

# Package ‘jSonarR’

February 20, 2015

**Type** Package

**Title** jSonar Analytics Platform API for R

**Version** 1.1.1

**Date** 2014-07-02

**Author** jSonar Inc.

**Maintainer** Dylan McCall <dylan@jsonar.com>

**Description** This package enables users to access MongoDB by running queries and returning their results in R data frames. Usually, data in MongoDB is only available in the form of a JSON document. jSonarR uses data processing and conversion capabilities in the jSonar Analytics Platform and the JSON Studio Gateway (<http://www.jsonstudio.com>), to convert it to a tabular format which is easy to use with existing R packages.

**Depends** R (>= 2.12.1), RCurl, jsonlite, methods

**SystemRequirements** MongoDB, JSON Studio

**Copyright** jSonar Inc. <<http://www.jsonar.com>>

**License** AGPL-3

**URL** <http://www.jsonstudio.com/>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-09-26 18:50:29

## R topics documented:

jSonarR	2
new.SonarConnection	2
sonarAgg	4
sonarCSV	5
sonarFind	6
sonarJSON	7
<b>Index</b>	<b>8</b>

jSonarR

*jSonar Analytics Platform API for R*

---

**Description**

This package enables users to access MongoDB by running queries and returning their results in R data frames. Usually, data in MongoDB is only available in the form of a JSON document. jSonarR uses data processing and conversion capabilities in the jSonar Analytics Platform and the JSON Studio Gateway (<http://www.jsonstudio.com>), to convert it to a tabular format which is easy to use with existing R packages.

**Details**

To use jSonarR, you must have access to a server running JSON Studio. Create a connection using `new.SonarConnection`. Now you can run a saved query against a collection in the database using the connection object and `sonarAgg` or `sonarFind`.

**See Also**

MongoDB <http://www.mongodb.org>

JSON Studio <http://www.jsonstudio.com>

**Examples**

```
connection <- new.SonarConnection('https://example.com', 'localhost', 'test')

ny_by_day <- sonarAgg(connection, 'delays_by_day', 'NYCFlights')
summary(ny_by_day)

tx_to_co <- sonarFind(connection, 'flights_to', 'TXFlights',
  bind=list(state="CO"),
  colClasses=c(DAY_OF_MONTH='factor', DEST_AIRPORT_ID='factor'))
summary(tx_to_co$DEST_AIRPORT_ID)
```

---

`new.SonarConnection`     *JSON Studio connection*

---

**Description**

Create a connection to a Mongo database through JSON Studio

**Usage**

```
new.SonarConnection(url, host, db, port = 27017, username = NULL,
  pwd = NULL, sdb = NULL, ssl = FALSE, anyCert = FALSE, krb = FALSE,
  mapCredentials = FALSE, secondaryPref = FALSE)
```

## Arguments

url	the url where JSON Studio can be accessed
host	the hostname of the Mongo server, as it would be entered from the JSON Studio login screen
db	the name of the database you intend to access
port	the port number where Mongo is running
username	a username to log in to the database, if necessary
pwd	a password to log in to the database, if necessary
sdb	the name of a database to store JSON Studio-related collections
ssl	TRUE to connect using SSL
anyCert	TRUE to accept any SSL certificate
krb	TRUE to authenticate using Kerberos
mapCredentials	TRUE to map credentials to a functional user account with which to access data
secondaryPref	TRUE to allow connecting to a secondary of a replica set and prefer a secondary if the host value passed in is a replica set

## Details

This function returns a `SonarConnection` object which can be used with `sonarFind` and `sonarAgg` to query a Mongo database.

The parameters for this function are explained in greater detail in the JSON Studio help page *Using the Gateway*.

## Value

A `SonarConnection` object to connect to the given Mongo database through JSON Studio, which can be used with `sonarFind` or `sonarAgg`.

## See Also

[http://jsonstudio.com/wp-content/uploads/2014/04/manual141/\\_build/html/index.html](http://jsonstudio.com/wp-content/uploads/2014/04/manual141/_build/html/index.html)

Other connection: `sonarAgg`; `sonarCSV`; `sonarFind`; `sonarJSON`

## Examples

```
con <- new.SonarConnection('https://localhost:8443', 'localhost', 'test')
```

---

sonarAgg

*Run a saved aggregation pipeline*


---

### Description

Execute an aggregation pipeline which has been saved and published in JSON Studio Analytics, and get the result in a data frame.

### Usage

```
sonarAgg(connection, queryName, queryCol, bind = list(), limit = NULL,
  idCol = "_id", publishedBy = NULL, colClasses = NA)
```

### Arguments

connection	a SonarConnection object created with <a href="#">new.SonarConnection</a>
queryName	the name of the saved query to execute
queryCol	a collection in the database to use with the query
bind	a list of bind variables and their values
limit	the maximum number of results to return
idCol	the name of a field which uniquely identifies each document. This will be used for the row names in the returned data frame. The default is X_id, which is the name of Mongo's _id field (adjusted by <a href="#">make.names</a> ).
publishedBy	the name of the user who we expect published the API
colClasses	a list of column names and their respective classes, as used in <a href="#">read.csv</a> . This may be necessary if some columns' types are not being detected automatically.

### Details

The parameters for this function are explained in greater detail in the JSON Studio help page *Using the Gateway*.

### See Also

[http://jsonstudio.com/wp-content/uploads/2014/04/manual141/\\_build/html/index.html](http://jsonstudio.com/wp-content/uploads/2014/04/manual141/_build/html/index.html)

Other connection: [new.SonarConnection](#); [sonarCSV](#); [sonarFind](#); [sonarJSON](#)

Other csv: [sonarCSV](#); [sonarFind](#)

### Examples

```
connection <- new.SonarConnection('https://example.com', 'localhost', 'test')

ny_by_day <- sonarAgg(connection, 'delays_by_day', 'NYCFlights')
cor(ny_by_day$X_avg_ArrDelay, ny_by_day$X_avg_AirTime)
```

---

sonarCSV *Get a CSV document for a saved query*

---

### Description

Execute a find query which has been saved and published in JSON Studio Finder, and get the response in an R data frame that represents Mongo's data in tabular form.

### Usage

```
sonarCSV(connection, queryName, queryCol, type, bind = list(), limit = NULL,
  idCol = "_id", publishedBy = NULL, colClasses = NA)
```

### Arguments

connection	a SonarConnection object created with <a href="#">new.SonarConnection</a>
queryName	the name of the saved query to execute
queryCol	a collection in the database to use with the query
type	the type of query to execute ('agg' or 'find')
bind	a list of bind variables and their values
limit	the maximum number of results to return
idCol	the name of a field which uniquely identifies each document. This will be used for the row names in the returned data frame. The default is X_id, which is the name of Mongo's _id field (adjusted by <a href="#">make.names</a> ).
publishedBy	the name of the user who we expect published the API
colClasses	a list of column names and their respective classes, as used in <a href="#">read.csv</a> . This may be necessary if some columns' types are not being detected automatically.

### Details

The parameters for this function are explained in greater detail in the JSON Studio help page *Using the Gateway*.

### See Also

[http://jsonstudio.com/wp-content/uploads/2014/04/manual141/\\_build/html/index.html](http://jsonstudio.com/wp-content/uploads/2014/04/manual141/_build/html/index.html)

Other connection: [new.SonarConnection](#); [sonarAgg](#); [sonarFind](#); [sonarJSON](#)

Other csv: [sonarAgg](#); [sonarFind](#)

### Examples

```
connection <- new.SonarConnection('https://example.com', 'localhost', 'test')

delays <- sonarCSV(connection, 'delayed_flights', 'WAFlights', type='find')
cor(delays$ACTUAL_ELAPSED_TIME, delays$WEATHER_DELAY)
```

---

sonarFind	<i>Run a saved find query</i>
-----------	-------------------------------

---

### Description

Execute a find query which has been saved and published in JSON Studio Finder, and get the result in a data frame.

### Usage

```
sonarFind(connection, queryName, queryCol, bind = list(), limit = NULL,
  idCol = "_id", publishedBy = NULL, colClasses = NA)
```

### Arguments

connection	a SonarConnection object created with <a href="#">new.SonarConnection</a>
queryName	the name of the saved query to execute
queryCol	a collection in the database to use with the query
bind	a list of bind variables and their values
limit	the maximum number of results to return
idCol	the name of a field which uniquely identifies each document. This will be used for the row names in the returned data frame. The default is X_id, which is the name of Mongo's _id field (adjusted by <a href="#">make.names</a> ).
publishedBy	the name of the user who we expect published the API
colClasses	a list of column names and their respective classes, as used in <a href="#">read.csv</a> . This may be necessary if some columns' types are not being detected automatically.

### Details

The parameters for this function are explained in greater detail in the JSON Studio help page *Using the Gateway*.

### See Also

[http://jsonstudio.com/wp-content/uploads/2014/04/manual141/\\_build/html/index.html](http://jsonstudio.com/wp-content/uploads/2014/04/manual141/_build/html/index.html)

Other connection: [new.SonarConnection](#); [sonarAgg](#); [sonarCSV](#); [sonarJSON](#)

Other csv: [sonarAgg](#); [sonarCSV](#)

### Examples

```
connection <- new.SonarConnection('https://example.com', 'localhost', 'test')

delays <- sonarFind(connection, 'delayed_flights', 'WAFlights')
cor(delays$ACTUAL_ELAPSED_TIME, delays$WEATHER_DELAY)
```

```
tx_to_co <- sonarFind(connection, 'flights_to', 'TXFlights',
  bind=list(state="CO"),
  colClasses=c(DAY_OF_MONTH='factor', DEST_AIRPORT_ID='factor'))
hist(tx_to_co$ACTUAL_ELAPSED_TIME)
```

---

sonarJSON

*Get a JSON document for a saved query*

---

## Description

Execute a query which has been saved and published in JSON Studio Finder, and get the response in an R object that is structured like a JSON document. This object is generated by the `jsonlite` package.

## Usage

```
sonarJSON(connection, queryName, queryCol, type, bind = list(),
  limit = NULL, publishedBy = NULL)
```

## Arguments

<code>connection</code>	a <code>SonarConnection</code> object created with <a href="#">new.SonarConnection</a>
<code>queryName</code>	the name of the saved query to execute
<code>queryCol</code>	a collection in the database to use with the query
<code>type</code>	the type of query to execute ('agg' or 'find')
<code>bind</code>	a list of bind variables and their values
<code>limit</code>	the maximum number of results to return
<code>publishedBy</code>	the name of the user who we expect published the API

## Details

The parameters for this function are explained in greater detail in the JSON Studio help page *Using the Gateway*.

## See Also

[http://jsonstudio.com/wp-content/uploads/2014/04/manual141/\\_build/html/index.html](http://jsonstudio.com/wp-content/uploads/2014/04/manual141/_build/html/index.html)

Other connection: [new.SonarConnection](#); [sonarAgg](#); [sonarCSV](#); [sonarFind](#)

## Examples

```
connection <- new.SonarConnection('https://example.com', 'localhost', 'test')

delays <- sonarJSON(connection, 'delayed_flights', 'ExampleFlights', type='find', limit=5)
summary(delays$Origin$city)
```

# Index

## \*Topic **connection**

- jSonarR, [2](#)
- new.SonarConnection, [2](#)

## \*Topic **database**

- jSonarR, [2](#)
- new.SonarConnection, [2](#)
- sonarAgg, [4](#)
- sonarCSV, [5](#)
- sonarFind, [6](#)
- sonarJSON, [7](#)

jSonarR, [2](#)

jSonarR-package (jSonarR), [2](#)

make.names, [4–6](#)

new.SonarConnection, [2, 2, 4–7](#)

read.csv, [4–6](#)

sonarAgg, [2, 3, 4, 5–7](#)

sonarCSV, [3, 4, 5, 6, 7](#)

sonarFind, [2–5, 6, 7](#)

sonarJSON, [3–6, 7](#)