Handling XML Data in R

XML is a file format which shares both the file format and the data on the World Wide Web, intranets, and elsewhere using standard ASCII text. It stands for Extensible Markup Language (XML). Similar to HTML it contains markup tags. But unlike HTML where the markup tag describes structure of the page, in xml the markup tags describe the meaning of the data contained into he file.

You can read a xml file in R using the "XML" package. This package can be installed using following command.

install.packages("XML")

Input Data

Create a XMl file by copying the below data into a text editor like notepad. Save the file with a **.xml** extension and choosing the file type as **all files(\*.\*)**.

<RECORDS>

<EMPLOYEE>

<ID>1</ID>

<NAME>Rick</NAME>

<SALARY>623.3</SALARY>

<STARTDATE>1/1/2012</STARTDATE>

<DEPT>IT</DEPT>

</EMPLOYEE>

<EMPLOYEE>

<ID>2</ID>

<NAME>Dan</NAME>

<SALARY>515.2</SALARY>

<STARTDATE>9/23/2013</STARTDATE>

<DEPT>Operations</DEPT>

</EMPLOYEE>

<EMPLOYEE>

<ID>3</ID>

<NAME>Michelle</NAME>

<SALARY>611</SALARY>

<STARTDATE>11/15/2014</STARTDATE>

<DEPT>IT</DEPT>

</EMPLOYEE>

<EMPLOYEE>

<ID>4</ID>

<NAME>Ryan</NAME>

<SALARY>729</SALARY>

<STARTDATE>5/11/2014</STARTDATE>

<DEPT>HR</DEPT>

</EMPLOYEE>

<EMPLOYEE>

<ID>5</ID>

<NAME>Gary</NAME>

<SALARY>843.25</SALARY>

<STARTDATE>3/27/2015</STARTDATE>

<DEPT>Finance</DEPT>

</EMPLOYEE>

<EMPLOYEE>

<ID>6</ID>

<NAME>Nina</NAME>

<SALARY>578</SALARY>

<STARTDATE>5/21/2013</STARTDATE>

<DEPT>IT</DEPT>

</EMPLOYEE>

<EMPLOYEE>

<ID>7</ID>

<NAME>Simon</NAME>

<SALARY>632.8</SALARY>

<STARTDATE>7/30/2013</STARTDATE>

<DEPT>Operations</DEPT>

</EMPLOYEE>

<EMPLOYEE>

<ID>8</ID>

<NAME>Guru</NAME>

<SALARY>722.5</SALARY>

<STARTDATE>6/17/2014</STARTDATE>

<DEPT>Finance</DEPT>

</EMPLOYEE>

</RECORDS>

Reading XML File

The xml file is read by R using the function **xmlParse()**. It is stored as a list in R.

# Load the package required to read XML files.

library("XML")

# Also load the other required package.

library("methods")

# Give the input file name to the function.

result <- xmlParse(file = "input.xml")

# Print the result.

print(result)

When we execute the above code, it produces the following result −

1

Rick

623.3

1/1/2012

IT

2

Dan

515.2

9/23/2013

Operations

3

Michelle

611

11/15/2014

IT

4

Ryan

729

5/11/2014

HR

5

Gary

843.25

3/27/2015

Finance

6

Nina

578

5/21/2013

IT

7

Simon

632.8

7/30/2013

Operations

8

Guru

722.5

6/17/2014

Finance

Get Number of Nodes Present in XML File

# Load the packages required to read XML files.

library("XML")

library("methods")

# Give the input file name to the function.

result <- xmlParse(file = "input.xml")

# Exract the root node form the xml file.

rootnode <- xmlRoot(result)

# Find number of nodes in the root.

rootsize <- xmlSize(rootnode)

# Print the result.

print(rootsize)

When we execute the above code, it produces the following result −

output

[1] 8

Details of the First Node

Let's look at the first record of the parsed file. It will give us an idea of the various elements present in the top level node.

# Load the packages required to read XML files.

library("XML")

library("methods")

# Give the input file name to the function.

result <- xmlParse(file = "input.xml")

# Exract the root node form the xml file.

rootnode <- xmlRoot(result)

# Print the result.

print(rootnode[1])

When we execute the above code, it produces the following result −

$EMPLOYEE

1

Rick

623.3

1/1/2012

IT

attr(,"class")

[1] "XMLInternalNodeList" "XMLNodeList"

Get Different Elements of a Node

# Load the packages required to read XML files.

library("XML")

library("methods")

# Give the input file name to the function.

result <- xmlParse(file = "input.xml")

# Exract the root node form the xml file.

rootnode <- xmlRoot(result)

# Get the first element of the first node.

print(rootnode[[1]][[1]])

# Get the fifth element of the first node.

print(rootnode[[1]][[5]])

# Get the second element of the third node.

print(rootnode[[3]][[2]])

When we execute the above code, it produces the following result −

1

IT

Michelle

XML to Data Frame

To handle the data effectively in large files we read the data in the xml file as a data frame. Then process the data frame for data analysis.

# Load the packages required to read XML files.

library("XML")

library("methods")

# Convert the input xml file to a data frame.

xmldataframe <- xmlToDataFrame("input.xml")

print(xmldataframe)

When we execute the above code, it produces the following result −

ID NAME SALARY STARTDATE DEPT

1 1 Rick 623.30 2012-01-01 IT

2 2 Dan 515.20 2013-09-23 Operations

3 3 Michelle 611.00 2014-11-15 IT

4 4 Ryan 729.00 2014-05-11 HR

5 NA Gary 843.25 2015-03-27 Finance

6 6 Nina 578.00 2013-05-21 IT

7 7 Simon 632.80 2013-07-30 Operations

8 8 Guru 722.50 2014-06-17 Finance

As the data is now available as a dataframe we can use data frame related function to read and manipulate the file.