demo_Rselenium

2024-09-23

#What is RSelenium?

Selenium is an open-source and automated testing tool that is used for testing web applications. This was an improvement on manual testing of web applications, making it easier for software developers to test applications across various browsers. One of the advantages of using Selenium is that it can be used with multiple programming languages, including R. It is also helpful in scraping dynamic websites, especially when the URL does not change with each new page.

In order to start using RSelenium on your browser (For today's example, I use Chrome), you need to download chromedriver. Follow instructions from here: https://stackoverflow.com/questions/74811360/how-to-install-chromedriver-with-binman-package-on-mac. In order to download newer chrome driver, you can use the link here: https://googlechromelabs.github.io/chrome-for-testing/#stable

For today's presentation, I will not be going into the detail of this because Xiuling has organized a workshop for this topic on Wednesday, which will go into detail of setting up everything.

The website that we will be scraping today is: https://infopemilu.kpu.go.id/Pemilu/Dct_dpd. This is the profile of the members of the People's Representative Council of Indonesia.

Before we begin with the codes, let's inspect the website. Each website is different. Therefore, it is essential to spend sometime understanding the layout of the website before jumping in. Doing this will allow you to plan the scraping accordingly and save you a lot of time in the long run.

Now that we have inspected the page, let's begin coding.

```
eCaps <- list(chromeOptions = list(args = c('--disable-gpu', '--lang=en'),
                                   prefs = list(
                                     "profile.managed_default_content_settings.images" = 2L
                                   )
))

rs_driver_object <- rsDriver(browser = "chrome",
                             chromever = "128.0.6613.137",
                             verbose = F,
                             port = free_port(),
                             extraCapabilities = eCaps)
```

Now that we have set it up, let's navigate to the page:

```
remDr <- rs_driver_object$client
remDr$open()
```

```
## [1] "Connecting to remote server"
## $acceptInsecureCerts
## [1] FALSE
##
## $browserName
## [1] "chrome"
##
```

```
## $browserVersion
## [1] "129.0.6668.59"
##
## $chrome
## $chrome$chromedriverVersion
## [1] "128.0.6613.137 (fe621c5aa2d6b987e964fb1b5066833da5fb613d-refs/branch-heads/6613@{#1711})"
##
## $chrome$userDataDir
## [1] "C:\\Users\\0111s\\AppData\\Local\\Temp\\scoped_dir8428_674891500"
##
##
## $`fedcm:accounts`
## [1] TRUE
##
## $`goog:chromeOptions`
## $`goog:chromeOptions`$debuggerAddress
## [1] "localhost:61518"
##
##
## $networkConnectionEnabled
## [1] FALSE
##
## $pageLoadStrategy
## [1] "normal"
##
## $platformName
## [1] "windows"
##
## $proxy
## named list()
##
## $setWindowRect
## [1] TRUE
##
## $strictFileInteractability
## [1] FALSE
##
## $timeouts
## $timeouts$implicit
## [1] 0
##
## $timeouts$pageLoad
## [1] 300000
##
## $timeouts$script
## [1] 30000
##
##
## $unhandledPromptBehavior
## [1] "dismiss and notify"
##
## $`webauthn:extension:credBlob`
## [1] TRUE
##
```

```
## $`webauthn:extension:largeBlob`
## [1] TRUE
##
## $`webauthn:extension:minPinLength`
## [1] TRUE
##
## $`webauthn:extension:prf`
## [1] TRUE
##
## $`webauthn:virtualAuthenticators`
## [1] TRUE
##
## $webdriver.remote.sessionid
## [1] "bbf9e986c5e09b31bf09f6df5aebff70"
##
## $id
## [1] "bbf9e986c5e09b31bf09f6df5aebff70"
```

```
remDr$setTimeout(type = "page load",
                 milliseconds = 200000)
remDr$setTimeout(type = "implicit",
                 milliseconds = 200000)
remDr$maxWindowSize()
remDr$navigate("https://infopemilu.kpu.go.id/Pemilu/Dct_dpd")
```

We see that our website is open on RSelenium. Now, let's plan the next steps. We will start by clicking on the dropdown menus for the profile page.

To learn more about selectors: https://docs.ropensci.org/RSelenium/articles/basics.html

```
dropdown_100_items <- remDr$findElement(using = "css", "option[value='100']")
dropdown_100_items$clickElement()
Sys.sleep(5)

regions_dropdown <- remDr$findElement(using = "id", value = "filterDapil")
regions <- regions_dropdown$selectTag()$text

#We are also setting up the start region and start candidate
start_region <- 11 # valid region number 11-96 (This is specific to the website, which is why inspectin
start_candidate <- 1
```

Now that we have set up the dropdown for region, let's set up the selectors for profile button

```
region_name <- regions[start_region-9]
selector <- sprintf("option[value='%s']", start_region)

dropdownmenu2 <- remDr$findElement(using = "id", value = "filterDapil")
Sys.sleep(1)
dropdownmenu2$clickElement()
item2 <- remDr$findElement(using = "css", selector)
item2$clickElement()
Sys.sleep(8)


profileButtons <- remDr$findElements(using = "css", "input[class='btn btn-secondary']")
len <- length(profileButtons)
```

```
start_region_log_msg <- sprintf("Start Region %s: %s has %s active profiles", start_region-10, region_na


# Extract candidate info
print(sprintf("** Start Region: %s, Candidate: %s", region_name, start_candidate))
```

```
## [1] "** Start Region: ACEH, Candidate: 1"
```

```
tmpProfileButtons <- remDr$findElements(using = "css", "input[class='btn btn-secondary']")
#Sys.sleep(2)
btn <- tmpProfileButtons[[start_candidate]]

#print('-- Scrolling')
loc <- btn$getElementLocation()
script <- sprintf("window.scrollTo(%s,%s);", loc$x, loc$y-500)
remDr$executeScript(script, args = list("dummy"))
```

```
## list()
```

```
#print('-- Scrolled')
Sys.sleep(2)

btn$highlightElement()
btn$clickElement()
#print(cat("btn clicked"))
```

Now that we have clicked on the profile button, let's extract the data table. Here, we leverage the rvest package to read html and return the tables in the profile.

```
Sys.sleep(10)
data_table <- remDr$findElement(using = 'class', 'table')
#print(data_table)
data_table_html <- data_table$getPageSource()
#print(data_table_html)
page <- read_html(data_table_html %>% unlist())
df <- html_table(page)
```

Okay, let's see what it looks like:

```
#Let's choose one table from our list
df[[3]]
```

```
## # A tibble: 1 x 4
##   `NAMA INSTANSI` JABATAN `TAHUN MULAI` `TAHUN SELESAI`
##   <chr>           <chr>           <int>           <int>
## 1 Eumpang Breuh   Aktor            2006            2020
```

Aside from the tables, we also wanted to get the links for the image, here is how we do that

```
# extract photo url
img <- remDr$findElement(using = "css", "img[alt='Foto']") # alt text is Foto or Photo
img_url <- img$getElementAttribute("src")
print(img_url)
```

```
## [[1]]
## [1] "https://infopemilu.kpu.go.id/berkas-dpd-dct/11/11_1_ABDUL%20HADI%20BANG%20JONI.png"
```

Once you have all of the required information from the page, rest of the work is all about setting up folders, cleaning the initial table you received so that you can make the data useful for your research purposes! I

have a R sript where I automate it for all of the profiles and all of the regions. Let's go to that example. However, before we leave this page, we need to close the driver.

```
remDr$close()
rs_driver_object$server$stop()
```

```
## [1] TRUE
```

Fin!