

---

# **ReQON Documentation**

***Release 0.1.0***

**Derek Payton**

August 11, 2016



<b>1</b>	<b>Query descriptor</b>	<b>3</b>
1.1	\$db . . . . .	3
1.2	\$table . . . . .	3
1.3	\$query . . . . .	3
<b>2</b>	<b>Terms</b>	<b>5</b>
2.1	Selecting data . . . . .	5
2.2	Transformations . . . . .	5
2.3	Manipulation . . . . .	6
2.4	Aggregation . . . . .	7
2.5	Geospatial . . . . .	8
<b>3</b>	<b>Indices and tables</b>	<b>11</b>



ReQON ([ikn], /rikn/, RE-kon) lets you build simple, read-only ReQL queries from a JSON data structure.

Contents:



---

## Query descriptor

---

The ReQON query descriptor is an object with specific attributes:

```
{
  '$db': 'imdb',
  '$table': 'movies',
  '$query': [],
}
```

### 1.1 \$db

References a specific database.

ReQON	ReQL
{ '\$db': 'imdb' }	r.db('imdb')

This attribute is optional.

### 1.2 \$table

Return all of the documents in the specified table of the default database.

ReQON	ReQL
{ '\$table': 'movies' }	r.table('movies')
{ '\$db': 'imdb', '\$table': 'movies' }	r.db('imdb').table('movies')

This attribute is required.

### 1.3 \$query

The query attribute is a sequence of [terms](#) that filter, manipulate, or aggregate the document sequence in some way. Each term in the sequence is a list of 1 or 2 items, where the first item is the name of the term, followed (optionally, depending on the term) by a list of arguments.

This attribute is optional; omitting this attribute is the same as fetching every document in the specified `$table`.





## 2.1 Selecting data

### 2.1.1 \$get

**get** (*key*)

See also: <http://rethinkdb.com/api/python/get/>

### 2.1.2 \$get\_all

**get\_all** (*keys*[, *index* ])

See also: [http://rethinkdb.com/api/python/get\\_all/](http://rethinkdb.com/api/python/get_all/)

### 2.1.3 \$filter

**filter** (*predicate*)

Boolean operator	ReQL	See also:
\$and	<code>r.and_</code>	<a href="http://rethinkdb.com/api/python/and/">http://rethinkdb.com/api/python/and/</a>
\$or	<code>r.or_</code>	<a href="http://rethinkdb.com/api/python/or/">http://rethinkdb.com/api/python/or/</a>
\$not	<code>r.not_</code>	<a href="http://rethinkdb.com/api/python/not/">http://rethinkdb.com/api/python/not/</a>

See also: <http://rethinkdb.com/api/python/filter/>

### 2.1.4 \$between

**between** (*lower\_key*, *upper\_key*[, *index*, *left\_bound*, *right\_bound* ])

## 2.2 Transformations

### 2.2.1 \$has\_fields

**has\_fields** (*fields*)

See also: [http://rethinkdb.com/api/python/has\\_fields/](http://rethinkdb.com/api/python/has_fields/)

### 2.2.2 \$with\_fields

**with\_fields** (*fields*)

See also: [http://rethinkdb.com/api/python/with\\_fields/](http://rethinkdb.com/api/python/with_fields/)

### 2.2.3 \$order\_by

**order\_by** (*key* | *index*[, *ordering*])

See also: [http://rethinkdb.com/api/python/order\\_by/](http://rethinkdb.com/api/python/order_by/)

### 2.2.4 \$skip

**skip** (*n*)

See also: <http://rethinkdb.com/api/python/skip/>

### 2.2.5 \$limit

**limit** (*n*)

See also: <http://rethinkdb.com/api/python/limit/>

### 2.2.6 \$slice

**slice** (*start\_offset*[, *end\_offset*, *left\_bound*, *right\_bound*])

See also: <http://rethinkdb.com/api/python/slice/>

### 2.2.7 \$nth

**nth** (*n*)

See also: <http://rethinkdb.com/api/python/nth/>

### 2.2.8 \$sample

**sample** (*n*)

See also: <http://rethinkdb.com/api/python/sample/>

## 2.3 Manipulation

### 2.3.1 \$pluck

**pluck** (*fields*)

See also: <http://rethinkdb.com/api/python/pluck/>

### 2.3.2 \$without

**without** (*fields*)

See also: <http://rethinkdb.com/api/python/without/>

## 2.4 Aggregation

### 2.4.1 \$group

**group** (*field* | *index* [, *multi* ])

See also: <http://rethinkdb.com/api/python/group/>

### 2.4.2 \$count

**count** ()

See also: <http://rethinkdb.com/api/python/count/>

### 2.4.3 \$sum

**sum** (*field*)

See also: <http://rethinkdb.com/api/python/sum/>

### 2.4.4 \$avg

**avg** (*field*)

See also: <http://rethinkdb.com/api/python/avg/>

### 2.4.5 \$min

**min** (*field*)

See also: <http://rethinkdb.com/api/python/min/>

### 2.4.6 \$max

**max** (*field*)

See also: <http://rethinkdb.com/api/python/max/>

## 2.5 Geospatial

### 2.5.1 \$get\_intersecting

**get\_intersecting** (*geometry, index*)

See also: [http://rethinkdb.com/api/python/get\\_intersecting/](http://rethinkdb.com/api/python/get_intersecting/)

### 2.5.2 \$get\_nearest

**get\_nearest** (*geometry, index* [, *max\_results, max\_dist, unit, geo\_system* ])

See also: [http://rethinkdb.com/api/python/get\\_nearest/](http://rethinkdb.com/api/python/get_nearest/)

#### Selecting data

Term	Description
<i>\$get</i>	Get a single document by its primary key
<i>\$get_all</i>	Get all documents where the given value matches the requested index
<i>\$filter</i>	Get all the documents for which the specified sequence is true

#### Transformations

Term	Description
<i>\$with_fields</i>	Exclude documents that do not have the specified fields and return only those fields
<i>\$has_fields</i>	Test if a document has the specified fields, filtering out any that do not
<i>\$order_by</i>	Sort the documents by the specified field or index
<i>\$skip</i>	Skip a number of documents from the head of the sequence
<i>\$limit</i>	End the sequence after the given number of documents
<i>\$slice</i>	Return the documents within the specified range
<i>\$nth</i>	Get the <i>nth</i> document in the sequence
<i>\$sample</i>	Select a given number of elements from a sequence with uniform random distribution

#### Manipulation

Term	Description
<i>\$pluck</i>	Return only the specified fields
<i>\$without</i>	The opposite of <i>\$pluck</i> , return the documents without the specified fields

#### Aggregation

Term	Description
<i>\$group</i>	Partition the documents into multiple groups based on the specified field
<i>\$count</i>	Count the number of documents in the sequence
<i>\$sum</i>	Sum the specified field of the sequence
<i>\$avg</i>	Average the specified field of the sequence
<i>\$min</i>	Find the minimum value of the specified field in the sequence
<i>\$max</i>	Find the maximum value of the specified field of the sequence

**Geospatial**

Term	Description
<i>\$get_intersecting</i>	Get all documents where the given geometry object intersects with a geometry object of a geospatial index
<i>\$get_nearest</i>	Return the documents closest to the specified point based on a geospatial index



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`





## A

avg() (built-in function), 7

## B

between() (built-in function), 5

## C

count() (built-in function), 7

## F

filter() (built-in function), 5

## G

get() (built-in function), 5

get\_all() (built-in function), 5

get\_intersecting() (built-in function), 8

get\_nearest() (built-in function), 8

group() (built-in function), 7

## H

has\_fields() (built-in function), 5

## L

limit() (built-in function), 6

## M

max() (built-in function), 7

min() (built-in function), 7

## N

nth() (built-in function), 6

## O

order\_by() (built-in function), 6

## P

pluck() (built-in function), 6

## S

sample() (built-in function), 6

skip() (built-in function), 6

slice() (built-in function), 6

sum() (built-in function), 7

## W

with\_fields() (built-in function), 6

without() (built-in function), 7