R news and tutorials contributed by (552) R bloggers

- Home
- About
- add your blog!
- Contact us
- RSS
- R jobs

Welcome!

Here you will find daily **news and tutorials about R**, contributed by over 555 bloggers. There are many ways to **follow us -**

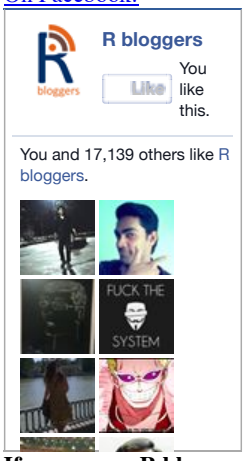
By e-mail:

Your e-mail here
Subscribe
13717 readers
BY FEEDBURNER

On Twitter:

Follow @rbloggers 10.6K f

On Facebook:



If you are an R blogger yourself you are invited to add your own R content feed to this site (**Non-English** R bloggers should add themselves- here)

Search & Hit Enter

Jobs for R-users

- Applied Researcher in Algorithmic Research
- Sr. Data Scientist in the Financial

Industry
- R programmer / research analyst
- Sr. Java Software Engineer (m/f) – Data Mining – IT Startup
- Executive Director of Institutional Research

Popular Searches

- heatmap
- hadoop
- web scraping
- maps
- pca
- twitter
- ggplot
- finance
- shiny
- excel
- ggplot2
- LaTex
- googlevis
- boxplot
- trading
- animation
- eclipse
- RStudio
- knitr
- quantmod
- time series
- tutorial
- how to import image file to R
- alt=
- market research
- gis
- rattle
- ecdf
- rcmdr
- sweave

Recent Posts

- 8 new R jobs (for Augest 25th 2014)
- New in OpenCPU 1.4.4: session namespaces
- Forecasting with R in WA
- R: Stem (Pre-Processed) Text Blocks
- JAGS and Stan
- Computational Actuarial Science, with R
- VCI — The Value Charts Indicator
- R OOP – a little

privacy please?
- An adverse consequence of fitting "maximal" linear mixed models
- More ESA 2014 Program Text-Mining: Topics as Communities
- Entering the field as a data scientist with certification
- Pro Grammar and Devel Hoper
- Running “Native” Data Wrangling Applications in the Browser – IPython Notebooks (and R?) in Chrome
- stringdist 0.8: now with soundex
- RcppEigen 0.3.2.2.0

Other sites

- SAS blogs
- Jobs for R-users
- Statistics of Israel

Connecting to a MongoDB database from R using Java

September 24, 2010
By nsaunders

(This article was first published on **What You're Doing Is Rather Desperate » R**, and kindly contributed to R-bloggers)

It would be nice if there were an R package, along the lines of RMySQL, for MongoDB. For now there is not – so, how best to get data from a MongoDB database into R?

One option is to retrieve JSON via the MongoDB REST interface and parse it using the *rjson* package. Assuming, for example, that you have retrieved your CiteULike collection in JSON format from this URL:

```
http://www.citeulike.org/json/user/neils
```

- and saved it to a database named *citeulike* in a collection named *articles*, you can fetch the first 5 articles into R like so:

```
library(RCurl)
library(rjson)

db <- "http://localhost:28017/citeulike/articl
articles <- fromJSON(getURL(db))
articles$rows[[1]]$title
# [1] "A computational genomics pipeline for p
```

That works, but you may not want to use the MongoDB REST interface: for example, it may be slow for large queries or there might

be security concerns.

MongoDB has both C and Java drivers. R has packages that interface with these languages: .C/.Call and rJava, respectively. My only problem is that I can write what I know about C and Java on the back of a postage stamp.

Not to be deterred, I took the approach that has served me well my whole professional life: wing it, using what I could glean from Google searches and the Web. In the end, using Java in R to connect with MongoDB was surprisingly easy. Here's a basic how-to.

I'll assume that MongoDB is installed and running on your machine. Packages for Ubuntu/Debian can be obtained here.

1. Install R packages
You'll need rJava and rjson. The latter was a simple *install.packages("rjson")* from the R console. The former gave me some problems so as I use Ubuntu, I went with *sudo apt-get install r-cran-rjava*. That should also install the necessary dependencies, including a JDK if you don't already have one.

2. Install the MongoDB Java driver
Create a directory, *e.g. ~/mongodb/java*, change into it and grab the latest driver from GitHub. I renamed the file to *mongo.jar*. Having no idea what to do with it, I searched and discovered this guide. I ran:

```
jar xf mongo.jar
# generates these directories
com  git-hash  META-INF  mongo.jar  org
```

The Java class files are located in *com/mongodb*.

3. Experiment with rJava
Still in *~/mongodb/java*, I started an R console and loaded the libraries:

```
library(rJava)
library(rjson)
```

Next, I added the MongoDB classes to the classpath:

```
.jinit()
.jaddClassPath("~/mongodb/java/mongo.jar")
```

The next step was to consult the MongoDB Java tutorial and try to figure out how to convert "normal" Java syntax to rJava. First, rJava has no *import*, so you create a new Mongo object like this:

```
m <- .jnew("com/mongodb/Mongo", "localhost")
print(m)
# [1] "Java-Object{com.mongodb.Mongo@c2ea3f}"
```

OK – that seems to have worked; we have a Java object of class *Mongo*, connected to the server on *localhost*.
You can see the available methods like this:

```
.jmethods(m)
# result
 [1] "public com.mongodb.DB com.mongodb.Mongo
 [2] "public java.util.List com.mongodb.Mongo
 [3] "public void com.mongodb.Mongo.dropDatab
 [4] "public java.lang.String com.mongodb.Mon
 [5] "public java.lang.String com.mongodb.Mon
 [6] "public java.util.List com.mongodb.Mongo
 [7] "public void com.mongodb.Mongo.setWriteC
 [8] "public com.mongodb.WriteConcern com.mon
 [9] "public com.mongodb.ServerAddress com.mo
[10] "public void com.mongodb.Mongo.close()"
[11] "public static com.mongodb.DB com.mongod
[12] "public java.lang.String com.mongodb.Mon
```

```
[13] "public final native void java.lang.Obje
[14] "public final void java.lang.Object.wait
[15] "public final void java.lang.Object.wait
[16] "public boolean java.lang.Object.equals(
[17] "public java.lang.String java.lang.Objec
[18] "public native int java.lang.Object.hash
[19] "public final native java.lang.Class jav
[20] "public final native void java.lang.Obje
[21] "public final native void java.lang.Obje
```

As a non-Java programmer, that means very little to me. Instead, I typed *m$*, hit the tab key a couple of times and saw this:

```
m$MAJOR_VERSION       m$dropDatabase(       m$
m$MINOR_VERSION       m$debugString()       m$
m$getDB(              m$getConnectPoint()   m$
m$getDatabaseNames()  m$getAllAddress()     m$
```

That's much more useful – I recognise those methods. Let's try connecting with the *citeulike* database:

```
db <- m$getDB("citeulike")
print(db)
# [1] "Java-Object{citeulike}"
```

Progress, no errors, it's all good. Using the same approach – type *db$* and hit tab, I saw this:

```
db$requestStart()             db$getCollection
db$requestDone()              db$doEval(
db$requestEnsureConnection()  db$eval(
db$dropDatabase()             db$getStats()
db$setWriteConcern(           db$getCollection
db$getWriteConcern()          db$collectionExi
db$getCollection(             db$resetIndexCac
db$createCollection(          db$getLastError(
```

Which led me to believe that I could access the *articles* collection like this:

```
col <- db$getCollection("articles")
print(col)
# [1] "Java-Object{articles}"
```

You get the idea. The Java methods follow the names of the MongoDB shell commands. Let's fetch the first article:

```
article <- col$findOne()
article <- article$toString
```

Success! The *toString()* method converts the article to a JSON string. Now all that's left is to get that into an R data structure:

```
article <- fromJSON(article)
article$title
# [1] "A computational genomics pipeline for
article$authors
#  [1] "Andrey O. Kislyuk"    "Lee S. Katz"
 [4] "Matthew S. Hagen"     "Andrew B. Conley
 [7] "Viswateja Nelakuditi" "Jay C. Humphrey"
[10] "Dhwani Govil"         "Raydel D. Mair"
[13] "Maria L. Tondella"    "Brian H. Harcour
[16] "I. King Jordan"
```

Let the statistical analysis of your CiteULike library (or any other data from MongoDB) begin.

Filed under: computing, programming, R, research diary, statistics
Tagged: java, mongodb, rjava, rjson Comments: 3 Tweet it!

e a comment for the author, please follow the link and comment on his blog: **What You're Doing Is Rather Desperate » R**.

Tags: computing, java, mongodb, programming, R, research diary, rjava, rjson, statistics

Comments are closed.

Top 3 Posts from the past 2 days

- Using apply, sapply, lapply in R
- dplyr: A gamechanger for data manipulation in R
- Installing R packages

Top 9 articles of the week

1. dplyr: A gamechanger for data manipulation in R
2. Using apply, sapply, lapply in R
3. R Passes SPSS in Scholarly Use, Stata Growing Rapidly
4. Installing R packages
5. Do your "data janitor work" like a boss with dplyr
6. Data Cleaning is a critical part of the Data Science process
7. googleVis 0.5.5 released
8. 21 R navigation tools
9. Box-plot with R – Tutorial

Sponsors