

class MongoDB::Cursor

Cursor to iterate a set of documents

Table of Contents

- 1 [Synopsis](#)
- 2 [Readonly attributes](#)
 - 2.1 [client](#)
 - 2.2 [full-collection-name](#)
- 3 [Methods](#)
 - 3.1 [new](#)
 - 3.2 [iterator](#)
 - 3.3 [fetch](#)
 - 3.4 [kill](#)

```
unit package MongoDB;  
class Cursor does Iterable {...}
```

Synopsis

First example using find().

```
my MongoDB::Client $client .= new(:uri<mongodb://>);  
my MongoDB::Database $database = $client.database('contacts');  
my MongoDB::Collection $collection = $database.collection('perl_users');  
  
$d = $database.run-command: (count => $collection.name.);  
if $d<n> { say 'some docs available'; }  
  
# Get the documents  
my MongoDB::Cursor $cursor = $collection.find;  
while $cursor.fetch -> BSON::Document $document { $document.perl.say; }
```

Second example using run-command to get information about collections

```
$doc = $database.run-command: (listCollections => 1);  
is $doc<ok>, 1, 'list collections request ok';  
  
my MongoDB::Cursor $c .= new(:cursor-doc($doc<cursor>));  
while $c.fetch -> BSON::Document $d {  
    ...  
}
```

In both examples find()/new() and the following **while** can be combined with **for** because of the iterable role used on class Cursor.

```
...  
for $collection.find -> BSON::Document $document { $document.perl.say; }  
...
```

and

```
$doc = $database.run-command: (listCollections => 1);  
for MongoDB::Cursor.new(:cursor-doc($doc<cursor>)) -> BSON::Document $d {  
    ...  
}
```

After calling find (in the Collection module) to query the collection for data, a Cursor object is returned. With this cursor it is possible to iterate over the documents returned from the server. Cursor documents can be returned from specific calls to run-command. These documents must be converted to Cursor objects. See examples above.

Readonly attributes

client

```
has $.client;
```

Client object defined for this cursor.

full-collection-name

```
has $.full-collection-name;
```

This is the same as [~] `$database.name, '.', $collection.name`.

Methods

new

```
multi submethod BUILD (
  MongoDB::Collection:D :$collection!,
  BSON::Document:D :$server-reply
  :$server! where .^name eq 'MongoDB::Server'
)

multi submethod BUILD (
  MongoDB::Client:D :$client!,
  BSON::Document:D :$cursor-doc!
)

multi submethod BUILD (
  MongoDB::ClientType:D :$!client!, MongoDB::DatabaseType:D :$!database!,
  BSON::Document:D :$cursor-doc!, Bool:D :$modern!
)
```

Create a new Cursor object. the find() method in MongoDB::Collection will return a MongoDB::Cursor object. The first method is used by find() and second can be used by you using cursor information returned from a server.

Read-concern is by default taken from the client. When defined, it must be the same as used for the run-command returning the cursor information.

iterator

```
method iterator ( )
```

Method needed for the Iterable role placed on this Cursor class. **Not to be used directly!** Instead the following is possible;

```
for $collection.find -> BSON::Document $d { ... }
```

The method fetch() is indirectly used to get the next document.

fetch

```
method fetch ( --> BSON::Document )
```

Get the document from the query. The method returns a document or the undefined type object when no (more) documents are found. The documents are not saved in the Cursor object so after using fetch the action is not repeatable to get the same document again.

kill

```
method kill ( --> Nil )
```

Delete the cursor when it is no longer needed. By default the server will delete the cursor after a non-usage period of 10 minutes. See also find() flags described in Collection. The flag C-QF-NOCURSOR TIMEOUT for example will prevent the deletion of the cursor.