namespace

Value

The UI for the datasources module

`evidenceSynthesisHelperFile` *The location of the evidence synthesis module helper file*

Description

The location of the evidence synthesis module helper file

Usage

```
evidenceSynthesisHelperFile()
```

Details

Returns the location of the evidence synthesis helper file

Value

string location of the evidence synthesis helper file

`evidenceSynthesisServer` *The module server for exploring PatientLevelPrediction*

Description

The module server for exploring PatientLevelPrediction

Usage

```
evidenceSynthesisServer(
  id,
  connectionHandler,
  resultDatabaseSettings = list(port = 1)
)
```

Arguments

`id` the unique reference id for the module

`connectionHandler`
a connection to the database with the results

`resultDatabaseSettings`
a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server for the PatientLevelPrediction module

`evidenceSynthesisViewer` *The module viewer for exploring evidence-synthesis*

Description

The module viewer for exploring evidence-synthesis

Usage

```
evidenceSynthesisViewer(id = 1)
```

Arguments

`id`	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the evidence-synthesis viewer module

`getCirceRenderedExpression` *Returns list with circe generated documentation*

Description

Returns list with circe generated documentation

Usage

```
getCirceRenderedExpression(
  cohortDefinition,
  cohortName = "Cohort Definition",
  includeConceptSets = FALSE
)
```

Arguments

`cohortDefinition`	An R object (list) with a list representation of the cohort definition expression, that may be converted to a cohort expression JSON using RJSONIO::toJSON(x = cohortDefinition, digits = 23, pretty = TRUE)
`cohortName`	Name for the cohort definition
`includeConceptSets`	Do you want to inclued concept set in the documentation

Value

list object

`getEnabledCdReports` *Get enable cd reports from available data*

Description

Get enable cd reports from available data

Usage

```
getEnabledCdReports(dataSource)
```

Arguments

`dataSource` C

`getLogoImage` *The location of the OHDSI logo*

Description

The location of the OHDSI logo

Usage

```
getLogoImage()
```

Details

Returns the location of the OHDSI logo

Value

string location of the OHDSI logo

`incidenceRatesView` *incidence Rates View*

Description

Use for customizing UI

Usage

```
incidenceRatesView(id)
```

Arguments

`id` Namespace Id - use namespaced id ns("incidenceRates") inside diagnosticsExplorer module

`inclusionRulesView` *inclusion Rules View*

Description

Use for customizing UI

Usage

```
inclusionRulesView(id)
```

Arguments

`id`	Namespace Id - use namespaced id ns("inclusionRules") inside diagnosticsExplorer module

`indexEventBreakdownView` *Index event breakdown view*

Description

Use for customizing UI

Usage

```
indexEventBreakdownView(id)
```

Arguments

`id`	Namespace Id - use namespaced id ns("indexEvents") inside diagnosticsExplorer module

`LargeDataTable` *Large Data Table*

Description

Large data table R6 class.

Uses ResultModelManager::ConnectionHandler class to create paginating tables

NOTE Only currently works with sqlite and postgresql database backends (probably redshift too) as this method uses limit and offset for the queries

Alternatively, you might want to subclass this class. For example, if your backend query is against an API such as and ATLAS instance or ATHENA

If subclassing use inheritance and treat this class as an interface to implement - implementing the methods:

get

Public fields

`baseQuery` query string sql

`countQuery` count query string (should match query). Can be auto generated with sub query (default) but this will likely result in slow results

`connectionHandler` ResultModelManager connection handler to execute query inside initialize

Methods

Public methods:

- `LargeDataTable$new()`
- `LargeDataTable$getCount()`
- `LargeDataTable$getPage()`
- `LargeDataTable$getAllResults()`
- `LargeDataTable$clone()`

Method `new()`**:**

Usage:

```
LargeDataTable$new(connectionHandler, baseQuery, countQuery = NULL)
```

Arguments:

`connectionHandler` ResultModelManager connectionHandler instance

`baseQuery` base sql query

`countQuery` count query string (should match query). Can be auto generated with sub query (default) but this will likely result in slow results

Returns: self get count

Method `getCount()`**:** execute count query with specified parameters

Usage:

```
LargeDataTable$getCount(...)
```

Arguments:

`...`

Returns: count Get Page

Method `getPage()`**:**

Usage:

```
LargeDataTable$getPage(pageNum, pageSize = self$pageSize, ...)
```

Arguments:

`pageNum` page number

`pageSize` page size

`...`

Returns: data.frame of query result get all results

Method `getAllResults()`**:**

Usage:

```
LargeDataTable$getAllResults(...)
```

Arguments:

...

Returns: data.frame of all results. Used for large file downloads

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

Usage:

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

Arguments:

`deep` Whether to make a deep clone.

`largeTableServer` *Large Table Component Server*

Description

Display large data tables in a consistent way - server side pagination for reactable objects

Usage

```
largeTableServer(
  id,
  ldt,
  inputParams,
  modifyData = NULL,
  columns = shiny::reactive(list()),
  ...
)
```

Arguments

`id`	Shiny module id. Must match Large Table Viewer
`ldt`	LargeDataTable instance
`inputParams`	reactive that returns list of parameters to be passed to ldt
`modifyData`	optional callback function that takes the data page, page number, page size as parameters must return data.frame compatable instance
`columns`	List or reactable returning list of reactable::columnDef objects
`...`	Additional reactable options (searchable, sortable

`largeTableView` *Large Table Component Viewer*

Description

Componenet for results sets with many thousands of rows More limited than other table components in terms of automatic handling of search and filtering but will allow responsive apps

Usage

```
largeTableView(
  id,
  pageSizeChoices = c(10, 25, 50, 100),
  selectedPageSize = 10,
  fullDownloads = TRUE
)
```

Arguments

`id`	Shiny module id. Must match largeTableServer
`pageSizeChoices`	numeric selection options for pages
`selectedPageSize`	numeric selection options for pages
`fullDownloads`	allow download button of full dataset from query

`makeButtonLabel` *Make a label for an html button*

Description

Make a label for an html button

Usage

```
makeButtonLabel(label)
```

Arguments

`label`	The desired label for hte button

Value

html code to make a button label

`OhdsiShinyModules` *OhdsiShinyModules*

Description

A selection of shiny modules for exploring standardized OHDSI results

`orpahanConceptsView` *Orphan Concepts View*

Description

Use for customizing UI

Usage

```
orpahanConceptsView(id)
```

Arguments

`id` Namespace Id - use namespaced id ns("orphanConcepts") inside diagnosticsExplorer module

`patientLevelPredictionCalibrationServer` *The module server for exploring prediction validation results*

Description

The module server for exploring prediction validation results

Usage

```
patientLevelPredictionCalibrationServer(
  id,
  performanceId,
  connectionHandler,
  inputSingleView,
  resultDatabaseSettings
)
```

Arguments

`id` the unique reference id for the module

`performanceId` the performance id in the database

`connectionHandler` the connection to the prediction result database

`inputSingleView` the current tab

`resultDatabaseSettings` a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the prediction calibration module

`patientLevelPredictionCalibrationViewer`

The module viewer for exploring prediction model calibration results

---

Description

The module viewer for exploring prediction model calibration results

Usage

```
patientLevelPredictionCalibrationViewer(id)
```

Arguments

`id`	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the prediction model calibration module

---

`patientLevelPredictionCovariateSummaryServer`

The module server for exploring prediction covariate summary results

---

Description

The module server for exploring prediction covariate summary results

Usage

```
patientLevelPredictionCovariateSummaryServer(
  id,
  modelDesignId,
  developmentDatabaseId,
  performanceId,
  connectionHandler,
  inputSingleView,
  resultDatabaseSettings
)
```

Arguments

`id` the unique reference id for the module

`modelDesignId` unique id for the model design

`developmentDatabaseId` unique id for the development database

`performanceId` unique id for the performance results

`connectionHandler` the connection to the prediction result database

`inputSingleView` the current tab

`resultDatabaseSettings` a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the covariate summary module

`patientLevelPredictionCovariateSummaryViewer` *The module viewer for exploring prediction covariate summary results*

Description

The module viewer for exploring prediction covariate summary results

Usage

```
patientLevelPredictionCovariateSummaryViewer(id)
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the covariate summary module

patientLevelPredictionCutoffServer

The module server for exploring prediction cut-off results

Description

The module server for exploring prediction cut-off results

Usage

```
patientLevelPredictionCutoffServer(
  id,
  performanceId,
  connectionHandler,
  inputSingleView,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`performanceId`	the performance id in the database
`connectionHandler`	the connection to the prediction result database
`inputSingleView`	the current tab
`resultDatabaseSettings`	a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the prediction cut-off module

patientLevelPredictionCutoffViewer

The module viewer for exploring prediction cut-off results

Description

The module viewer for exploring prediction cut-off results

Usage

```
patientLevelPredictionCutoffViewer(id)
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the prediction cut-off module

patientLevelPredictionDesignSummaryServer

The module server for exploring prediction designs in the results

Description

The module server for exploring prediction designs in the results

Usage

```
patientLevelPredictionDesignSummaryServer(
  id,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id` the unique reference id for the module

`connectionHandler` the connection to the prediction result database

`resultDatabaseSettings` a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the prediction design module

`patientLevelPredictionDesignSummaryViewer` *The module viewer for exploring prediction designs that have been run*

Description

The module viewer for exploring prediction designs that have been run

Usage

```
patientLevelPredictionDesignSummaryViewer(id)
```

Arguments

`id`	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the prediction design module

`patientLevelPredictionDiagnosticsServer` *The module server for exploring prediction diagnostic results*

Description

The module server for exploring prediction diagnostic results

Usage

```
patientLevelPredictionDiagnosticsServer(
  id,
  modelDesignId,
  connectionHandler,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`modelDesignId`	the unique id for the model design
`connectionHandler`	the connection to the prediction result database
`resultDatabaseSettings`	a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the prediction diagnostic module

patientLevelPredictionDiagnosticsViewer

The module viewer for exploring prediction diagnostic results

Description

The module viewer for exploring prediction diagnostic results

Usage

```
patientLevelPredictionDiagnosticsViewer(id)
```

Arguments

id	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the prediction diagnostic module

patientLevelPredictionDiscriminationServer

The module server for exploring prediction model discrimination results

Description

The module server for exploring prediction model discrimination results

Usage

```
patientLevelPredictionDiscriminationServer(
  id,
  performanceId,
  connectionHandler,
  inputSingleView,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`performanceId`	the performance id in the database
`connectionHandler`	the connection to the prediction result database
`inputSingleView`	the current tab
`resultDatabaseSettings`	a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the model discrimination module

`patientLevelPredictionDiscriminationViewer` *The module viewer for exploring prediction model discrimination results*

Description

The module viewer for exploring prediction model discrimination results

Usage

```
patientLevelPredictionDiscriminationViewer(id)
```

Arguments

`id`	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the model discrimination results module

patientLevelPredictionHelperFile

The location of the prediction module helper file

Description

The location of the prediction module helper file

Usage

```
patientLevelPredictionHelperFile()
```

Details

Returns the location of the prediction helper file

Value

string location of the prediction helper file

patientLevelPredictionModelSummaryServer

The module server for exploring prediction summary results

Description

The module server for exploring prediction summary results

Usage

```
patientLevelPredictionModelSummaryServer(
  id,
  connectionHandler,
  resultDatabaseSettings,
  modelDesignId
)
```

Arguments

`id`	the unique reference id for the module
`connectionHandler`	the connection to the prediction result database
`resultDatabaseSettings`	a list containing the result schema and prefixes
`modelDesignId`	a reactable id specifying the prediction model design identifier

Details

The user specifies the id for the module

Value

The server to the summary module

`patientLevelPredictionModelSummaryViewer`

The module viewer for exploring prediction summary results

Description

The module viewer for exploring prediction summary results

Usage

```
patientLevelPredictionModelSummaryViewer(id)
```

Arguments

`id`	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the summary module

`patientLevelPredictionNbServer`

The module server for exploring prediction net-benefit results

Description

The module server for exploring prediction net-benefit results

Usage

```
patientLevelPredictionNbServer(
  id,
  performanceId,
  connectionHandler,
  inputSingleView,
  resultDatabaseSettings
)
```

Arguments

`id` the unique reference id for the module

`performanceId` the performance id in the database

`connectionHandler`
the connection to the prediction result database

`inputSingleView`
the current tab

`resultDatabaseSettings`
a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the net-benefit module

`patientLevelPredictionNbViewer`
The module viewer for exploring prediction net-benefit results

---

Description

The module viewer for exploring prediction net-benefit results

Usage

```
patientLevelPredictionNbViewer(id)
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the net-benefit module

patientLevelPredictionServer *The module server for exploring PatientLevelPrediction*

Description

The module server for exploring PatientLevelPrediction

Usage

```
patientLevelPredictionServer(
  id,
  connectionHandler,
  resultDatabaseSettings = list(port = 1)
)
```

Arguments

`id`	the unique reference id for the module
`connectionHandler`	a connection to the database with the results
`resultDatabaseSettings`	a list containing the prediction result schema and connection details

Details

The user specifies the id for the module

Value

The server for the PatientLevelPrediction module

patientLevelPredictionSettingsServer *The module server for exploring prediction settings*

Description

The module server for exploring prediction settings

Usage

```
patientLevelPredictionSettingsServer(
  id,
  modelDesignId,
  developmentDatabaseId,
  performanceId,
  connectionHandler,
  inputSingleView,
  resultDatabaseSettings
)
```

Arguments

`id` the unique reference id for the module

`modelDesignId` unique id for the model design

`developmentDatabaseId` unique id for the development database

`performanceId` unique id for the performance results

`connectionHandler` the connection to the prediction result database

`inputSingleView` the current tab

`resultDatabaseSettings` a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the settings module

`patientLevelPredictionSettingsViewer`

The module viewer for exploring prediction settings

Description

The module viewer for exploring prediction settings

Usage

```
patientLevelPredictionSettingsViewer(id)
```

Arguments

`id` the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the settings module

patientLevelPredictionValidationServer

The module server for exploring prediction validation results

Description

The module server for exploring prediction validation results

Usage

```
patientLevelPredictionValidationServer(
  id,
  modelDesignId,
  developmentDatabaseId,
  performanceId,
  connectionHandler,
  inputSingleView,
  resultDatabaseSettings
)
```

Arguments

`id`	the unique reference id for the module
`modelDesignId`	identifier for the model design
`developmentDatabaseId`	identifier for the development database
`performanceId`	identifier for the performance
`connectionHandler`	the connection to the prediction result database
`inputSingleView`	the current tab
`resultDatabaseSettings`	a list containing the result schema and prefixes

Details

The user specifies the id for the module

Value

The server to the validation module

patientLevelPredictionValidationViewer

The module viewer for exploring prediction validation results

Description

The module viewer for exploring prediction validation results

Usage

```
patientLevelPredictionValidationViewer(id)
```

Arguments

id	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the validation module

patientLevelPredictionViewer

The module viewer for exploring PatientLevelPrediction

Description

The module viewer for exploring PatientLevelPrediction

Usage

```
patientLevelPredictionViewer(id = 1)
```

Arguments

id	the unique reference id for the module

Details

The user specifies the id for the module

Value

The user interface to the PatientLevelPrediction viewer module

`phevaluatorHelperFile` *The location of the phevaluator module helper file*

Description

The location of the phevaluator module helper file

Usage

```
phevaluatorHelperFile()
```

Details

Returns the location of the cohort-generator helper file

Value

String location of the phevaluator helper file

`phevaluatorServer` *The module server for the main phevaluator module*

Description

The module server for the main phevaluator module

Usage

```
phevaluatorServer(id, connectionHandler, resultDatabaseSettings)
```

Arguments

`id`	The unique reference id for the module
`connectionHandler`	A connection to the database with the results
`resultDatabaseSettings`	A named list containing the cohort generator results database details (schema, table prefix)

Value

The phevaluator main module server

phevaluatorViewer *The viewer of the phevaluator module*

Description

The viewer of the phevaluator module

Usage

```
phevaluatorViewer(id)
```

Arguments

id	The unique reference id for the module

Value

The user interface to the phevaluator results viewer

resultTableServer *Result Table Server*

Description

Result Table Server

Usage

```
resultTableServer(
  id,
  df,
  colDefsInput,
  selectedCols = NULL,
  sortedCols = NULL,
  elementId = NULL,
  addActions = NULL,
  downloadedFileName = NULL,
  groupBy = NULL
)
```

Arguments

id	string, table id must match resultsTableViewer function
df	reactive that returns a data frame
colDefsInput	named list of reactable::colDefs
selectedCols	string vector of columns the reactable should display to start by default. Defaults to ALL if not specified.
sortedCols	string vector of columns the reactable should sort by by default. Defaults to no sort if not specified.

`elementId` optional string vector of element Id name for custom dropdown filtering if present in the customColDef list. Defaults to NULL.

`addActions` add a button row selector column to the table to a column called 'actions'. actions must be a column in df

`downloadedFileName` string, desired name of downloaded data file. can use the name from the module that is being used

`groupBy` The columns to group by

Value

shiny module server

`resultTableViewer` *Result Table Viewer*

Description

Result Table Viewer

Usage

```
resultTableViewer(id = "result-table", downloadedFileName = NULL)
```

Arguments

`id` string

`downloadedFileName` string, desired name of downloaded data file. can use the name from the module that is being used

Value

shiny module UI

`sccsHelperFile` *The location of the description module helper file*

Description

The location of the description module helper file

Usage

```
sccsHelperFile()
```

Details

Returns the location of the description helper file

Value

string location of the description helper file

`sccsServer` *The module server for exploring SCCS*

Description

The module server for exploring SCCS

Usage

```
sccsServer(id, connectionHandler, resultDatabaseSettings = list(port = 1))
```

Arguments

`id` the unique reference id for the module

`connectionHandler`
a connection to the database with the results

`resultDatabaseSettings`
a list containing the prediction result schema and connection details

Details

The user specifies the id for the module

Value

The server for the PatientLevelPrediction module

`sccsView` *SCCS shiny module UI code*

Description

Load the ui for the sccs module

Usage

```
sccsView(id = "sccs-module")
```

Arguments

`id` id for module

`timeDistributionsView` *timeDistributions view*

Description

Use for customizing UI

Usage

```
timeDistributionsView(id)
```

Arguments

`id`	Namespace Id - use namespaced id ns("imeDistributions") inside diagnosticsExplorer module

`visitContextView` *Visit context module view*

Description

Use for customizing UI

Usage

```
visitContextView(id)
```

Arguments

`id`	Namespace Id - use namespaced id ns("vistConext") inside diagnosticsExplorer module