Data_607_Assignment_5_Working_with_XML_and_JSON_in_R

Enid Roman

2022-10-16

XML was designed to describe data and to focus on what data is. HTML was designed to display data and to focus on how data looks. In other words, HTML is about displaying information, XML is about describing information.

The tags used to markup HTML documents and the structure of HTML documents are predefined. The author of HTML documents can only use tags that are defined in the HTML standard. On the other hand XML allows the author to define his own tags and his own document structure.

JSON is a data interchange format and only provides a data encoding specification.

Here we will do a comparison of the three, HTML, XML, and JSON.

LOADED THE NECESSARY LIBRARIES NEEDED

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(XML)
library(rvest)
```

```
##
## Attaching package: 'rvest'
##
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
library(RCurl)
```

```
##
## Attaching package: 'RCurl'
##
## The following object is masked from 'package:tidyr':
##
##     complete
```

```
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(RJSONIO)
```

```
##
## Attaching package: 'RJSONIO'
##
## The following objects are masked from 'package:jsonlite':
##
##     fromJSON, toJSON
```

```
library(rjson)
```

```
##
## Attaching package: 'rjson'
##
## The following objects are masked from 'package:RJSONIO':
##
##     fromJSON, toJSON
##
## The following objects are masked from 'package:jsonlite':
##
##     fromJSON, toJSON
```

CREATED THREE FILES , HTML, XML, AND JSON FORMATS IN VISUAL STUDIO CODE

HTML CREATED

<!DOCTYPE html>

My Books

Title

Author

Publisher

Year

Edition

ISBN

R Graphics Cookbook

Winston Chang

O'Reilly Media Inc

2019

2nd

978-1-4919-7860-3

R for Everyone

Jared P. Lander

Addison-Wesley Professional

2014

2nd

978-0-1345-4692-6

Data Science for Business

Foster Provost, Tom Fawcett

O'Reilly Media Inc.

2013

1st

978-1-4493-6132-7

XML CREATED

R Graphics Cookbook

```
    <Author>Winston Chang</Author>
    <Publisher>O'Reilly Media Inc</Publisher>
    <Year>2019</Year>
    <Edition>2nd</Edition>
    <ISBN>078-1-4919-7860-3</ISBN>
</Book>
 <Book ID = "2">
    <Title>R for Everyonek</Title>
    <Author>Jared P. Lander</Author>
    <Publisher>Addison-Wesley Professional</Publisher>
    <Year>2014</Year>
    <Edition>2nd</Edition>
    <ISBN>978-0-1345-4692-6</ISBN>
</Book>
<Book ID = "3">
    <Title>Data Science for Business</Title>
```

```
    <Author>Foster Provost, Tom Fawcett</Author>
    <Publisher>O'Reilly Media Inc.</Publisher>
    <Year>2013</Year>
    <Edition>1st</Edition>
    <ISBN>978-1-4493-6132-7</ISBN>
</Book>
```

JSON CREATED

{"My_Books" :[{ "Title" : "R Graphics Cookbook", "Author" : "Winston Chang", "Publisher" : "O'Reilly Media Inc", "Year" : "2019", "Edition" : "2nd", "ISBN" : "978-1-4919-7860-3" }, { "Title" : "R for Everyone", "Author" : "Jared P. Lander", "Publisher" : "Addison-Wesley Professional", "Year" : "2014", "Edition" : "2nd", "ISBN" : "978-0-1345-4692-6" }, {
"Title" : "Data Science for Business", "Authors" : ["Foster Provost", "Tom Fawcett"], "Publisher" : "O'Reilly Media Inc.", "Year" : "2013", "Edition" : "1st", "ISBN" : "978-1-4493-6132-7" }] }

IMPORT DATA FROM FILE

HTML

The following actions are performed to load the HTML table into R as dataframe:

Used getURL function to extract the link of the html file.

Parsed the html file with read_html function.

Used html_table function to extract a list of tables if any from the html file and convert the tables into dataframes.

```
url <- getURL('https://raw.githubusercontent.com/enidroman/data_607_data_acquisition_and_management/mai
df_HTML <- url %>%
  read_html(encoding = 'UTF-8') %>%
  html_table(header = NA, trim = TRUE) %>%
  .[[1]]

df_HTML
```

Only one table in the html file, therefore the first element of the list is returned.

```
## # A tibble: 3 x 6
##   Title                     Author                   Publi~1  Year Edition ISBN
##   <chr>                     <chr>                    <chr>   <int> <chr>   <chr>
## 1 R Graphics Cookbook       Winston Chang            O'Reil~  2019 2nd     978-~
## 2 R for Everyone            Jared P. Lander          Addiso~  2014 2nd     978-~
## 3 Data Science for Business Foster Provost, Tom Faw~ O'Reil~  2013 1st     978-~
## # ... with abbreviated variable name 1: Publisher
```

XML

The following actions are performed to load the HTML table into R as dataframe:

Parsed values in all elements into R dataframe.

Parsed the XML table into R named df_XML using xmlParse function.

Find the root node of the parsed file using xmlRoot function.

```
url <- getURL('https://raw.githubusercontent.com/enidroman/data_607_data_acquisition_and_management/mai
df_XML <- url %>%
  xmlParse() %>%
  xmlRoot() %>%
  xmlToDataFrame(stringsAsFactors = FALSE)
df_XML
```

Convert the XML table to dataframe using function xmlToDataFrame

```
##                       Title                     Author
## 1       R Graphics Cookbook              Winston Chang
## 2           R for Everyonek             Jared P. Lander
## 3 Data Science for Business Foster Provost, Tom Fawcett
##                     Publisher Year Edition              ISBN
## 1          O'Reilly Media Inc 2019     2nd 078-1-4919-7860-3
## 2 Addison-Wesley Professional 2014     2nd 978-0-1345-4692-6
## 3         O'Reilly Media Inc. 2013     1st 978-1-4493-6132-7
```

JSON

Below I know it is incorrect. If you see below this chunk I tried parcing the github and it wouldn't work for the life of me. Even Melvin was trying to help me figure it out. But we were unsucessful. Mayby you can tell us what we were doing wrong.

```
df_JSON <- rjson::fromJSON(file = "books.json")
df_JSON = as.data.frame(df_JSON)
df_JSON
```

```
##        My_books.Title My_books.Author My_books.Publisher My_books.Year
## 1 R Graphics Cookbook   Winston Chang O'Reilly Media Inc          2019
## 2 R Graphics Cookbook   Winston Chang O'Reilly Media Inc          2019
##   My_books.Edition     My_books.ISBN My_books.Title.1 My_books.Author.1
## 1              2nd 978-1-4919-7860-3   R for Everyone   Jared P. Lander
## 2              2nd 978-1-4919-7860-3   R for Everyone   Jared P. Lander
##          My_books.Publisher.1 My_books.Year.1 My_books.Edition.1
## 1 Addison-Wesley Professional            2014                2nd
```

```
## 2 Addison-Wesley Professional            2014                2nd
##     My_books.ISBN.1          My_books.Title.2 My_books.Author.2
## 1 978-0-1345-4692-6 Data Science for Business    Foster Provost
## 2 978-0-1345-4692-6 Data Science for Business       Tom Fawcett
##   My_books.Publisher.2 My_books.Year.2 My_books.Edition.2   My_books.ISBN.2
## 1  O'Reilly Media Inc.            2013                1st 978-1-4493-6132-7
## 2  O'Reilly Media Inc.            2013                1st 978-1-4493-6132-7
```

```
#json_books <- fromJSON("https://raw.githubusercontent.com/enidroman/data_607_data_acquisition_and_mana
#json_books
```

COMPARISON

HTML AND XML

HTML and XML tables seems identical. The only diffence is when parcing numeric values from source file to dataframe. The data type for "Year" is different in HTML and XML. In HTML "Year" is an integer, while XML "Year" is a character.

```
all.equal(df_HTML,df_XML)
```

```
## [1] "Attributes: < Component \"class\": Lengths (3, 1) differ (string compare on first 1) >"
## [2] "Attributes: < Component \"class\": 1 string mismatch >"
## [3] "Component \"Title\": 1 string mismatch"
## [4] "Component \"Year\": Modes: numeric, character"
## [5] "Component \"Year\": target is numeric, current is character"
## [6] "Component \"ISBN\": 1 string mismatch"
```

```
all.equal(df_HTML$Year, as.integer(df_XML$Year))
```

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