Loading form JSON, HTML, XML

By Tony Fraser

2023-09-30

Assignment

Use R to load data about books from three different file types, HTML, JSON and XML. Then, compare.

Explanation: each of these books has more than one author, and therefore this data is not inherently flat. This could be transformed and loaded into a tidy data frame, but there are other data types better suited for this structure. In R, this data is better represented by a list of lists, in python a dictionary, in scala a Map[String,Array] type, etc.

That said, to get a frame of reference before we load from JSON, HTML and XML, let's first look at what this data looks like if explicitly declared.

Load from list of lists

```
books_list <- list(
  list( title = "The Wildest Dream: The Biography of George Mallory",
        authors = c("Peter Gillman", "Leni Gillman"),
        description = "This biography delves ... disappeared during .. 1924 ... ",
        year_published = 2000
  ),
  list( title = "K2: Life and Death on the World's Most Dangerous Mountain",
        authors = c("Ed Viesturs", "David Roberts"),
        description = "This book ... allure of K2, the world's second-highest mountain..."
        year_published = 2009
  )
)
## Access and test
# > books[[1]]$authors[[2]]
#[1] "Leni Gillman"
```

Load from JSON

Most all languages have a loader that goes straight from JSON into a dictionary type data set. In R, it's called `jsonlite`. We'll omit showing what hte JSON looks like as it's almost identical to the previous page.

The four mountaineering books as JSON

```
json_path = paste(base_path, "week7/mountains.json", sep = "")
library(jsonlite)

# books <- fromJSON(readLines(json_path, warn = FALSE)) <- not this!!  This goes straight
# a dataframe. We don't want this additional processing, we want to keep it in native list

books_json <- fromJSON(readLines(json_path, warn = FALSE), simplifyDataFrame = FALSE)

## Access and test
#> books_json[[1]]$authors[[2]]
#[1] "Leni Gillman"
```

Load from XML

The four mountaineering books as XML

There is an R xml parser called xml2. xml2 is an important package, but it's mostly because it is used as the engine that powers rvest. rvest is a critically important R package to be familiar with. It scrapes websites, extracts data from HTML DOM's, reads through CSS's, etc. And for the purpose of this paper, it also extracts data from both HTML and XML.

```
library(rvest)
xml_parsed <- read_html(paste(base_path, "week7/mountains.xml", sep = "") )

titles <- xml_parsed %>% html_nodes(xpath = ".//title") %>% html_text()

# Extract authors
authors_nodes <- xml_parsed %>% html_nodes(xpath = ".//authors")
authors <- lapply(authors_nodes, function(node) {
  node %>% html_nodes(xpath = ".//author") %>% html_text()
})

descriptions <- xml_parsed %>% html_nodes(xpath = ".//description") %>% html_text()
years <- xml_parsed %>% html_nodes(xpath = ".//year_published") %>% html_text() %>% as.num

books_xml <- lapply(1:length(titles), function(i) {
  list(
    title = titles[i],
    authors = authors[[i]],
    description = descriptions[i],
    year_published = years[i]
  )
})

# > books_xml[[1]]$authors[[2]]
# [1] "Leni Gillman"
```

Load from HTML

R's Rvest is paraser that that can extract content out of HTML files. If you want to find all the links on a page, or extract all the books in a DOM, rvest is what you are looking for.

The four mountaineering books as HTML

```
library(rvest)
html_parsed <- read_html(paste(base_path, "week7/mountains.html", sep = "") )
titles <- html_parsed %>%
  html_nodes(".book h2") %>% html_text()
authors_texts <- html_parsed %>%
  html_nodes(xpath = "//div[@class='book']/p[1]") %>% html_text()
authors <- strsplit(gsub("Authors: ", "", authors_texts), ", ")

descriptions <- html_parsed %>%
  html_nodes(xpath = "//div[@class='book']/p[2]") %>% html_text()
years_texts <- html_parsed %>%
  html_nodes(xpath = "//div[@class='book']/p[3]") %>% html_text()
years <- as.numeric(gsub("Year Published: ", "", years_texts))

books_html <- lapply(1:length(titles), function(i) {
  list(
    title = titles[i],
    authors = authors[[i]],
    description = descriptions[i],
    year_published = years[i]
  )
})
# > books_html[[1]]$authors[[2]]
#[1] "Leni Gillman"
```