Scraper les communiqués de presse du secrétaire général des nations unies

Malo Jan

```
# Import packages

needs(tidyverse, rvest)
```

Principales fonctions d'`rvest` :

- `read_html()` : Extraction du code source d'une page HTML
- `html_element()` : Selection d'un élément sur cette page
- `html_text()` : Extraction du texte de cet élément
- `html_table()` : Extraction d'un tableau de cet élément
- `html_attr()` : Extraction d'un attribut de cet élément (ex : liens)

Etape 1 : familiarisation avec le site web

- Communiqués de presse du secrétaire général des nations unies

Etape 2 : extraction du contenu d'une seule page web

```
# Création d'une url d'un communiqué

url_test <- "https://press.un.org/en/2023/sgsm22043.doc.htm"

# Extraction du code html de la page
(page_html <- read_html(url_test))
```

```
{html_document}
<html lang="en" dir="ltr">
[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8 ...
[2] <body class="node-327507 node-type--press">\n        <a href="#main-conte ...
```

```r
# Extraction du titre du communiqué

(title <- page_html %>%
  html_element(".page-header") %>%
  html_text())
```

```
[1] "Secretary-General Welcomes Agreement between Israel, Hamas, Calling It ‘an Important St
```

```r
# Extraction du texte du communiqué

(text <- page_html %>%
  html_element(".field--type-text-with-summary") %>%
  html_text())
```

```
[1] "The following statement was issued today by the Spokesman for UN Secretary-General Antó
```

```r
# Extraction de la date du communiqué

(date <- page_html %>%
  html_element("time") %>%
  html_text())
```

```
[1] "22 November 2023"
```

```r
# Extraction des keywords

(keywords <- page_html %>%
  html_elements(".field__items a") %>%
  html_text() |>
        str_c(collapse = "|"))
```

```
[1] "Palestinian issues|Middle East|Israel|State of Palestine"
```

```
(id <- page_html |>
    html_element(".field--name-field-symbol") |>
    html_text())
```

```
[1] "SG/SM/22043"
```

```
un_pr <- tibble(title, text, date, keywords, id)
```

Etape 3 : extraction des urls de tous les communiqués

```
ex_urls <- "https://press.un.org/en/content/secretary-general/press-release" |>
    read_html() |>
    html_elements("h3 > a") |>
    html_attr("href")

ex_urls <- str_c("https://press.un.org", ex_urls)

ex_urls
```

```
 [1] "https://press.un.org/en/2023/sgsm22043.doc.htm"
 [2] "https://press.un.org/en/2023/sgsm22047.doc.htm"
 [3] "https://press.un.org/en/2023/sgsm22046.doc.htm"
 [4] "https://press.un.org/en/2023/sgsm22045.doc.htm"
 [5] "https://press.un.org/en/2023/sgsm22044.doc.htm"
 [6] "https://press.un.org/en/2023/sgsm22042.doc.htm"
 [7] "https://press.un.org/en/2023/sgsm22041.doc.htm"
 [8] "https://press.un.org/en/2023/sga2242.doc.htm"
 [9] "https://press.un.org/en/2023/sgsm22040.doc.htm"
[10] "https://press.un.org/en/2023/sgsm22039.doc.htm"
```

- Souvent, on veut plus d'une url (et là, c'est là que commence le fun et la beauté de l'automatisation !)
- Les urls sont parfois structurées de façon logique ce qui rend l'extraction facile
  - https://press.un.org/en/content/secretary-general/press-release?page=1
  - https://press.un.org/en/content/secretary-general/press-release?page=2
  - https://press.un.org/en/content/secretary-general/press-release?page=3
- Dans ce cas, le workflow consiste à :

- Identifier toutes les urls des pages où il y a des communiqués de presse
- Pour chacune de ces pages, collecter les urls des communiqués de presses
- Pour chaque communiqué de presse : extraire le titre, l'auteur, la date et le texte

```
urls <- str_c("https://press.un.org/en/content/secretary-general/press-release?page=", 1:1


collect_urls <- function(x) {
    page <- read_html(x)

    urls <- tibble(
        link = page |>
            html_elements("h3 > a") |>
            html_attr("href")
    ) |>
        mutate(link = str_c("https://press.un.org", link))
}

# Test sur 10 pages
pr_links <- map_df(urls[1:10], collect_urls, .progress = T)
```

```
======>-----------------------   20% |  ETA:  8s

============>-----------------   40% |  ETA:  8s

==================>------------   60% |  ETA:  6s

======================>---------   70% |  ETA:  4s

============================>---   90% |  ETA:  1s
```

Etape 4 : Collecter le texte de chaque communiqué de presse

```
# Write a function to scrap everything


(title <- page_html %>%
  html_element(".page-header") %>%
  html_text())
```

```
[1] "Secretary-General Welcomes Agreement between Israel, Hamas, Calling It ‘an Important Ste
```

```
# Extraction du texte du communiqué

(text <- page_html %>%
  html_element(".field--type-text-with-summary") %>%
  html_text())
```

```
[1] "The following statement was issued today by the Spokesman for UN Secretary-General Antó
```

```
# Extraction de la date du communiqué

(date <- page_html %>%
  html_element("time") %>%
  html_text())
```

```
[1] "22 November 2023"
```

```
# Extraction des keywords

(keywords <- page_html %>%
  html_elements(".field__items a") %>%
  html_text() |>
        str_c(collapse = "|"))
```

```
[1] "Palestinian issues|Middle East|Israel|State of Palestine"
```

```
(id <- page_html |>
    html_element(".field--name-field-symbol") |>
    html_text())
```

```
[1] "SG/SM/22043"
```

```
collect_content <- function(x) {
    # Get html code of page
    page <- read_html(x)
```

```
    # Create dataframe with the different elements

    tibble(
        # Extract title
        title = html_element(page, ".page-header") |>
            html_text(),
        text = html_element(page, ".field--type-text-with-summary") |>
            html_text(),
        date = html_element(page, "time") |>
            html_text() |>
            dmy(),
        keywords = html_elements(page, ".field__items a") |>
            html_text() |>
            str_c(collapse = "|"),
        id = html_element(page, ".field--name-field-symbol") |>
            html_text())
}

# Apply this function to all of the press releases urls

(cp <- map_df(pr_links$link[1:10], collect_content, .progress = T))
```

```
===>--------------------------   10% |  ETA: 11s

=========>--------------------   30% |  ETA:  9s

==========================>---   90% |  ETA:  1s
```

```
# A tibble: 10 x 5
   title                                          text  date       keywords id
   <chr>                                          <chr> <date>     <chr>    <chr>
 1 'By Moving at Jet Speed', International Civi~ Foll~ 2023-11-20 ""       SG/S~
 2 Deeply Shocked by Two Fatal Attacks on Pales~ The ~ 2023-11-19 "Palest~ SG/S~
 3 Commending Elections in Liberia, Secretary-G~ The ~ 2023-11-19 "Liberi~ SG/S~
 4 Activities of Secretary-General in Nepal, In~ The ~ 2023-11-17 "Nepal|~ SG/T~
 5 Activities of Secretary-General in United Ki~ On W~ 2023-11-17 "United~ SG/T~
 6 With World 'in Dire Straits', International ~ Foll~ 2023-11-16 ""       SG/S~
 7 Secretary-General Deeply Concerned by Expans~ The ~ 2023-11-15 "Human ~ SG/S~
 8 'Stand Up, Speak out' towards Building World~ Foll~ 2023-11-14 "Offici~ SG/S~
 9 Secretary-General Calls for Immediate Humani~ The ~ 2023-11-14 "Palest~ SG/S~
10 World 'Massively Off Track to Limiting Globa~ Foll~ 2023-11-14 "Enviro~ SG/S~
```

```
# If you run this for all the 19000 press releases, it will take a while
```

Etape 6 : Explorer les données