User Manual/Technical Documentation

2023-06-09

# Data Entry Application

## Access

## Entering Data

## Updating Data Entry

### Adding or removing sites

### Adding a category

# Troubleshooting

authenitcation

# auto-authenticate google sheets ... this will have you interactively authenticate using broswer  
 options(gargle\_oauth\_cache = ".secrets/")  
## auto authenticate without browser  
 gs4\_auth(  
 cache = ".secrets",  
 email = "email@gmail.com" # email for authenitcation  
 )

drive\_auth(cache = ".secrets",  
 email = "email@gmail.com") #email for authentication

# file access

#read in all the data  
#url to the folder that contains the data set google sheet  
folder\_url <- "https://drive.google.com/drive/u/1/folders/1AvavGBfoZx\_ThcXVn5gL\_buQkip76ZtQ"  
  
files <- drive\_ls(folder\_url) |>  
 filter(name == "mps\_tracker\_data") #name of data set file   
#save the file id  
main\_sheet\_id <- as\_id(files)  
#read in the data set  
main\_sheet <- read\_sheet(main\_sheet\_id) |>   
 mutate(year = as.numeric(year))

# Updating categories

## updating the ui

menuItem(  
 text = "Consistent Funding",  
 tabName = "funding",  
 icon = icon("table")  
 ),

# new chunk

# Consistent Funding tabItem----  
 tabItem(  
 #update to the name of the tab, must match tabname in lookup table  
 tabName = "funding",  
 fluidRow(  
 #update this with the actual name of the category  
 h1("Consistent Funding", align = "center"),  
 br()  
 ),  
   
 #update this to say ui\_tabname  
 uiOutput("ui\_funding"),  
  
 # start third fluid row  
   
 #update the previous and next button number to one higher than the following numbers if this is the first time a new category is being added. Otherwise the number must be one higher than the highest previous and next button in the application  
 fluidRow(  
 column(10, actionButton("prev\_4", "Previous", class = "btn-primary", style="color: #FFFFFF; background-color: #094074")),  
 column(2, align = "right", actionButton("next\_6", "Save and Continue", class = "btn-primary", style="color: #FFFFFF; background-color: #094074")),  
 ) # end fourth fluid row  
 ), # end consistent funding tab item

# updating the server code

#start consistent funding tab actions ----  
 #consistent funding tab data entry  
   
 #start consistent funding data entry  
  
 #replace with the name of the following tab, this will likely remain as summary if you are placing the new tab last  
 observe(  
 if (entry\_con() && input$tabs == "summary") {  
   
 # read in the google sheet  
 # need to do this each time we write in case multiple people are on the app  
 # identify the url  
 # also read in for checking for existing data  
 main\_sheet <- read\_sheet(main\_sheet\_id) |>   
 mutate(year = as.numeric(year))  
   
 #replace all instances of con with the three letter code for the subcategory  
 con\_lookuptable <- main\_lookuptable |>   
 filter(tab == "funding") #change to the name of the tab, must match ui and lookup table  
   
 #initialize blank data frame  
 append\_data <- tibble()  
   
 for (i in seq\_along(con\_lookuptable$subcategory)) {  
 # name of the subcategory  
 con\_sub\_category\_name <- con\_lookuptable$subcategory[i]  
   
 # get the name of the score id  
 con\_score\_input <- con\_lookuptable$score\_id[i]  
   
 # get the value of the score  
 con\_score\_value <- input[[con\_score\_input]]  
   
 # name of comment id  
 con\_comment\_input <- con\_lookuptable$comment\_id[i]  
   
 # get the value of the comment  
 con\_comment\_value <- input[[con\_comment\_input]]  
   
   
 #replace th category value with the full name of the category (what you want to appear in the data)  
 con\_row <- data\_entry\_function(google\_instance = main\_sheet\_id, google\_data = main\_sheet, year\_entered = input$year\_input, category = "Consistent Funding", sub\_category\_entered = con\_sub\_category\_name, indicator\_type = "Process Indicator", score = con\_score\_value, country = input$country\_input, site\_entered = input$site\_input, comments\_entered = con\_comment\_value, evaluator = input$name\_input)  
 if (!is.null(nrow(con\_row))){  
 append\_data <- bind\_rows(con\_row, append\_data)   
   
 }  
   
 }  
 if (nrow(append\_data) >0){  
 sheet\_append(main\_sheet\_id, data = append\_data) }  
 entry\_con(FALSE)  
 }  
 )   
   
  
 entry\_con <- reactiveVal(FALSE)  
 # consistent funding tab previous button  
 #change to one higher than the highest number previous button, needs to match ui previous button  
 observeEvent(input$prev\_4, {  
 newtab <- switch(input$tabs,  
 "community" = "funding",  
 "funding" = "community")  
 updateTabItems(session, "tabs", newtab)  
 }) # end consistent funding tab previous button  
   
 # consistent funding next button  
 observeEvent(input$next\_6, {  
 newtab <- switch(input$tabs,  
 "funding" = "summary",  
 "data" = "summary")   
 # change to the last tab  
 updateTabItems(session, "tabs", newtab)  
 entry\_con(TRUE)  
 })  
   
   
   
   
   
 #generate funding ui  
 funding\_row <- main\_lookuptable |> filter(tab == "funding")  
   
 f <- list()  
 current\_row <- fluidRow()  
 box\_counter <- 0  
   
 for (i in 1:nrow(funding\_row)) {  
 current\_place <- funding\_row[i,]  
 box <- sub\_category\_box(current\_place,sub\_category\_number)  
 column <- column(width = 4, box)  
 current\_row <- tagAppendChild(current\_row, column)  
 box\_counter <- box\_counter + 1  
 sub\_category\_number <- sub\_category\_number +1  
   
 if (box\_counter == 3 || i == nrow(funding\_row)) {  
 # Add the current row to the list and reset the counter and row  
 f[[length(f) + 1]] <- current\_row  
 current\_row <- fluidRow()  
 box\_counter <- 0  
 }  
 }   
   
 output$ui\_funding <- renderUI(f)  
   
 outputOptions(output, "ui\_funding", suspendWhenHidden = FALSE)  
   
 #end consistent funding ui generation  
   
# end consistent funding tab actions