

```
> db.data.find().pretty()
{
  "_id" : ObjectId("61764e65c9963fa5565e7c58"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "brad pitt",
    "Diane Kruger",
    "Eki roth"
  ]
}
> db.data.insert({'title':'Fight club','writer':'Chuck Palahniuko','year':1999,'actors':["Brad Pitt","Edward Norton"]})
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:88
> db.data.insert({'title':'Fight club','writer':'Chuck Palahniuko','year':1999,'actors':["Brad Pitt","Edward Norton"]})
WriteResult({ "nInserted" : 1 })
> db.data.insert({'title':'Pulp fiction','writer':'Quentin Tarantino','year':1994,'actors':["John Travolta","Uma Thurman"]})
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:79
> db.data.insert({'title':'Pulp fiction','writer':'Quentin Tarantino','year':1994,'actors':["John Travolta","Uma Thurman"]})
WriteResult({ "nInserted" : 1 })
> db.data.find().pretty()
{
  "_id" : ObjectId("61764e65c9963fa5565e7c58"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "brad pitt",
    "Diane Kruger",
    "Eki roth"
  ]
}
{
  "_id" : ObjectId("61764f7ed59660479c07a5e7"),
  "title" : "Fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("61764feed59660479c07a5e8"),
  "title" : "Pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
>
```



```
"title" : "Inglorious Basterds",
"writer" : "Quentin Tarantino",
"year" : 2009,
"actors" : [
  "brad pitt",
  "Diane Kruger",
  "Eki roth"
]
}
{
  "_id" : ObjectId("61764f7ed59660479c07a5e7"),
  "title" : "Fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("61764feed59660479c07a5e8"),
  "title" : "Pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
> db.data.find('writer':"Quentin Tarantino").pretty()
uncaught exception: SyntaxError: missing ) after argument list :
@(shell):1:21
> db.data.find({'writer':"Quentin Tarantino"}).pretty()
{
  "_id" : ObjectId("61764e65c9963fa5565e7c58"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "brad pitt",
    "Diane Kruger",
    "Eki roth"
  ]
}
{
  "_id" : ObjectId("61764feed59660479c07a5e8"),
  "title" : "Pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
>
```

```
"title" : "Fight club",
"writer" : "Chuck Palahniuko",
"year" : 1999,
"actors" : [
  "Brad Pitt",
  "Edward Norton"
]
}
{
  "_id" : ObjectId("61764feed59660479c07a5e8"),
  "title" : "Pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
> db.data.find('writer':"Quentin Tarantino").pretty()
uncaught exception: SyntaxError: missing ) after argument list :
@(shell):1:21
> db.data.find({'writer':"Quentin Tarantino"}).pretty()
{
  "_id" : ObjectId("61764e65c9963fa5565e7c58"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "brad pitt",
    "Diane Kruger",
    "Eki roth"
  ]
}
{
  "_id" : ObjectId("61764feed59660479c07a5e8"),
  "title" : "Pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
> db.data.find({'actors':"Brad Pitt"}).pretty()
{
  "_id" : ObjectId("61764f7ed59660479c07a5e7"),
  "title" : "Fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
>
```



```
Windows PowerShell
"actors" : [
  "John Travolta",
  "Uma Thurman"
]
}
> db.data.find({'actors':"Brad Pitt"}).pretty()
{
  "_id" : ObjectId("61764f7ed59660479c07a5e7"),
  "title" : "Fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
> db.data.sort({year:1}).pretty()
uncaught exception: TypeError: db.data.sort is not a function :
@(shell):1:1
> db.data.sort({'year':1}).pretty()
uncaught exception: TypeError: db.data.sort is not a function :
@(shell):1:1
> db.data.find().sort({'year':1}).pretty()
{
  "_id" : ObjectId("61764feed59660479c07a5e8"),
  "title" : "Pulp fiction",
  "writer" : "Quentin Tarantino",
  "year" : 1994,
  "actors" : [
    "John Travolta",
    "Uma Thurman"
  ]
}
{
  "_id" : ObjectId("61764f7ed59660479c07a5e7"),
  "title" : "Fight club",
  "writer" : "Chuck Palahniuko",
  "year" : 1999,
  "actors" : [
    "Brad Pitt",
    "Edward Norton"
  ]
}
{
  "_id" : ObjectId("61764e65c9963fa5565e7c58"),
  "title" : "Inglorious Basterds",
  "writer" : "Quentin Tarantino",
  "year" : 2009,
  "actors" : [
    "brad pitt",
    "Diane Kruger",
    "Eki roth"
  ]
}
>
```

Local

> 7 DBS 7 COLLECTIONS

☆ FAVORITE

Filter your data

Movies

data

users

> Prasad

> admin

> comment

> config

> data

> local

Movies.users Documents

Movies.users

DOCUMENTS 6 TOTAL SIZE 666B AVG. SIZE 111B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

PROJECT { field: 0 }

SORT { field: -1 } or [['field', -1]]

COLLATION { locale: 'simple' }

OPTIONS

FIND RESET ↺ ⋮

MAX TIME MS 60000

SKIP 0 LIMIT 0

ADD DATA

VIEW

Displaying documents 1 - 6 of 6

REFRESH

users

	_id ObjectId	username String	title String	body String
1	617673bfaa4c415fd905a05a	"GoodGuyGreg"	"Passes out an party"	"Wakes up early and cleans hous"
2	617674b1aa4c415fd905a05e	"GoodGuyGreg"	"Steals your identity"	"Raises your credit score"
3	617674f3aa4c415fd905a05f	"GoodGuyGreg"	"Reports a bug in your code"	"Sends you a Pull request"
4	61767556aa4c415fd905a060	"ScrubagSteve"	"Borrows something"	"Sells it"
5	61767595aa4c415fd905a061	"ScrubagSteve"	"Borrows Everything"	"The end"
6	617675ebaa4c415fd905a062	"ScrubagSteve"	"Forks your repo on github"	"Sets to private"

>_MONGOSH

Local

> 7 DBS 7 COLLECTIONS

☆ FAVORITE

Filter your data

Movies

data

users

> Prasad

> admin

> comment

> config

> data

> local

+

> MONGOSH

Movies.users Documents

Movies.users

DOCUMENTS 6

TOTAL SIZE
823BAVG. SIZE
137B

INDEXES 1

TOTAL SIZE
36.0KBAVG. SIZE
36.0KB

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER [{"username": "GoodGuyGreg"}]

OPTIONS

FIND

RESET

↺

⋮

ADD DATA



VIEW



Displaying documents 1 - 3 of 3



REFRESH

```
{
  "_id": ObjectId("617673bfaa4c415fd905a05a"),
  "username": "GoodGuyGreg",
  "title": "Passes out an party",
  "body": "Wakes up early and cleans house",
  "post": "[post_obj-id]"
}
```

```
{
  "_id": ObjectId("617674b1aa4c415fd905a05e"),
  "username": "GoodGuyGreg",
  "title": "Steals your identity",
  "body": "Raises your credit score",
  "post": "[post_obj_id]"
}
```

```
{
  "_id": ObjectId("617674f3aa4c415fd905a05f"),
  "username": "GoodGuyGreg",
  "title": "Reports a bug in your code",
  "body": "Sends you a Pull request",
  "post": "[post_obj_id]"
}
```



Type here to search



32°C

15:01
25-10-2021

Local

> 7 DBS 7 COLLECTIONS

☆ FAVORITE

Filter your data

Movies

data

users

> Prasad
> admin
> comment
> config
> data
> local

+

> _MONGOSH

Movies.users Documents

Movies.users

DOCUMENTS 6 TOTAL SIZE 823B AVG. SIZE 137B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {"username": "ScrubagSteve"}

OPTIONS FIND RESET ↺ ⋮

ADD DATA

VIEW

Displaying documents 1 - 3 of 3

REFRESH

```
{ "_id": ObjectId("61767556aa4c415fd905a060"),  
  "username": "ScrubagSteve",  
  "title": "Borrows something",  
  "body": "Sells it",  
  "post": "[post_obj_id]" }
```

```
{ "_id": ObjectId("61767595aa4c415fd905a061"),  
  "username": "ScrubagSteve",  
  "title": "Borrows Everything",  
  "body": "The end",  
  "post": "[post_obj_id]" }
```

```
{ "_id": ObjectId("617675ebaa4c415fd905a062"),  
  "username": "ScrubagSteve",  
  "title": "Forks your repo on github",  
  "body": "Sets to private",  
  "post": "[post_obj_id]" }
```

Local

> 7 DBS 7 COLLECTIONS

☆ FAVORITE

Filter your data

Movies

data

users

> Prasad

> admin

> comment

> config

> data

> local

+

> MONGOSH

Movies.users Documents

Movies.users

DOCUMENTS 6 TOTAL SIZE 823B AVG. SIZE 137B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {author:GoodGuyGeng}

PROJECT {_id:0}

SORT { field: -1 } or [['field', -1]]

COLLATION { locale: 'simple' }

OPTIONS

MAX TIME MS 60000

SKIP 0

LIMIT 0

FIND

RESET

↺

⋮

VIEW

Displaying documents 1 - 6 of 6

↶

↷

REFRESH

```
username: "GoodGuyGreg"
title: "Passes out an party"
body: "Wakes up early and cleans house"
"post": "[post_obj-id]"
"comment": "Hope you got a good deal"
```

```
username: "GoodGuyGreg"
title: "Steals your identity"
body: "Raises your credit score"
"post": "[post_obj_id]"
"comment": "Whats mine is yours"
```

```
username: "GoodGuyGreg"
title: "Reports a bug in your code"
body: "Sends you a Pull request"
"post": "[post_obj_id]"
comment: "Dont violate the licensing aggrement"
```

```
username: "ScrubagSteve"
title: "Borrows something"
body: "Sells it"
"post": "[post_obj_id]"
```


Local

> 7 DBS 7 COLLECTIONS

☆ FAVORITE

Filter your data

Movies

data

users

> Prasad

> admin

> comment

> config

> data

> local

+

> _MONGOSH

Type here to search

Movies.users
Documents

Movies.users

DOCUMENTS	6	TOTAL SIZE	1.0KB	AVG. SIZE	172B
INDEXES	1	TOTAL SIZE	36.0KB	AVG. SIZE	36.0KB

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER {comment:"Reports a bug in your code"}

OPTIONS

FIND

RESET

↺

⋮

ADD DATA



VIEW



Displaying documents 1 - 1 of 1



REFRESH

```
_id: ObjectId("617675ebaa4c415fd905a062")
username: "ScrumbagSteve"
title: "Forks your repo on github"
body: "Sets to private"
"post": "[post_obj_id]"
comment: "Reports a bug in your code"
```