

# Package ‘nodbi’

April 13, 2019

**Title** 'NoSQL' Database Connector

**Description** Simplified document database manipulation and analysis, including support for many 'NoSQL' databases, including document databases ('Elasticsearch', 'CouchDB', 'MongoDB'), 'key-value' databases ('Redis'), and other 'NoSQL' types ('etcd').

**Version** 0.2.0

**License** MIT + file LICENSE

**LazyData** true

**URL** <https://github.com/ropensci/nodbi>

**BugReports** <https://github.com/ropensci/nodbi/issues>

**Depends** R (>= 2.10)

**Encoding** UTF-8

**Language** en-US

**Imports** data.table, jsonlite

**Suggests** sofa (>= 0.3.0), etseed (>= 0.1.0), elastic (>= 1.0.0), mongolite (>= 1.6), redux, testthat

**RoxygenNote** 6.1.1

**X-schema.org-applicationCategory** Databases

**X-schema.org-keywords** database, MongoDB, Redis, Elasticsearch, CouchDB, NoSQL, JSON, documents

**X-schema.org-isPartOf** <https://ropensci.org>

**NeedsCompilation** no

**Author** Scott Chamberlain [aut, cre] (<<https://orcid.org/0000-0003-1444-9135>>),  
Rich FitzJohn [aut],  
Jeroen Ooms [aut]

**Maintainer** Scott Chamberlain <[sckott@protonmail.com](mailto:sckott@protonmail.com)>

**Repository** CRAN

**Date/Publication** 2019-04-13 04:22:49 UTC

**R topics documented:**

nodbi-package . . . . .	2
diamonds . . . . .	2
docdb_create . . . . .	3
docdb_delete . . . . .	4
docdb_exists . . . . .	5
docdb_get . . . . .	7
docdb_query . . . . .	8
docdb_update . . . . .	9
src . . . . .	10
src_couchdb . . . . .	11
src_elastic . . . . .	12
src_etcd . . . . .	13
src_mongo . . . . .	13
src_redis . . . . .	14
<b>Index</b>	<b>15</b>

---

nodbi-package	<i>Document database connector</i>
---------------	------------------------------------

---

**Description**

Supports NoSQL databases (Elasticsearch, CouchDB, MongoDB), key-value databases (Redis), and other NoSQL types (etcd).

**Author(s)**

Scott Chamberlain <sckott@protonmail.com>  
 Rich FitzJohn <rich.fitzjohn@gmail.com>  
 Jeroen Ooms <jeroen.ooms@stat.ucla.edu>

---

diamonds	<i>diamonds</i>
----------	-----------------

---

**Description**

diamonds

**Format**

A data frame with 53940 rows and 10 variables:

- price price in US dollars (\\$326-\\$18,823)
- carat weight of the diamond (0.2-5.01)
- cut quality of the cut (Fair, Good, Very Good, Premium, Ideal)
- color diamond colour, from J (worst) to D (best)
- clarity a measurement of how clear the diamond is (I1 (worst), SI1, SI2, VS1, VS2, VVS1, VVS2, IF (best))
- x length in mm (0-10.74)
- y width in mm (0-58.9)
- z depth in mm (0-31.8)
- depth total depth percentage =  $z / \text{mean}(x, y) = 2 * z / (x + y)$  (43-79)
- table width of top of diamond relative to widest point (43-95)

**Source**

from **ggplot2**

---

docdb_create	<i>Create documents</i>
--------------	-------------------------

---

**Description**

Create documents

**Usage**

```
docdb_create(src, key, value, ...)
```

**Arguments**

src	source object, result of call to an <a href="#">src</a> function
key	(character) A key. ignored for mongo
value	(data.frame) A single data.frame
...	Ignored

**Details**

Note that with etcd, you have to prefix a key with a forward slash.

**Examples**

```

## Not run:
# CouchDB
src <- src_couchdb()
docdb_create(src, key="mtcars2", value=mtcars)
docdb_get(src, "mtcars2")

# etcd
# src <- src_etcd()
# docdb_create(src, key = "/newmtcars7", value = mtcars)
# docdb_get(src, "/newmtcars7")

# Elasticsearch
src <- src_elastic()
docdb_create(src, key = "mtcars", value = mtcars)
docdb_create(src, key = "iris", value = iris)
docdb_create(src, key = "diamonds_small", value = diamonds[1:3000L,])

# Redis
src <- src_redis()
docdb_create(src, key = "mtcars", value = mtcars)
docdb_get(src, "mtcars")
docdb_delete(src, "mtcars")

# MongoDB
src <- src_mongo()
docdb_create(src, key = "mtcars", value = mtcars)
docdb_get(src, "mtcars")

## End(Not run)

```

---

docdb\_delete

*Delete documents*


---

**Description**

Delete documents

**Usage**

```
docdb_delete(src, key, ...)
```

**Arguments**

src	source object, result of call to src, an object of class docdb_src
key	(character) A key. ignored for mongo
...	Ignored for now

## Details

Note that with etcd, you have to prefix a key with a forward slash.

## Examples

```
## Not run:
# couchdb
(src <- src_couchdb())
docdb_create(src, "mtcars", mtcars)
docdb_get(src, "mtcars")
docdb_delete(src, "mtcars")

# etcd
# src <- src_etcd()
# docdb_create(src, "/iris", iris)
# docdb_get(src, "/iris")
# docdb_delete(src, "/iris")

# elasticsearch
src <- src_elastic()
docdb_create(src, "iris", iris)
docdb_get(src, "iris")
docdb_delete(src, "iris")

# Redis
src <- src_redis()
docdb_create(src, key = "mtcars", value = mtcars)
docdb_get(src, "mtcars")
docdb_delete(src, "mtcars")

# mongo
src <- src_mongo("stuff")
docdb_create(src, "iris", iris)
docdb_get(src, "iris")
docdb_delete(src)

## End(Not run)
```

---

docdb\_exists

*Check if a database exists*

---

## Description

Check if a database exists

## Usage

```
docdb_exists(src, key, ...)
```

**Arguments**

src	source object, result of call to src
key	(character) A key. ignored for mongo
...	Ignored for now

**Details**

Note that with etcd, you have to prefix a key with a forward slash.

**Value**

logical, TRUE or FALSE

**Note**

no docdb\_exists method for MongoDB at this time

**Examples**

```
## Not run:
# CouchDB
(src <- src_couchdb())
docout <- docdb_create(src, key = "mtcars2", value = mtcars)
docdb_exists(src, "mtcars2")
docdb_exists(src, "asdfadf")

# etcd
# src <- src_etcd()
# docdb_exists(src, "/hello")
# docdb_exists(src, "/asdfadsf")

# Elasticsearch
(src <- src_elastic())
if (docdb_exists(src, "iris")) docdb_delete(src, "iris")
docdb_exists(src, "iris")
docdb_create(src, "iris", iris)
docdb_exists(src, "iris")
docdb_exists(src, "adfadf")

# Redis
(src <- src_redis())
docdb_create(src, "mtcars", mtcars)
docdb_exists(src, "mtcars")
docdb_exists(src, "asdfadf")

## End(Not run)
```

---

docdb_get	<i>Get documents</i>
-----------	----------------------

---

## Description

Get documents

## Usage

```
docdb_get(src, key, limit = NULL, ...)
```

## Arguments

src	source object, result of call to src
key	(character) A key. ignored for mongo
limit	(integer) number of records/rows to return. by default not passed, so you get all results. Only works for CouchDB, Elasticsearch and MongoDB; ignored for others
...	passed on to functions: <ul style="list-style-type: none"><li>• CouchDB: passed to <code>sofa::db_alldocs()</code></li><li>• etcd: passed to the <code>\$key()</code> method</li><li>• Elasticsearch: passed to <code>elastic::Search()</code></li><li>• Redis: ignored</li><li>• MongoDB: ignored</li></ul>

## Details

Note that with etcd, you have to prefix a key with a forward slash.

## Examples

```
## Not run:
# CouchDB
src <- src_couchdb()
docout <- docdb_create(src, key = "mtcars2", value = mtcars)
docdb_get(src, "mtcars2")
docdb_get(src, "mtcars2", limit = 5)

# etcd
# src <- src_etcd()
# docdb_create(src, "/hello", mtcars)
# docdb_get(src, "/hello")

# Elasticsearch
src <- src_elastic()
docdb_create(src, "iris", iris)
docdb_get(src, "iris")
```

```

docdb_create(src, "d2", diamonds)
docdb_get(src, "d2", limit = 1010)

# Redis
src <- src_redis()
docdb_create(src, "mtcars", mtcars)
docdb_get(src, "mtcars")

# Mongo
src <- src_mongo()
docdb_create(src, "mtcars", mtcars)
docdb_get(src, "mtcars")
docdb_get(src, "mtcars", limit = 4)

## End(Not run)

```

---

docdb\_query

*Get documents with a filtering query*


---

## Description

Get documents with a filtering query

## Usage

```
docdb_query(src, key, query, ...)
```

## Arguments

src	source object, result of call to src
key	(character) A key. ignored for mongo
query	various. see Query section below.
...	Additional named parameters passed on to each package: <ul style="list-style-type: none"> <li>• CouchDB: passed on to <code>sofa::db_query()</code></li> <li>• Elasticsearch: passed on to <code>elastic::Search()</code></li> <li>• MongoDB: passed on to the <code>\$find</code> method of <code>mongolite</code></li> </ul>

## Details

Note that with etcd, you have to prefix a key with a forward slash.



**What is expected for each source**

- CouchDB: a list, see docs for `sofa::db_query()`
- Elasticsearch: query parameters, see `elastic::Search()`; passed to the query parameter of `elastic::Search`, thus performs a URI based search where the query is passed in the URI instead of the body. In theory you can instead pass in a JSON or list to the body parameter, but if you want to do complicated Elasticsearch queries you may be better off using `elastic` package directly
- MongoDB: query parameters, see `mongolite` docs for help with searches

**Not supported yet**

- Etc
- Redis

**Examples**

```
## Not run:
# CouchDB
(src <- src_couchdb())
if (docdb_exists(src, "mtcars2")) docdb_delete(src, "mtcars2")
invisible(docdb_create(src, key = "mtcars2", value = mtcars))
docdb_exists(src, "mtcars2")
(query <- list(cyl = list("$gt" = 6)))
docdb_query(src, "mtcars2", query = query)

# Elasticsearch
src <- src_elastic()
if (docdb_exists(src, "iris")) docdb_delete(src, "iris")
docdb_create(src, "iris", iris)
docdb_exists(src, "iris")
docdb_query(src, "iris", query = "setosa")
docdb_query(src, "iris", query = "1.5")
docdb_query(src, "iris", query = "Petal.Width:1.5")

# Mongo
src <- src_mongo()
if (docdb_exists(src, "mtcars")) docdb_delete(src, "mtcars")
docdb_create(src, "mtcars", mtcars)
docdb_query(src, query = '{"mpg":21}')
docdb_query(src, query = '{"mpg":21}', fields = '{"mpg":1, "cyl":1}')
```

## End(Not run)

---

docdb\_update

*Update documents*


---

**Description**

Update documents

**Usage**

```
docdb_update(src, key, value, ...)
```

**Arguments**

src	source object, result of call to src
key	(character) A key. ignored for mongo
value	(data.frame) A single data.frame
...	Ignored

**Details**

Only CouchDB supported for now

**Examples**

```
## Not run:
# CouchDB
src <- src_couchdb()
docdb_create(src, "mtcars2", mtcars)
docdb_get(src, "mtcars2")

mtcars$letter <- sample(letters, NROW(mtcars), replace = TRUE)
invisible(docdb_update(src, "mtcars2", mtcars))
docdb_get(src, "mtcars2")

## End(Not run)
```

---

 src

---

*Setup database connections*


---

**Description**

Setup database connections

**Details**

There is a `src_*()` function to setup a connection to each of the database backends. Each has their own unique set of parameters.

- MongoDB - [src\\_mongo\(\)](#)
- CouchDB - [src\\_couchdb\(\)](#)
- etcd - [src\\_etcd\(\)](#)
- Elasticsearch - [src\\_elastic\(\)](#)
- Redis - [src\\_redis\(\)](#)

Documentation details for each database:

- MongoDB - <https://docs.mongodb.com/>
- CouchDB - <http://docs.couchdb.org/>
- etcd - <https://coreos.com/etcd/docs/latest/getting-started-with-etcd.html>
- Elasticsearch - <https://www.elastic.co/guide/en/elasticsearch/reference/current>
- Redis - <https://redis.io/documentation>

---

src\_couchdb

*Setup a CouchDB database connection*

---

## Description

Setup a CouchDB database connection

## Usage

```
src_couchdb(host = "127.0.0.1", port = 5984, path = NULL,
            transport = "http", user = NULL, pwd = NULL, headers = NULL)
```

## Arguments

host	(character) host value, default: 127.0.0.1
port	(integer/numeric) Port. Remember that if you don't want a port set, set this parameter to NULL. Default: 5984
path	(character) context path that is appended to the end of the url. e.g., bar in http://foo.com/bar. Default: NULL, ignored
transport	(character) http or https. Default: http
user	(character) Username, if any
pwd	(character) Password, if any
headers	(list) list of named headers

## Details

uses **sofa** under the hood; uses [sofa::Cushion](#) for connecting

## Examples

```
## Not run:
src_couchdb()

## End(Not run)
```

---

 src\_elastic

*Setup an Elasticsearch database connection*


---

### Description

Setup an Elasticsearch database connection

### Usage

```
src_elastic(host = "127.0.0.1", port = 9200, path = NULL,
            transport_schema = "http", user = NULL, pwd = NULL,
            force = FALSE, ...)
```

### Arguments

host	(character) the base url, defaults to localhost (http://127.0.0.1)
port	(character) port to connect to, defaults to 9200 (optional)
path	(character) context path that is appended to the end of the url. Default: NULL, ignored
transport_schema	(character) http or https. Default: http
user	(character) User name, if required for the connection. You can specify, but ignored for now.
pwd	(character) Password, if required for the connection. You can specify, but ignored for now.
force	(logical) Force re-load of connection details
...	Further args passed on to <a href="#">elastic::connect()</a>

### Details

uses **elastic** under the hood; uses [elastic::connect\(\)](#) for connecting

### Examples

```
## Not run:
src_elastic()

## End(Not run)
```

---

src_etcd	<i>Setup an Etcd database connection</i>
----------	--

---

**Description**

Setup an Etcd database connection

**Usage**

```
src_etcd(host = "127.0.0.1", port = 2379, api_version = "v2",  
         allow_redirect = TRUE, scheme = "http")
```

**Arguments**

host	(character) Host url. Deafault: 127.0.0.1
port	(integer/numeric) Port. Default: 2379
api_version	(character) etcd API version. Default: 'v2'
allow_redirect	(logical) Allow redirects? Default: TRUE
scheme	(character) http scheme, one of http or https. Default: http

**Details**

uses **etseed** under the hood; uses `etseed::etcd()` for connecting

**Examples**

```
## Not run:  
(src <- src_etcd())  
class(src)  
  
## End(Not run)
```

---

src_mongo	<i>Setup a mongoDB database connection</i>
-----------	--

---

**Description**

Setup a mongoDB database connection

**Usage**

```
src_mongo(collection = "test", db = "test",  
          url = "mongodb://localhost", ...)
```

**Arguments**

collection	(character) name of collection
db	(character) name of database
url	(character) address of the mongodb server in mongo connection string URI format.
...	additional named params passed on to <a href="#">mongolite::mongo</a>

**Details**

uses **monoglite** under the hood; uses [mongolite::mongo\(\)](#) for connecting

**Examples**

```
## Not run:
(con <- src_mongo())
print(con)

## End(Not run)
```

---

src\_redis

*Setup an Redis database connection*


---

**Description**

Setup an Redis database connection

**Usage**

```
src_redis(...)
```

**Arguments**

...	Redis connection configuration options. See <a href="#">redux::redis_config</a> for more information
-----	--

**Details**

uses **redux** under the hood; uses [redux::hiredis\(\)](#) for connecting

**Examples**

```
## Not run:
(con <- src_redis())
class(con)

## End(Not run)
```

# Index

## \*Topic **data**

diamonds, [2](#)

## \*Topic **package**

nodbi-package, [2](#)

diamonds, [2](#)

docdb\_create, [3](#)

docdb\_delete, [4](#)

docdb\_exists, [5](#)

docdb\_get, [7](#)

docdb\_query, [8](#)

docdb\_update, [9](#)

elastic::connect(), [12](#)

elastic::Search(), [7-9](#)

etseed::etcd(), [13](#)

mongolite::mongo, [14](#)

mongolite::mongo(), [14](#)

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

nodbi-package, [2](#)

redux::hireis(), [14](#)

redux::redis\_config, [14](#)

sofa::Cushion, [11](#)

sofa::db\_alldocs(), [7](#)

sofa::db\_query(), [8, 9](#)

src, [3, 10](#)

src\_couchdb, [11](#)

src\_couchdb(), [10](#)

src\_elastic, [12](#)

src\_elastic(), [10](#)

src\_etcd, [13](#)

src\_etcd(), [10](#)

src\_mongo, [13](#)

src\_mongo(), [10](#)

src\_redis, [14](#)

src\_redis(), [10](#)