C:\Users\HP>mongosh "mongodb+srv://cluster0.6oikakw.mongodb.net/myFirstDatabase" --apiVersion 1 --username snehal281

Enter password: \*\*\*\*\*\*\*\*\*\*\*\*

Current Mongosh Log ID: 62b1ce968a264aa2ba0e9dd3

Connecting to: mongodb+srv://<credentials>@cluster0.6oikakw.mongodb.net/myFirstDatabase?appName=mongosh+1.5.0

Using MongoDB: 5.0.9 (API Version 1)

Using Mongosh: 1.5.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).

You can opt-out by running the disableTelemetry() command.

Atlas atlas-lzt1ql-shard-0 [primary] myFirstDatabase> mongo

ReferenceError: mongo is not defined

Atlas atlas-lzt1ql-shard-0 [primary] myFirstDatabase> mongod

ReferenceError: mongod is not defined

Atlas atlas-lzt1ql-shard-0 [primary] myFirstDatabase> show dbs

mongo\_practice 64.00 KiB

admin 336.00 KiB

local 5.41 GiB

**Connect to a running mongo instance, use a database named mongo\_practice. use mongo\_practice:-**

Atlas atlas-lzt1ql-shard-0 [primary] myFirstDatabase> use mongo\_practice

switched to db mongo\_practice

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> show collection

MongoshInvalidInputError: [COMMON-10001] 'collection' is not a valid argument for "show".

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> show collections

movies

**Insert the following documents into a movies collection.:-**

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.insert({'title':'Fight Club','writer':'Chuck Palahniuko','year':'1999','actors':['Brad Pitt','Edward Norton']})

{

acknowledged: true,

insertedIds: { '0': ObjectId("62b1dd08d4fa4b777a51d2ca") }

}

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.insertMany([{title:"Inglorious Basterds",writer:"Quentin Tarantino",year:2009,actors:["Brad Pitt","Diane Kruger","Eli Roth"]},{title:"The Hobbit:An Unexpected Journey",writer:"J.R.R. Tolkein",year:2012,franchise:"The Hobbit"}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("62b1f50bd4fa4b777a51d2cd"),

'1': ObjectId("62b1f50bd4fa4b777a51d2ce")

}

}

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.insertMany([{title:"The Hobbit: The Desolation of Smaug",writer:"J.R.R. Tolkein",year:2013,franchise:"The Hobbit"},{title:"The Hobbit: The Battle of the Five Armies",writer:"J.R.R. Tolkein",year:2012,synopsis:"Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness."},{title:"Pee Wee Herman's Big Adventure"},{title:"Avatar"}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("62b1f727d4fa4b777a51d2cf"),

'1': ObjectId("62b1f727d4fa4b777a51d2d0"),

'2': ObjectId("62b1f727d4fa4b777a51d2d1"),

'3': ObjectId("62b1f727d4fa4b777a51d2d2")

}

}

**query the movies collection to:-**

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find()

[

{

\_id: ObjectId("62b1dd08d4fa4b777a51d2ca"),

title: 'Fight Club',

writer: 'Chuck Palahniuko',

year: '1999',

actors: [ 'Brad Pitt', 'Edward Norton' ]

},

{

\_id: ObjectId("62b1eee4d4fa4b777a51d2cb"),

title: 'Pulp Fiction',

writer: 'Quentin Tarantino',

year: '1994',

actors: [ 'John Travolta', 'Uma Thurman' ]

},

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2cd"),

title: 'Inglorious Basterds',

writer: 'Quentin Tarantino',

year: 2009,

actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]

},

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2ce"),

title: 'The Hobbit:An Unexpected Journey',

writer: 'J.R.R. Tolkein',

year: 2012,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2cf"),

title: 'The Hobbit: The Desolation of Smaug',

writer: 'J.R.R. Tolkein',

year: 2013,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2d0"),

title: 'The Hobbit: The Battle of the Five Armies',

writer: 'J.R.R. Tolkein',

year: 2012,

synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2d1"),

title: "Pee Wee Herman's Big Adventure"

},

{ \_id: ObjectId("62b1f727d4fa4b777a51d2d2"), title: 'Avatar' }

]

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({writer:"Quentin Tarantino"})

[

{

\_id: ObjectId("62b1eee4d4fa4b777a51d2cb"),

title: 'Pulp Fiction',

writer: 'Quentin Tarantino',

year: '1994',

actors: [ 'John Travolta', 'Uma Thurman' ]

},

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2cd"),

title: 'Inglorious Basterds',

writer: 'Quentin Tarantino',

year: 2009,

actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]

}

]

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({actors:"Brad Pitt"})

[

{

\_id: ObjectId("62b1dd08d4fa4b777a51d2ca"),

title: 'Fight Club',

writer: 'Chuck Palahniuko',

year: '1999',

actors: [ 'Brad Pitt', 'Edward Norton' ]

},

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2cd"),

title: 'Inglorious Basterds',

writer: 'Quentin Tarantino',

year: 2009,

actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ]

}

]

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({"The Hobbit"})

Uncaught:

SyntaxError: Unexpected token (1:28)

> 1 | db.movies.find({"The Hobbit"})

| ^

2 |

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({franchise:"The Hobbit"})

[

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2ce"),

title: 'The Hobbit:An Unexpected Journey',

writer: 'J.R.R. Tolkein',

year: 2012,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2cf"),

title: 'The Hobbit: The Desolation of Smaug',

writer: 'J.R.R. Tolkein',

year: 2013,

franchise: 'The Hobbit'

}

]

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({$or:[{year:{$gte:2010}},{year:{$lt:2000}}]})

[

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2ce"),

title: 'The Hobbit:An Unexpected Journey',

writer: 'J.R.R. Tolkein',

year: 2012,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2cf"),

title: 'The Hobbit: The Desolation of Smaug',

writer: 'J.R.R. Tolkein',

year: 2013,

franchise: 'The Hobbit'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2d0"),

title: 'The Hobbit: The Battle of the Five Armies',

writer: 'J.R.R. Tolkein',

year: 2012,

synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'

}

]

**Update Documents:-**

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.update({"\_id": ObjectId("62b1f50bd4fa4b777a51d2ce")},{$set:{synopsis:"A reluctant hobbit,Bilbo Baggins,sets out to the Lonely Mountain witha spirited group of dwarves to reclaim their mountain home - and gold within it - from the dragon Smaug"}})

DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.update({"\_id": ObjectId("62b1f727d4fa4b777a51d2cf")},{$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey , continue their quest to recliam Erebor, their homeland, from Smaug, Bilbo Baggins is in possession of a mysterious and magical ring."}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.update({title:"Pulp Fiction"},{$set:{actors:"Samuel L. Jackson"}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

## Text Search:-

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({synopsis:{$regex:"Bilbo"}})

[

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2ce"),

title: 'The Hobbit:An Unexpected Journey',

writer: 'J.R.R. Tolkein',

year: 2012,

franchise: 'The Hobbit',

synopsis: 'A reluctant hobbit,Bilbo Baggins,sets out to the Lonely Mountain witha spirited group of dwarves to reclaim their mountain home - and gold within it - from the dragon Smaug'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2cf"),

title: 'The Hobbit: The Desolation of Smaug',

writer: 'J.R.R. Tolkein',

year: 2013,

franchise: 'The Hobbit',

synopsis: 'The dwarves, along with Bilbo Baggins and Gandalf the Grey , continue their quest to recliam Erebor, their homeland, from Smaug, Bilbo Baggins is in possession of a mysterious and magical ring.'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2d0"),

title: 'The Hobbit: The Battle of the Five Armies',

writer: 'J.R.R. Tolkein',

year: 2012,

synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'

}

]

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({synopsis:{$regex:"Gandalf"}})

[

{

\_id: ObjectId("62b1f727d4fa4b777a51d2cf"),

title: 'The Hobbit: The Desolation of Smaug',

writer: 'J.R.R. Tolkein',

year: 2013,

franchise: 'The Hobbit',

synopsis: 'The dwarves, along with Bilbo Baggins and Gandalf the Grey , continue their quest to recliam Erebor, their homeland, from Smaug, Bilbo Baggins is in possession of a mysterious and magical ring.'

}

]

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]})

[

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2ce"),

title: 'The Hobbit:An Unexpected Journey',

writer: 'J.R.R. Tolkein',

year: 2012,

franchise: 'The Hobbit',

synopsis: 'A reluctant hobbit,Bilbo Baggins,sets out to the Lonely Mountain witha spirited group of dwarves to reclaim their mountain home - and gold within it - from the dragon Smaug'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2d0"),

title: 'The Hobbit: The Battle of the Five Armies',

writer: 'J.R.R. Tolkein',

year: 2012,

synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.'

}

]

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({$or:[{synopsis:{$regex:"dwarves"}}, {synopsis:{$regex:"hobbit"}}]})

[

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2ce"),

title: 'The Hobbit:An Unexpected Journey',

writer: 'J.R.R. Tolkein',

year: 2012,

franchise: 'The Hobbit',

synopsis: 'A reluctant hobbit,Bilbo Baggins,sets out to the Lonely Mountain witha spirited group of dwarves to reclaim their mountain home - and gold within it - from the dragon Smaug'

},

{

\_id: ObjectId("62b1f727d4fa4b777a51d2cf"),

title: 'The Hobbit: The Desolation of Smaug',

writer: 'J.R.R. Tolkein',

year: 2013,

franchise: 'The Hobbit',

synopsis: 'The dwarves, along with Bilbo Baggins and Gandalf the Grey , continue their quest to recliam Erebor, their homeland, from Smaug, Bilbo Baggins is in possession of a mysterious and magical ring.'

}

]

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}}]})

[

{

\_id: ObjectId("62b1f50bd4fa4b777a51d2ce"),

title: 'The Hobbit:An Unexpected Journey',

writer: 'J.R.R. Tolkein',

year: 2012,

franchise: 'The Hobbit',

synopsis: 'A reluctant hobbit,Bilbo Baggins,sets out to the Lonely Mountain witha spirited group of dwarves to reclaim their mountain home - and gold within it - from the dragon Smaug'

}

]

## Delete Documents:-

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.remove({title:"Pee Wee Herman's Big Adventure"})

{ acknowledged: true, deletedCount: 1 }

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.movies.remove({title:"Avatar"})

{ acknowledged: true, deletedCount: 1 }

### **Insert the following documents into a users collection:-**

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> db.createCollection('users')

{ ok: 1 }

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> show collections

movies

users

Atlas atlas-lzt1ql-shard-0 [primary] mongo\_practice> use users

switched to db users

Atlas atlas-lzt1ql-shard-0 [primary] users> db.users.insert({\_id:1,username:"GoodGuyGreg", first\_name:"Good Guy", last\_name:"Greg"})

{ acknowledged: true, insertedIds: { '0': 1 } }

Atlas atlas-lzt1ql-shard-0 [primary] users> db.users.insert({\_id:2, username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})

{ acknowledged: true, insertedIds: { '0': 2 } }

### **Insert the following documents into a posts collection:-**

Atlas atlas-lzt1ql-shard-0 [primary] users> db.createCollection('posts')

{ ok: 1 }

Atlas atlas-lzt1ql-shard-0 [primary] users> show collections

posts

users

Atlas atlas-lzt1ql-shard-0 [primary] users> use posts

switched to db posts

Atlas atlas-lzt1ql-shard-0 [primary] posts> db.posts.insertMany([{username:"GoodGuyGreg", title:"Passes out at Party", body:"Wakes up early and cleans house"},{username:"GoodGuyGreg", title:"Steals your identity", body:"Raises your credit score"},{username:"GoodGuyGreg", title:"Reports a bug in your code", body:"Sends you a pull request"},{username:"ScumbagSteve", title:"Borrows something", body:"Sells it"},{username:"ScumbagSteve", title:"Borrows everything", body:"The end"},{username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private"}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("62b20951d4fa4b777a51d2d3"),

'1': ObjectId("62b20951d4fa4b777a51d2d4"),

'2': ObjectId("62b20951d4fa4b777a51d2d5"),

'3': ObjectId("62b20951d4fa4b777a51d2d6"),

'4': ObjectId("62b20951d4fa4b777a51d2d7"),

'5': ObjectId("62b20951d4fa4b777a51d2d8")

}

}

### **Insert the following documents into a comments collection:-**

Atlas atlas-lzt1ql-shard-0 [primary] posts> db.createCollection('comments')

{ ok: 1 }

Atlas atlas-lzt1ql-shard-0 [primary] posts> use comments

switched to db comments

Atlas atlas-lzt1ql-shard-0 [primary] comments> db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("62b20951d4fa4b777a51d2d6")})

{

acknowledged: true,

insertedIds: { '0': ObjectId("62b20a3fd4fa4b777a51d2d9") }

}

Atlas atlas-lzt1ql-shard-0 [primary] comments> db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("62b20951d4fa4b777a51d2d7")})

{

acknowledged: true,

insertedIds: { '0': ObjectId("62b20a72d4fa4b777a51d2da") }

}

Atlas atlas-lzt1ql-shard-0 [primary] comments> db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("62b20951d4fa4b777a51d2d8")})

{

acknowledged: true,

insertedIds: { '0': ObjectId("62b20aa2d4fa4b777a51d2db") }

}

Atlas atlas-lzt1ql-shard-0 [primary] comments> db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("62b20951d4fa4b777a51d2d3")})

{

acknowledged: true,

insertedIds: { '0': ObjectId("62b20ad5d4fa4b777a51d2dc") }

}

Atlas atlas-lzt1ql-shard-0 [primary] comments> db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("62b20951d4fa4b777a51d2d5")})

{

acknowledged: true,

insertedIds: { '0': ObjectId("62b20b1bd4fa4b777a51d2dd") }

}

## Querying related collections:-

Atlas atlas-lzt1ql-shard-0 [primary] comments> use users

switched to db users

Atlas atlas-lzt1ql-shard-0 [primary] users> db.users.find().pretty()

[

{

\_id: 1,

username: 'GoodGuyGreg',

first\_name: 'Good Guy',

last\_name: 'Greg'

},

{

\_id: 2,

username: 'ScumbagSteve',

fullname: { first: 'Scumbag', last: 'Steve' }

}

]

Atlas atlas-lzt1ql-shard-0 [primary] users> use posts

switched to db posts

Atlas atlas-lzt1ql-shard-0 [primary] posts> db.posts.find()

[

{

\_id: ObjectId("62b20951d4fa4b777a51d2d3"),

username: 'GoodGuyGreg',

title: 'Passes out at Party',

body: 'Wakes up early and cleans house'

},

{

\_id: ObjectId("62b20951d4fa4b777a51d2d4"),

username: 'GoodGuyGreg',

title: 'Steals your identity',

body: 'Raises your credit score'

},

{

\_id: ObjectId("62b20951d4fa4b777a51d2d5"),

username: 'GoodGuyGreg',

title: 'Reports a bug in your code',

body: 'Sends you a pull request'

},

{

\_id: ObjectId("62b20951d4fa4b777a51d2d6"),

username: 'ScumbagSteve',

title: 'Borrows something',

body: 'Sells it'

},

{

\_id: ObjectId("62b20951d4fa4b777a51d2d7"),

username: 'ScumbagSteve',

title: 'Borrows everything',

body: 'The end'

},

{

\_id: ObjectId("62b20951d4fa4b777a51d2d8"),

username: 'ScumbagSteve',

title: 'Forks your repo on github',

body: 'Sets to private'

}

]

Atlas atlas-lzt1ql-shard-0 [primary] posts> db.posts.find({username:"GoodGuyGreg"})

[

{

\_id: ObjectId("62b20951d4fa4b777a51d2d3"),

username: 'GoodGuyGreg',

title: 'Passes out at Party',

body: 'Wakes up early and cleans house'

},

{

\_id: ObjectId("62b20951d4fa4b777a51d2d4"),

username: 'GoodGuyGreg',

title: 'Steals your identity',

body: 'Raises your credit score'

},

{

\_id: ObjectId("62b20951d4fa4b777a51d2d5"),

username: 'GoodGuyGreg',

title: 'Reports a bug in your code',

body: 'Sends you a pull request'

}

]

Atlas atlas-lzt1ql-shard-0 [primary] posts> db.posts.find({username:"ScumbagSteve"})

[

{

\_id: ObjectId("62b20951d4fa4b777a51d2d6"),

username: 'ScumbagSteve',

title: 'Borrows something',

body: 'Sells it'

},

{

\_id: ObjectId("62b20951d4fa4b777a51d2d7"),

username: 'ScumbagSteve',

title: 'Borrows everything',

body: 'The end'

},

{

\_id: ObjectId("62b20951d4fa4b777a51d2d8"),

username: 'ScumbagSteve',

title: 'Forks your repo on github',

body: 'Sets to private'

}

]

Atlas atlas-lzt1ql-shard-0 [primary] posts> use comments

switched to db comments

Atlas atlas-lzt1ql-shard-0 [primary] comments> db.comments.find()

[

{

\_id: ObjectId("62b20a3fd4fa4b777a51d2d9"),

username: 'GoodGuyGreg',

comment: 'Hope you got a good deal!',

post: ObjectId("62b20951d4fa4b777a51d2d6")

},

{

\_id: ObjectId("62b20a72d4fa4b777a51d2da"),

username: 'GoodGuyGreg',

comment: "What's mine is yours!",

post: ObjectId("62b20951d4fa4b777a51d2d7")

},

{

\_id: ObjectId("62b20aa2d4fa4b777a51d2db"),

username: 'GoodGuyGreg',

comment: "Don't violate the licensing agreement!",

post: ObjectId("62b20951d4fa4b777a51d2d8")

},

{

\_id: ObjectId("62b20ad5d4fa4b777a51d2dc"),

username: 'ScumbagSteve',

comment: "It still isn't clean",

post: ObjectId("62b20951d4fa4b777a51d2d3")

},

{

\_id: ObjectId("62b20b1bd4fa4b777a51d2dd"),

username: 'ScumbagSteve',

comment: 'Denied your PR cause I found a hack',

post: ObjectId("62b20951d4fa4b777a51d2d5")

}

]

Atlas atlas-lzt1ql-shard-0 [primary] comments> db.comments.find({username:"GoodGuyGreg"})

[

{

\_id: ObjectId("62b20a3fd4fa4b777a51d2d9"),

username: 'GoodGuyGreg',

comment: 'Hope you got a good deal!',

post: ObjectId("62b20951d4fa4b777a51d2d6")

},

{

\_id: ObjectId("62b20a72d4fa4b777a51d2da"),

username: 'GoodGuyGreg',

comment: "What's mine is yours!",

post: ObjectId("62b20951d4fa4b777a51d2d7")

},

{

\_id: ObjectId("62b20aa2d4fa4b777a51d2db"),

username: 'GoodGuyGreg',

comment: "Don't violate the licensing agreement!",

post: ObjectId("62b20951d4fa4b777a51d2d8")

}

]

Atlas atlas-lzt1ql-shard-0 [primary] comments> db.comments.find({username:"ScumbagSteve"})

[

{

\_id: ObjectId("62b20ad5d4fa4b777a51d2dc"),

username: 'ScumbagSteve',

comment: "It still isn't clean",

post: ObjectId("62b20951d4fa4b777a51d2d3")

},

{

\_id: ObjectId("62b20b1bd4fa4b777a51d2dd"),

username: 'ScumbagSteve',

comment: 'Denied your PR cause I found a hack',

post: ObjectId("62b20951d4fa4b777a51d2d5")

}

]