Updating Documents in MongoDB



Douglas StarnesAUTHOR / SPEAKER

@poweredbyaltnet douglasstarnes.com



Updating Documents



API methods

- updateOne
- updateMany
- update
- replaceOne

Query operators return

Update operators

- Increment
- Rename or remove fields
- Arrays

Return document

- matchedCount
- modifiedCount



Everything about write concerns, collation documents, and atomicity is still valid



Replacing a Document

```
{
    "_id": "123xyz",
    "color": "red"
}
```

deleteOne()

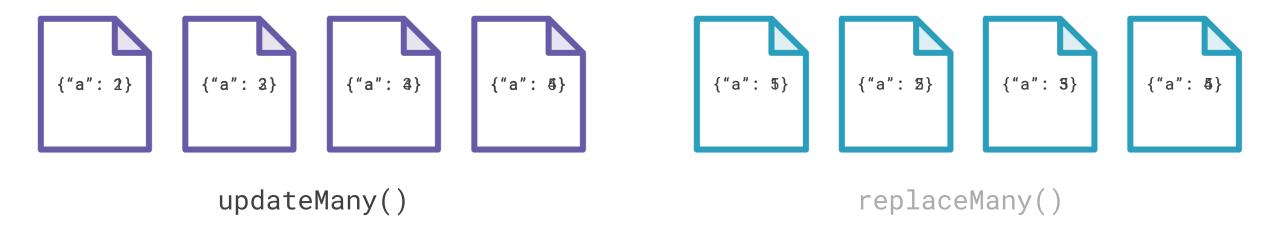
```
{
    "_id": "789xyz",
    "size": "tiny"
}
```

createOne()

replaceOne()



Why No replaceMany?



```
> db.sessions.updateMany(
  {"topic": {$eq: "javascript"}},
  {$set: {"room": "Barracuda"}},
  { /* write concern, collation */ }
  "acknowledged": "true",
  "matchedCount": 2,
  "modifiedCount": 2
```

```
"title": "JavaScript 101",
"topic": "javascript",
"room": "Bbalanada"
"title": "MongoDB 101",
"topic": "mongodb",
"room": "Abalone"
"title": "JavaScript 102",
"topic": "javascript",
"room": "Bbalaneda"
```



arrayFilters

```
db.sessions.updateMany(
  {"topic": {$eq: "javascript"}},
  {$set: {"ratings.$[rating]": "4"}},
    /* write concern, collation */
    "arrayFilters": [
      {"rating": {$1t: 4}}
})
```

```
> db.sessions.updateMany(
  {"topic": {$eq: "javascript"}},
  {$set: {"room": "Barracuda"}},
  { /* write concern, collation */ }
  "acknowledged": "true",
  "matchedCount": 2,
  "modifiedCount": 1
```

```
"title": "JavaScript 101",
"topic": "javascript",
"room": "Bbalanada"
"title": "MongoDB 101",
"topic": "mongodb",
"room": "Abalone"
"title": "JavaScript 102",
"topic": "javascript",
"room": "Barracuda"
```



```
> db.sessions.replaceOne(
  {"topic": {$eq: "javascript"}},
    "title": "Virtual Python",
    "topic": "python"
  },
  { /* write concern, collation */ }
})
  "acknowledged": "true",
  "matchedCount": 2,
  "modifiedCount": 1
```

```
"title": "Yava8aliPyth0h",
  "topic": "pythonfipt",
}, "room": "Abalone"
{ "title": "MongoDB 101",
  "tople": "MongodB",101",
  "topmc": "Abangaeb",
}, "room": "Abalone"
{ "title": "JavaScript 102",
  "tople": "java@cript",102",
  "topmč": "Baavasudapt",
} "room": "Barracuda"
```



INPUT

Field Update Operators



\$set

```
db.conferences.updateOne(
    {"_id": "mongoconf"},
    {$set: {"ticket_cost": 400}}
)
```

```
"_id": "mongoconf",
"title": "MongoDB Conf"
"ticket_cost": 200
"_id": "jsconf",
"title": "JavaScript Conf"
"ticket_cost": 150
```

\$inc

```
db.conferences.updateOne(
    {"_id": "mongoconf"},
    {$inc: {"attendees": 1}}
)

db.conferences.updateOne(
    {"_id": "jsconf"},
    {$inc: {"attendees": -25}}
)
```

```
"_id": "mongoconf",
"title": "MongoDB Conf"
"attendees": 250
"_id": "jsconf",
"title": "JavaScript Conf"
"attendees": 120
```

\$mul

```
db.conferences.updateOne(
  {"_id": "mongoconf"},
  {$mul:
    {"ticket_cost": NumberDecimal("1.5")}
db.conferences.updateOne(
  {"_id": "mongoconf"},
  {$mul:
    {"tcket_cost": NumberDecimal("1.5")}
```

```
"_id": "mongoconf",
 "title": "MongoDB Conf"
  "ticket_cost": 330,
}, "tcket_cost": 0
{ "_id": "jsconf",
 "tidle":jsdemascript Conf"
  "ticket:costvaSc5@pt Conf"
 "ticket_cost": 150
```

\$unset

```
db.conferences.updateOne(
    {"_id": "mongoconf"},
    {$unset: {"tcket_cost": "")}
    }
)
```

```
"_id": "mongoconf",
 "title": "MongoDB Conf"
  "ticket_cost": 375,
}, "tcket_cost": 0
{ "_id": "jsconf",
 "tidle" "jsdemสริcript Conf"
  "ticket:costvaSc5@pt Conf"
 "ticket_cost": 150
```



Array Update Operators



\$pop

```
db.conferences.updateOne(
    {"_id": "mongo101"},
    {$pop: {"topics": -1}}
)
```

```
{"_id": "mongo101"},
   {$pop: {"ratings": 1}}
)
```

db.conferences.updateOne(

-1 removes first element

1 removes last element

```
{
    "_id": "mongo101",
    "title": "MongoDB 101"
    "topics": ["mengedbipt"dataongodb", "data"],
    "ratings": [6, 8, 5, 10] 7]
}
```

\$pull

```
db.conferences.updateOne(
    {"_id": "mongo101"},
    {$pull: {"ratings": {$lte: 3}}}
)
```

Remove all ratings less than or equal to 3

```
{
    "_id": "mongo101",
    "title": "MongoDB 101"
    "topics": ["javascript", "mongodb", "data"],
    "ratings": [6, 6, 5, 80,37]5, 10, 2, 7]
}
```

\$push

```
db.conferences.updateOne(
    {"_id": "mongo101"},
    {$push: {"ratings": 8}}
)
```

Add the value 8 to the end of ratings

```
{
    "_id": "mongo101",
    "title": "MongoDB 101"
    "topics": ["javascript", "mongodb", "data"],
    "ratings": [6, 8, 5, 10, 7] 8]
}
```

\$push with the \$each modifier

```
db.conferences.updateOne(
    {"_id": "mongo101"},
    {$push: {"ratings": {$each :[8, 7, 10]}}}
)
```

Add all values to the end of ratings

```
{
    "_id": "mongo101",
    "title": "MongoDB 101"
    "topics": ["javascript", "mongodb", "data"],
    "ratings": [6, 8, 5, 10, 7] 8, 7, 10]
}
```

\$push with the \$each and \$position modifiers

Add all values to the start of ratings

```
{
    "_id": "mongo101",
    "title": "MongoDB 101"
    "topics": ["javascript", "mongodb", "data"],
    "ratings": [8, 8, $0, 106,7$, 5, 10, 7]
}
```

Filtered Positional Operator

```
db.sessions.updateOne(
  {"_id": "mongoconf"},
    $inc: {"ratings.$[poorratings]": 1}
    "arrayFilters": [
      {"poorratings": {$lte: 3}}
```



Demo



Updating documents in MongoDB



Summary



API methods

- updateOne
- updateMany
- replaceOne

Update vs. replace

Update operators

- Field update operators
- Array update operators

Return document

- matchedCount
- modifiedCount
- Not always the same

