

Instantly share code, notes, and snippets.

Create a gist now



durran / **embedded_update.rb**

Created 7 years ago

embedded_update.rb

```
1 # Problem: Given the following document in the database, I want to update
2 # Dave Gahan's second prescription for Dilaudid from a quantity of 45 to
3 # a quantity of 30.
4
5 { "_id" : 1
6   "name" : "Dave Gahan"
7   "medications" : [
8     { "id" : 23,
9       "name" : "Dilaudid",
10      "type" : "Rx",
11      "prescriptions" : [
12        { "id" : 13,
13          "quantity" : 60,
14          "started" : 2009-01-01
15        },
16        { "id" : 77,
17          "quantity" : 45,
18          "started" : 2009-02-01
19        }
20      ]
21    },
22    { "id" : 41,
23      "name" : "Oxycodone",
24      "type" : "Rx"
25    }
26  ]
27 }
28
29 # First try was using 1.3.4 $ positional locator... These do not find the
30 # document embedded multiple levels:
31
32 collection.update(
33   { "_id" : 1, "medications._id" : 23, "medications.prescriptions._id" : 77 },
34   { $set : { "medications.prescriptions.$.quantity" : 30 } },
35   false,
36   true
37 )
38
39 collection.update(
40   { "_id" : 1, "medications._id" : 23, "medications.prescriptions._id" : 77 },
41   { $set : { "medications.$.prescriptions.$.quantity" : 30 } },
42   false,
43   true
44 )
45
46 # What I would like is to be able to match on multiples, given a syntax like the following:
47
48 collection.update(
49   { "_id" : 1, "medications._id" : 23, "medications.prescriptions._id" : 77 },
50   { $set : { "medications.$0.prescriptions.$1.quantity" : 30 } },
51   false,
52   true
53 )
```



rdetert commented on Aug 14, 2010

Any luck with this?



Blurjp commented on Sep 27, 2012

not yet



tejasshah93 commented on Dec 6, 2013

Got any solution for this? I'm stuck at the same point..



ovinokurov commented on Feb 14, 2014

```
setModifier.$set['medications.$'] = JSON.parse({ "id" : 77, "quantity" : 45, "started" : 2009-02-01});

db.DBname.update({'_id': 1,"medications.id":77}, setModifier, function(err, updated) {
if( err || !updated ) console.log("Error: "+err);
else console.log("updated");
});
} catch (e) {
console.log("Error");
console.log(e);
}
```



thalesfsp commented on Mar 25, 2014

Its working?

```
setModifier.$set['medications.$'] = JSON.parse({ "id" : 77, "quantity" : 45, "started" : 2009-02-01});

db.DBname.update({'_id': 1,"medications.id":77}, setModifier, function(err, updated) {
if( err || !updated ) console.log("Error: "+err);
else console.log("updated");
});
} catch (e) {
console.log("Error");
console.log(e);
}
```



LoicSellaye commented on May 18, 2014

Hello everybody,

I am really interested about finding the solution of this problem because I face the same problem like you.

Best
Loic



xhh261672 commented on Sep 8, 2014

Stuck,too. Waiting for solutions
best wishes~



somethvictory commented on Oct 30, 2014

4 years till now, is there any luck with this?



thatisuday commented on Nov 5, 2014

It works with "medications.\$prescriptions.0.quantity" : 30,
which means mongodb is unable to find position of nested document...



kundansharma commented on Jan 3, 2015

Try this... I hope this solution will help you...

```
collection.aggregate({$match: {_id:1}},{$unwind:'$medications'},{$match: { 'medications.id' : 23 }},function(err,result1){
  if(err){
    throw(err);
  }else{
    indexes = result1[0].medications.prescriptions.map(function(obj, index) {
      if(obj.id == 13) {
        return index;
      }
    }).filter(isFinite);
    var updateData = {};
    updateData["medications.$prescriptions."+indexes[0]+"quantity"]=30;
    collection.update({_id:1,'medications.id': 23,'medications.prescriptions.id': 13 },{ $set: updateData },function(err,result2){
      console.log(result2);
    });
  }
});
```



whilda commented on Jan 10, 2015

There are any solution for this problem if i work with java?



shibindot commented on Apr 4

for the below aggregation query it results error

```
assert: command failed: {
  "ok" : 0,
  "errmsg" : "pipeline element 3 is not an object",
  "code" : 15942
} : aggregate failed

db.country_state_city.aggregate({$unwind:'$stateinfo'},{$unwind:'$stateinfo.cityinfo'},
{ $match: {$and:[{"_id":1},{ "stateinfo.state_id" : 1}]}},function(err,result1){
  if(err){
    throw(err);
  }else{
    indexes = result1[0].stateinfo.cityinfo.map(function(obj,index)
    {
      if(obj.city_id == 3) {
        return index;
      }
    }).filter(isFinite);
    var updateData = {};
    updateData["stateinfo.$cityinfo."+indexes[0]+"default"]=120;
    db.country_state_city.update({ "_id":1,"stateinfo.state_id": 1,"stateinfo.cityinfo.city_id": 3 },
```

```
{$set: updateData },function(err,result2){  
  console.log(result2);  
});  
}
```



stanisdev commented on May 14

This solution (`medications.$prescriptions.0.quantity`) was helpful, but arises next issue: to find index of array (in example is used "0"). As version of solution to find and retrieve index of necessary element in array as separate query and substitute inside "update query". Any way, thank you for advice.



jonathanmccormick commented on Jul 5

I don't know if there's a reason MongoDB doesn't support this, but boy would it make life so much easier...



dlee148 commented on Aug 2

It's 2016 and this is very much still an issue. I have created a simple workaround using the C# driver, hopefully it helps.
<https://gist.github.com/dlee148/a0ec830baff5c71471213616f22cb902>

[Sign up for free](#) to join this conversation on GitHub. Already have an account? [Sign in to comment](#)