

# Package ‘mongopipe’

January 20, 2021

**Title** Query MongoDB Documents with R

**Version** 0.1.1

**Maintainer** Oliver Haag <oliver\_haag@e.mail.de>

**Description** Translate R code into MongoDB aggregation pipelines.

**URL** <https://rpkgs.gitlab.io/mongopipe>,  
<https://gitlab.com/rpkgs/mongopipe>

**BugReports** <https://gitlab.com/rpkgs/mongopipe/-/issues>

**License** MIT + file LICENSE

**Suggests** testthat (>= 3.0.0), mongolite (>= 2.2.0), nycflights13

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** magrittr, jsonlite, rlang

**NeedsCompilation** no

**Author** Oliver Haag [aut, cre]

**Repository** CRAN

**Date/Publication** 2021-01-20 11:50:06 UTC

## R topics documented:

add_stage . . . . .	2
condition . . . . .	2
match . . . . .	3
mongopipe . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

add_stage	<i>Create or append a mongo aggregation stage</i>
-----------	---

---

### Description

Create or append a mongo aggregation stage

### Usage

```
add_stage(x, ...)
```

## Default S3 method:  
add\_stage(x, ...)

## S3 method for class 'mongopipe'  
add\_stage(x, type, ...)

### Arguments

x	Object of class mongo or mongo_pipe
...	query
type	Name of the stage.

### Value

Object of type mongopipe.

---

condition	<i>Conditional expression (\$cond)</i>
-----------	--

---

### Description

Evaluates a boolean expression to return one of the two specified return expressions.

### Usage

```
condition(test, yes, no)
```

### Arguments

test	Expression which returns a boolean value.
yes	Return this if the test returns true.
no	Return this if the test returns false.

**Value**

Return a list for using in mongopipe.

**Examples**

```
## Not run:
cond <- condition(test = list("$isArray"="$chart"),
                 yes = list("$size"="$chart"),
                 no = 0)
jsonlite::toJSON(cond)

## End(Not run)
```

---

 match

*Pipeline Stages*


---

**Description**

These functions translate R code to json readably by MongoDB.

**Usage**

```
match(x, ...)

## S3 method for class 'mongopipe'
match(x, ...)

field(x, ...)

## S3 method for class 'mongopipe'
field(x, ...)

lookup(x, ...)

## S3 method for class 'mongopipe'
lookup(x, from, local_field = from, foreign_field = "_id", as = from, ...)

unwind(x, ...)

## S3 method for class 'mongopipe'
unwind(x, field, ...)

limit(x, ...)

## S3 method for class 'mongopipe'
limit(x, limit, ...)
```

```
project(x, ...)

## S3 method for class 'mongopipe'
project(x, ...)
```

### Arguments

<code>x</code>	Object of class <code>mongopipe</code>
<code>...</code>	list object
<code>from</code>	Collection to join
<code>local_field</code>	Field from the input documents
<code>foreign_field</code>	Field from the documents of the "from" collection
<code>as</code>	Name of output array field (Default: <code>from</code> )
<code>field</code>	Field to unwind.
<code>limit</code>	Integer to limit the number of documents.

### Value

Object of type `mongopipe`.

---

`mongopipe`

*Mongopipe*

---

### Description

Mongopipe

### Usage

```
mongopipe()
```

### Value

Object of type `mongopipe`.

# Index

`add_stage`, 2

`condition`, 2

`field (match)`, 3

`limit (match)`, 3

`lookup (match)`, 3

`match`, 3

`mongopipe`, 4

`project (match)`, 3

`unwind (match)`, 3