

Package ‘OmopViewer’

April 15, 2025

Type Package

Title Visualise OMOP Results using 'shiny' Applications

Version 0.3.0

Maintainer Martí Català <martí.catalasabate@ndorms.ox.ac.uk>

Description Visualise results obtained from analysing data mapped to the Observational Medical Outcomes Partnership (OMOP) common data model using 'shiny' applications.

License Apache License (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

URL <https://ohdsi.github.io/OmopViewer/>

Depends R (>= 4.1)

Imports bslib, cli, dplyr, DT, glue, gt, markdown, omopgenerics (>= 0.4.0), purrr, rlang, shiny, snakecase, stringr, styler, tidyr, usethis, visOmopResults (>= 0.5.0), yaml

Suggests CDMConnector, CodelistGenerator (>= 3.2.0), CohortCharacteristics (>= 0.3.0), CohortConstructor, cpp11, devtools, DiagrammeR, DiagrammeRsvg, DrugUtilisation (>= 0.7.1), duckdb, ggplot2, here, htmlwidgets, IncidencePrevalence (>= 0.8.0), knitr, omock (>= 0.3.0), OmopSketch, PatientProfiles, plotly, progress, reactable, renv, rmarkdown, RPostgres, rsconnect, rsvg, shinycssloaders, shinyTree, shinyWidgets, sortable, testthat (>= 3.1.5), utils, webshot2, zip

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

NeedsCompilation no

Author Martí Català [aut, cre] (<<https://orcid.org/0000-0003-3308-9905>>),
 Yuchen Guo [aut] (<<https://orcid.org/0000-0002-0847-4855>>),
 Edward Burn [ctb] (<<https://orcid.org/0000-0002-9286-1128>>),
 Nuria Mercade-Besora [ctb] (<<https://orcid.org/0009-0006-7948-3747>>),
 Elin Rowlands [ctb] (<<https://orcid.org/0009-0005-5166-0417>>),
 Cecilia Campanile [ctb] (<<https://orcid.org/0009-0007-6629-4661>>)

Repository CRAN

Date/Publication 2025-04-15 15:00:06 UTC

Contents

defaultPanels	2
exportStaticApp	3
getPanel	4
launchDynamicApp	5
omopViewerPanels	5
omopViewerResults	5
panelDetailsFromResult	6
Index	7

defaultPanels	<i>Default panels defined in the package.</i>
---------------	---

Description

Default panels defined in the package.

Usage

```
defaultPanels()
```

Value

Names of the default panels defined in the package.

Examples

```
defaultPanels()
```

exportStaticApp	<i>Export and launch a static shiny specific to the provided results.</i>
-----------------	---

Description

Export and launch a static shiny specific to the provided results.

Usage

```
exportStaticApp(
  result,
  directory,
  logo = "ohdsi",
  title = "",
  background = TRUE,
  summary = TRUE,
  panelDetails = panelDetailsFromResult(result),
  panelStructure = NULL,
  theme = NULL,
  open = rlang::is_interactive()
)
```

Arguments

result	A summarised_result object.
directory	Directory to create the shiny.
logo	Name of a logo or path to a logo. If NULL no logo is included. Only svg format allowed for the moment.
title	title of the shiny
background	Whether to include a background panel. Background panel content will be controlled from the generated background.md file.
summary	Whether to include a panel with a summary of content in the 'result'.
panelDetails	A named list to provide details for each one of the panels, such as: result_id, result_type, title, icon, filters and content. By default it is created using the 'panelDetailsFromResult()' function.
panelStructure	A named list of panel identifiers to organise them in drop-down menus. Identifiers names are the ones used in 'panelDetails'. By default one panel per each 'panelDetails' element is created.
theme	Specify the theme for the Shiny application. You can either select a predefined theme provided by the package (e.g., "theme1"), or define a custom theme using 'bslib::bs_theme()'. If using a custom theme, it must be provided as a character string (e.g., "bslib::bs_theme(bg = 'white', fg = 'black')").
open	Whether to open the shiny app project.

Value

The shiny app will be created in directory.

Examples

```
exportStaticApp(  
  result = omopgenerics::emptySummarisedResult(),  
  directory = tempdir()  
)
```

getPanel

Get one of the default pre-built panels.

Description

Get one of the default pre-built panels.

Usage

```
getPanel(panelId)
```

Arguments

panelId	Name of the one of the default panels. Use 'defaultPanels()' to see the available default panels.
---------	---

Value

A panel definition.

Examples

```
getPanel("incidence")
```

launchDynamicApp	<i>Launch a dynamic shiny app where you can upload results.</i>
------------------	---

Description

Launch a dynamic shiny app where you can upload results.

Usage

```
launchDynamicApp()
```

Value

Launches the shiny app.

omopViewerPanels	<i>Panels defined in the package.</i>
------------------	---------------------------------------

Description

The panels are generated using the following file: <<https://github.com/OHDSI/OmopViewer/blob/main/data-raw/panels.R>>.

Usage

```
omopViewerPanels
```

Format

```
## 'omopViewerPanels' A list of 26 'omopviewer_panel' objects:
```

omopViewerResults	<i>Mock results obtained from 'GiBleed' dataset.</i>
-------------------	--

Description

The results are generated using the following file: <<https://github.com/OHDSI/OmopViewer/blob/main/data-raw/results.R>>.

Usage

```
omopViewerResults
```

Format

```
## 'omopViewerResults' A 'summarised_result' object with 12,017 rows:
```

```
panelDetailsFromResult
```

Obtain default panel details from a '<summarised_result>' object.

Description

Obtain default panel details from a '<summarised_result>' object.

Usage

```
panelDetailsFromResult(result)
```

Arguments

result A 'summarised_result' object.

Value

A list of 'omop_viewer_panel' objects.

Examples

```
panelDetailsFromResult(omopViewerResults)
```

Index

* datasets

omopViewerPanels, [5](#)

omopViewerResults, [5](#)

defaultPanels, [2](#)

exportStaticApp, [3](#)

getPanel, [4](#)

launchDynamicApp, [5](#)

omopViewerPanels, [5](#)

omopViewerResults, [5](#)

panelDetailsFromResult, [6](#)