

User Level Access in Shiny Apps

useR! Jogja

M Farkhan N

Business Intelligence
Amartha Mikro Fintek

Data Analyst - Data Scientist - BI

farkhan.novianto@amartha.com

farkhan.novianto@gmail.com

<http://t.me/mfarkhann>

What is Shiny

- [Shiny](#) is an open source R package that provides an elegant and powerful web framework for building web applications using R. Shiny helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge.
- You can host standalone apps on a webpage or embed them in [R Markdown](#) documents or build [dashboards](#).
- You can also extend your Shiny apps with [CSS themes](#), [htmlwidgets](#), and JavaScript [actions](#).

Hosting Shiny Web App

Deploy to the Cloud

- [Shinyapps.io](https://shinyapps.io) (Free and Paid Option)

Deploy on-premises (open source)

- [Shiny Server](https://shiny.posit.co/shiny-server/)

Deploy on-premises (Commercial)

- [RStudio Connect](https://posit.co/products/server/)
- [Shiny Server Pro](https://shiny.posit.co/shiny-server/)

If you have budget, please use Commercial Product

- Support Security & Authentication
- Scale applications across multiple R processes
- Performance and resource metrics
- Push-button publish from the RStudio IDE (shinyapps.io and RStudio Connect)

Several Alternative to protect your app with authentication

- Host the app behind a [reverse proxy server](#) (nginx)
- Use [ShinyProxy](#) (docker based)

All user will see same content and you can't know who has been logged in

User Level Access :
Each User (or group) who
logged-in can see different
content

Demo

<https://farkhan.shinyapps.io/render-ui/>

<https://farkhan.shinyapps.io/shinyjs-hide/>

<https://farkhan.shinyapps.io/persistent-login/>

username	setosa	versicolor	virginica	admin
password	1111	2222	3333	9999

<https://github.com/mfarkhann/shiny-password>

Use Render UI

All UI processed in server to check if user logged in

Step shiny password using Render UI

1. UI only contain renderUI
2. Make Reactive Values to save login data
3. Show login page when not logged in
4. Processing login when click button and then save username who logged in
5. Process Data based on username
6. Show UI for logged in user
7. Clear Reactive Values when logout

UI.R

```
library(shiny)
```

```
shinyUI(  
  fluidPage(  
    title = judul,  
    uiOutput("ui")  
  ))
```

Reactive Values to save login data

```
sudah_login <- reactiveVal(value = FALSE)  
data_user <- reactiveVal(value = NULL)
```

Show login page when not logged in

```
if (!sudah_login()) {  
    fluidRow(  
        column(width = 12, align='center',  
            br(), br(), br(), br(), br(),  
            h3(judul),  
            br(), br(),  
            textInput('username', "", placeholder = 'user'),  
            passwordInput('password', "", placeholder = 'pass'),  
            br(),  
            actionButton('login', 'Login', class = "btn-primary")  
        )  
    )  
}
```

Processing Login

```
observeEvent(input$login, {  
  cred_user <- df_credentials %>%  
    filter(username == input$username)  
  if(nrow(cred_user)) {  
    if (input$password == cred_user$password) {  
      sudah_login(TRUE)  
      data_user(cred_user)  
    }  
  }  
})
```

Process Data based on username

```
output$contohData <- renderTable({  
  req(data_user())$username)  
  df_data <- iris  
  nama = data_user()$username
```

```
  if(nama!='admin') {  
    df_data <- df_data %>%  
      filter(Species == nama)  
  }  
  df_data %>% group_by(Species) %>% summarise_all(list(Total = sum))  
})
```

Show UI for logged in user

```
if (!sudah_login()) {  
    UI when not logged in  
}else {  
    fluidRow(  
        column(12,  
            h3(judul), align='center',  
            br(),  
            tableOutput("contohData")  
        )))  
    })  
})
```


Clear Reactive Value when Logout

```
observeEvent(input$logout, {  
  sudah_login(FALSE)  
  data_user(NULL)  
})
```

Use ShinyJS Hide

Make Shiny App as usual, but hide several parts

Step shiny password using ShinyJS Hide

1. UI as usual, but hide several (or all) parts that need login
2. Show login page when not logged in
3. Processing login when click button, save username who logged in and show Hidden UI
4. Process Data based on username
5. Clear Reactive Values and hide content when logout

UI as Usual, hidden sensitive content

```
shinyjs::hidden(  
  div(  
    id = "div_data",  
    box(width = 12,  
      h3(textOutput('text_user')),  
      br(),  
      tableOutput("contohData")  
    )))
```

UI for fill password

```
div(id = "div_password",  
    align = 'center',  
    h3("Masukkan Password untuk melihat datanya"),  
    passwordInput("password","", placeholder = "pass"),  
    actionButton("login", strong("Login")),  
    shinyjs::hidden(  
        div(id = "login-error",  
            div(icon("exclamation-circle"),  
                shiny::tags$b("Password Salah")  
            )  
        )))  
,
```

Processing Login

```
observeEvent(input$login, {  
  cred_user <- df_credentials %>%  
    filter(password == input$password)  
  if(nrow(cred_user)==1) {  
    data_user(cred_user)  
    shinyjs::hide("div_password")  
    shinyjs::show("div_data")  
    shinyjs::show("logout")  
  } else {  
    shinyjs::show(id = "login-error", anim = TRUE, animType = "fade")  
    shinyjs::delay(2000, shinyjs::hide(id = "login-error", ..., time = 0.5))  
  }  
})
```

Clear Reactive Values and hide content when logout

```
observeEvent(input$logout, {  
  data_user(NULL)  
  shinyjs::hide("logout")  
  shinyjs::hide("div_data")  
  shinyjs::show("div_password")  
  shinyjs::reset('div_password')  
})
```

Persistent Login

Keep logged-in even in new session

Cookie Based Authentication

1. Check if already logged-in
Get access token in cookies data and check if match in user credential
2. If not logged-in (no cookies or not matched)
 - a. User Login
User insert user and password then send to server
 - b. Validating Login
Check if user exist and password matched with hashed password
 - c. Generation access token
If user matched, generate token which uniquely identify user's session
Save it in user credentials and send to cookies (with expiration time)
3. User will make request using token that also available in user credential

Source : [using-cookie-based-authentication-with-shin](#)

Setup R (Shiny) to interact with Cookies

Using shinyjs to Set and Get Cookies

Download Script to interact with Cookies using JS

```
if (!dir.exists('www/')) {  
  dir.create('www')  
}
```

```
download.file(  
  url =  
'https://raw.githubusercontent.com/js-cookie/js-cookie/master/src/js.cookie.js',  
  destfile = 'www/js.cookie.js'  
)
```

Script JS to get set and remove Cookies

```
jsCode <- '  
shinyjs.getcookie = function(params) {  
  var cookie = Cookies.get("shiny_user_session");  
  if (typeof cookie !== "undefined") {  
    Shiny.onInputChange("jscookie", cookie);  
  } else {  
    var cookie = "";  
    Shiny.onInputChange("jscookie", cookie);}}
```

```
shinyjs.setcookie = function(params) {  
  /* expires after 12 hours */  
  Cookies.set("shiny_user_session", escape(params), { expires: 0.5 });  
  Shiny.onInputChange("jscookie", params);}
```

```
shinyjs.rmcookie = function(params) {  
  Cookies.remove("shiny_user_session");  
  Shiny.onInputChange("jscookie", "");}
```

Source JS in Shiny using ShinyJS

```
dashboardBody(  
  shinyjs::useShinyjs(),  
  tags$script(src = "js.cookie.js"),  
  extendShinyjs(text = jsCode),  
  uiOutput("ui")  
)
```

Check if already logged-in

```
observe({  
  js$getcookie()  
  credentials <- readRDS("data/credentials.rds")  
  session_cred <- credentials$session  
  if (!is.null(input$jscookie) && input$jscookie!=" " && input$jscookie %in%  
session_cred) {  
    sudah_login(TRUE)  
    row_email <- which(session_cred==input$jscookie)  
    data_user(credentials[row_email,])  
    shinyjs::show("logout")  
  }  
})
```

Validate Login and generate Access Token

```
cred_user <- df_credentials %>%  
  filter(username == input$username)  
  
if (bcrypt::checkpw(input$password, hash = cred_user$password)) {  
  sudah_login(TRUE)  
  data_user(cred_user)  
  
  sessionid <- paste(collapse = "", sample(x = c(letters, LETTERS, 0:9), size = 64,  
replace = TRUE))  
  js$setcookie(sessionid)  
  update_user_session(user = input$username, sessionid)  
  
  shinyjs::show("logout")  
}
```

Terima kasih

<https://t.me/GNURIndonesia>

https://t.me/RIndonesia_Jateng

<https://bit.ly/2FoBt3i>