# Package ‘VisualSHIELD’

December 9, 2020

**Title** Shiny module to seamlessly analyze Opal (https://www.obiba.org/pages/products/opal/) and dbNP (https://www.dbnp.org/) remote data through the privacy-aware DataSHIELD (https://www.datashield.ac.uk/) analysis package

**Version** 1.0

**Author** Danilo Tomasoni

**Maintainer** Danilo Tomasoni <tomasoni@cosbi.eu>

**Description** VisualSHIELD is an open-source R package and shiny module that builds on the study database Opal and the DataSHIELD library to perform federated statistical analyses. It provides a ready-to-use web app module that can be integrated in any custom shiny app, providing a high-level interface to access all the DataSHIELD functionalities, and more, from the web. This package contains a shiny module, that means that it can be used as a building block of any custom shiny app and allows to configure and invoke DataSHIELD and Opal, and to render as HTML the output of the analyses without writing R code. The module is exposed through the VisualSHIELDUI and the VisualSHIELDServer functions. See 'example' for an example usage. It allows also to port dbNP data to Opal, in such a way that also dbNP data can be used in the analysis.

**License** BSD_3_clause + file LICENSE

**Depends** R (>= 3.6.0)

**Imports** shiny (>= 1.5.0), shinyjs (>= 2.0.0), shinydashboard (>= 0.7.1), opalr (>= 1.4.1), DSI (>= 1.1.0), dsBaseClient (>= 6.0.1), ggplot2 (>= 3.3.0), ggpubr (>= 0.4.0), cowplot (>= 1.1.0)

**Suggests** PhenotypeDatabaseRClient, knitr, rmarkdown

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

# R topics documented:

## `VisualSHIELDServer` *Load main VisualSHIELD Server component this function generates the Server counterpart for the UI with the matching id parameter.*

### Description

Load main VisualSHIELD Server component this function generates the Server counterpart for the UI with the matching id parameter.

### Usage

```
VisualSHIELDServer(
  id,
  servers,
  LOG_FILE = "VisualSHIELD.log",
  glm_max_iterations = 30
)
```

### Arguments

| | |
|---|---|
| `id` | The id of the corresponding VisualSHIELD UI. |
| `servers` | A R list with user information (name/email)and a list of opal/dbNP servers the user can connect to. |
| `LOG_FILE` | The path to the file where user activity will be logged |
| `glm_max_iterations` | The maximum number of iterations allowed for the federated analysis |

### Examples

```
## Not run:
shinyServer(function(input, output, session) {
 # login information, list of servers and user name
 login <- reactive({
   if( is.null(input$load) || !input$load )
     return(NULL)

   isolate(
     list(username="tomasoni", email="tomasoni@cosbi.eu",
               servers=list(
                 # server 1
                 list(
                         opal_server = list(id = "1",
                                           name = "DEMO",
                                           url = input$custom_server,
                                            username = "administrator",
                                            password = "password",
                                            certificate = NULL,
                                            private_key = NULL),
                         # dbNP server whose studies will be migrated
                         # to the opal server defined above
                         dashin_server = NULL
                 )
                 #, ... server n
               )
             )
     )
   })

 VisualSHIELDServer("VisualSHIELD", servers=login)

  output$server <- renderUI({
    textInput("custom_server",
              label="Server to connect to:", value="",
              placeholder = "https://opal-demo.obiba.org")
  })
})

## End(Not run)
```

## VisualSHIELDUI — *Load main VisualSHIELD UI component this function generates the UI for the shiny module corresponding to this app.*

### Description

Load main VisualSHIELD UI component this function generates the UI for the shiny module corresponding to this app.

### Usage

```
VisualSHIELDUI(id, title)
```

### Arguments

| | |
|---|---|
| `id` | The id of the module. It should match with the id param of the VisualSHIELD-Server function |
| `title` | The title of the module (as a shiny UI object such as h4) in your custom app. |

### Examples

```
## Not run:
library(shiny)
library(opalr)
library(DSI)
library(dsBaseClient)
library(VisualSHIELD)

shinyUI(
 fluidPage(
   fluidRow(column(10, uiOutput("server")),
            column(2, actionButton("load", "Update"),
            tags$style(type='text/css',
        "#load { vertical-align: middle; margin-top: 25px;}"))),
   fluidRow(VisualSHIELDUI("VisualSHIELD", h3("Demo VisualSHIELD app")))
   )
)

## End(Not run)
```