International Journal of Advanced Science and Research
ISSN: 2455-4227
Impact Factor: RJIF 5.12
www.allsciencejournal.com
Volume 3; Issue 1; January 2018; Page No. 112-114

A survey on update a document's array in MongoDB

Deep Kumar
Software Engineer, Igniva Solutions Private Limited, Mohali, Punjab, India

Abstract
MongoDB is NoSQL data base used to store a high volume document. In this you will learn different array update operations. The following modifiers are used to update document' array. mongoDb is open sourced data based used for document. You can store up to 16GB data in a document. It is written in C++. You should have basic knowledge of database, editor and executions of programs. And as least you should know about RDMS concept as you are going to study high level database. Database in document is known as collection. which is equal o a relation in RDBMS. Unlike RDBMS it's has no predefined schema. The schema in mongodb is dynamic. You can insert different key set in another document as well. Here is _id key which is by default key and work as a primary key in document. As schema is dynamic but you should save same type of document in a collection.

Keywords: MongoDb, array operations, array modifiers, nested array, collection, document operations

Introduction
Schema in mongodb is dynamic. So you can insert any kind of data in document. When you insert an array or array of object then it should be easy to maintainable. So mongodb provide a lot of operations for easy to use an array and array of objects. Like push, pop, pull all, position. You can easily insert and remove values from array based on their positions or based on their value condition. These are following explained in detail with proper example and syntax. You are getting knowledge about update operations on array in this. So you should have knowledge about update operations. To update any document, you should use update(condition, data). Update take 2 arguments, first when is condition and second is data which need to update in document. If you want to update at specific position then you should use the condition to access that document for update, for all send empty object to apply changes on all the document of collection.

Array Update Operators List
- $push: insert value in an array.
- $position: insert value at given position.
- $pop: delete value from first or last position in array.
- $pull: removes array values that match a specific condition.
- $: inert value at specific condition.
- $slice: to set the limit of array element.
- $sort: to sort the data in ascending or descending order.

Suppose below is teacher collection which contain 2 documents. Each teacher teaches few subjects and also described cost of subject. And teachAt contain the location where he/ she teach that subjects. This collection is used in below all examples.{_id: "teach101",teacher Name: "Rakesh", teach: [{name: "NODE", cost: 30, description: "Node with mongo db.", },{name: "REACT", cost: 50, description: "react with html.", }], teachAt:["PKL", "CHD", "ABS"]}

$push
This modifier is used to adds an item to an array. if you try to use for non-array element then mongoDbthorws an exception. Itwill add the new element in document of array type with given element if element is not exist in document Syntax:- db.<collection>.update({<Condition>},{$push:{<field>:<value>, ...}}); Exp:- Query when teacher start to teach at UK. db.teacher.update({_id: "teach101"},{$push: { teachAt: "UK" } });

Case 1
If you try to use for non-array element then mongoDbthorws an exception.xp:-db.teacher.update({_id: "teach101"},{$push: { tecaherName: 2 } });You can't apply this operation on non-array element. Because the tecaherName is string type object key.

Case 2
It will add the new element in document of array type with given element if element is not exist in document.Exp:- Query when teacher add his/her qualifications.b.teacher.update({_id: "teach101"},{$push: { qualifications: "MCA" } });In above example qualifications is not a element in document. So mongo push operation will add the qualifications element in document which has array type. And store MCA as first element in qualifications array.

$pop
this is used to delete the last or first element of an array. you need to pass 1 or -1 values. 1 to remove from last and -1 to remove first element. Syntax:- db.<collection>.update({<Condition>},{$pop:<field >: <1 | -1>, ...}}); Exp. Query when teacher leave to teach at last position which is ABS.db.teacher.update({_id:"teach101"},{$pop: { teachAt: 1} });Samewe can pass -1 as argument value. to remove first

element from array. if you try to use for non-array element then mongoDbthorws an exception.

$pull

As above we see how pop is working, to remove a specific value pull operation is required. It used to remove the instance of array that matches the specific condition. when we want to remove irrespective of index value of array.Syntax:- db.<collection>.update({<Condition>},{$pull:{<field>:<value|condition>, ... }});Exp:- Query when teacher leave to teach at specifc location like CHD.db.teacher.update({_id: "teach101"},{$pull: {teachAt: {$in: ["CHD"]}} }, {multi: true });when you want to delete a element based on the value's condition like greater than, less than, equal to etc then you need to use $elemMatch modifier. For example remove all teaching subject whose cost is less than give value. For such kind of values's condition you need to use $elemMatch modifier.Exp:- remove teaching subject whose cost is less than 40.db.teacher.update({_id: "teach101"},{$pull: {teach : $elemMatch{cost : {$lt : 40} }} }, {multi: true }); In above query teach object will removed which's cost is less than 40. so node will be deleted because its cost is 30 which is less than 40.

$position

As above you see that push is used to insert value of array at end position. if you have to insert a value at specific index then we need to use the position operation in query.In new versions of mongo you can also pass the negative values in position.Syntax:-
db.<collection>.update({<Condition>},{$push: {<field>: { $each: [<value>, ...], $position: <number> } }});Exp:-Query when teacher start to teach at UK, LDH and want to insert at 2 position.db.teacher.update({_id:"teach101"},{$push:{teachAt: :{$each: ["UK", "LDH"],$position: 1} } });In above teachAt is updated and the position of uk and LDH is at 2 index. And $each is used to insert all the array values in element.

Case 1

In new versions of mongo you can also pass the negative values in position.Which is work from last position of array as see in next example.Exp:- Query when teacher start to teach at UK, LDH and want to insert at second last position.db.teacher.update({_id:"teach101"},{$push:{teachAt: :{$each: ["UK", "LDH"], $position: -2} } }); In above teachAt is updated and the position of uk and LDH is at second last index. Negative number work as last in position element.

$

As we see above push and position opreation which are work on index. when you have no such index value to insert value in element then you can you $ opreation for insertion a vaule at specific match. as shown in below example.Syntax:- db.<collection>.update({<Condition>},{"<arr>.$": value}); Exp:-if teacher want to updated cost of NODE subject. db.teacher.update({_id:"teach101","teach.name":"NODE"},{$ set: {"teach.$.cost" : 100}}); As in above example you have no idea of index of array which contain NODE. So you can directly update the cost of array which contain NODE as subject name using $ operation.

$sort

when you want to insert data in sorted order then you can easily do it using sort operation. sort take field name and use value 1 and -1. 1 is used to sort that array in ascending order based on that field, and -1 is used for descending order.
$slice to specific the limit of push operation value. Like you want to insert only 10 element in an array so you can use limit in push operation. And in slice you have to use each modifier also. You can pass any integer in slice. Here 0 means to set the array element as empty array. And positive number means you want to keep n element from starting. And for negative number means you want to keep n elements from last of array
Syntax db.<collection>.update({<Condition>}, { $push: {<field>: {$each: [<value>,.....], $slice: <number>}}}); you can pass any integer in slice. Here 0 means to set the array element as empty array. andpositive number means you want to keep n element from starting. and for negative number means you want to keep n elements from last of array.

$sort

when you want to insert data in sorted order then you can easily do it using sort operation. Sort take field name and use value 1 and -1. 1 is used to sort that array in ascending order based on that field, and -1 is used for descending order.
Syntax db.<collection>.update({<Condition>},{$push:{ <field>: {$each: [<value>,.....], $sort: {<field>: 1|-1, ... }}})

Review of Literacture

Security of mongoDb is very low and the complexity of these operations are very high. And the different versions are not properly differentiated in any website or book, as every new version has new document so it's very hard to understand the difference between these versions. So in this you have got proper difference between these operations getting in these versions. And not proper test cases are not explained. And explained at high level which is difficult to understand for beginners and in this you got comparable with sql queries so these are easy to explain and understand for beginners. mongoDb is going to updated each and every month so you need to update on each and every month. So each and every old sites does not contain new concepts and test cases and changes in old operations. Few of them are deprecated. So you need to update on daily bases

Approaches Used

As everyone is familiar with RDBMS and during explaining update operations of array similar sql queries are explained. So it should be easy to use. To describe mongoDb, teacher collection is used, because everyone knows the basic activity of teacher. So taking the example of teacher is easy to understandable. All the modifiers are explained with syntax, example and their result each and every. And few cases of modifiers are also taken.

Conclusion

MongoDB is NoSQL data base used to store a high volume document. And here is too much easy to work with array, nested arrays because mongoDb gives a lot of modifiers, operators. And it's easy to learn and understand. In this you had learnt different array update operations with their syntax

and also see some special cases, properly explained with version of mongoDb. As every version extends the functionality of operations. You are getting knowledge upto 3.6th version of mongoDb. For more detail you can use below references.

References

1. https://docs.mongodb.com
2. http://tgrall.github.io/blog/2015/04/21/mongodb-playing-with-arrays/
3. https://www.tutorialspoint.com/mongodb/
4. https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.json.doc/ids_json_047.htm
5. https://www.guru99.com/mongodb-tutorials.html
6. https://beginnersbook.com/2017/09/mongodb-tutorial/
7. https://www.javatpoint.com/mongodb-tutorial
8. https://www.youtube.com/watch?v=1uFY60CESlM