

Updating and Replacing Documents



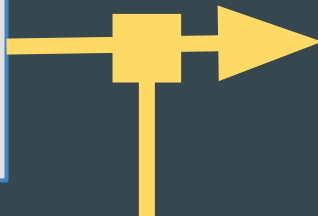
By Femi Obiwumi

Replacing Documents

The command **replaceOne** fully replaces a matching document with another.

It takes two parameters, the document to replace, and the document it's being replaced with.

```
{
  "_id" : ObjectId(4...a)
  "name" : "joe"
  "friends" : 32
  "enemies" : 2
}
```



```
{
  "_id" : ObjectId(4...a)
  "username" : "joe"
  "relationships" : {
    "friends" : 32
    "enemies" : 2
  }
}
```

```
> var joe = db.users.findOne({"name" : "joe"});
> joe.relationships = {"friends" : joe.friends, "enemies" :
joe.enemies};
> joe.username = joe.name;
> delete joe.friends;
> delete joe.enemies;
> delete joe.name;
> db.users.replaceOne({"name" : "joe"}, joe);
```

Updating Documents

It is often easier to update an existing document rather than replace it with a new one. The command **updateOne** can be used to do so. It takes two parameters: the document to be updated and an update modifier.

Types of modifiers include:

- **\$inc** - increments or decrements a number
- **\$set** - sets a value of an existing field or creates a new field and sets it to the specified value
- **\$unset** - removes a field

\$inc

```
{
  "_id" : ObjectId(4...1)
  "url" : "www.example.com"
  "pageviews" : 52
}
```



```
{
  "_id" : ObjectId(4...1)
  "url" : "www.example.com"
  "pageviews" : 53
}
```

```
> db.site.updateOne({"url" : "www.example.com"}, {"$inc" : {"pageviews" : 1}}) ... ;
```

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 500
}
```



```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 10500
}
```

```
> db.game.updateOne({"game" : "pinball"}, {"$inc" : {"score" : 10000}}) ... ;
```

* The increment value can be negative or positive

\$set and \$unset

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 10500
}
```



```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 0
}
```

```
> db.game.updateOne({"game" : "pinball"}, {"$set" : {"score" : 0}}) ... ;
```

```
{
  "_id" : ObjectId(4...g)
  "url" : "www.example.com"
  "pageviews" : 52
  "site title" : "Example Site"
}
```



```
{
  "_id" : ObjectId(4...g)
  "url" : "www.example.com"
  "pageviews" : 53
}
```

```
> db.site.updateOne({"game" : "pinball"}, {"$unset" : {"site title" : 1}}) ... ;
```



This value can be anything, it doesn't matter.

More Modifiers: Arrays

- **\$push** - adds an element to the end of an array field or creates a new array field with the specified elements
 - **\$each** - allows you to add multiple elements at once to the array
 - **\$slice** - prevents the array from growing past a certain size (must be used with the each modifier)
 - **\$sort** - sorts the array
 - **\$ne** - selects the documents where the field is not equal to a specified value
- **\$addToSet** - adds an element to the array, but treats it as a set, only adding elements that are not already in the array.
- **\$pop** - removes an element from the beginning or end of the array.
- **\$pull** - removes an element based on the value (removes all elements matching the specified value).

\$push and \$each

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 10500
}
```



```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 10500
  "playernames" : ["joe"]
}
```

```
> db.game.updateOne({"game" : "pinball"}, {"$push" : {"playernames" : "joe"}}) ... ;
```

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 10500
  "playernames" : ["joe"]
}
```



```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 10500
  "playernames" : ["joe", "bob", "sam", "tia"]
}
```

```
> db.game.updateOne({"game" : "pinball"},
  {"$push" : {"playernames" : {"$each" : ["bob", "sam", "tia"]}}}) ... ;
```

\$slice and \$sort

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "highscores" : [33500, 95000, 20000]
}
```



```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "highscores" : [20000, 60500]
}
```

```
> db.game.updateOne({"game" : "pinball"},
  {"$push" : {"highscores" : {"$each" : [60500], "$slice" : -2}}} ... ;
```

The number of elements to leave in the list; The sign determines which side to be trimmed.

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "highscores" : [33500, 95000, 20000]
}
```



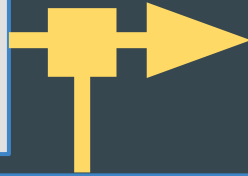
```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "highscores" : [95000, 60500]
}
```

```
> db.game.updateOne({"game" : "pinball"},
  {"$push" : {"highscores" : {"$each" : [60500], "$slice" : -2,
    "$sort" : -1}}} ... ;
```

Sort order. 1 is ascending, -1 is descending

\$ne and \$addToSet

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "playernames" : ["joe", "bob", "sam", "tia"]
}
```



```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "playernames" : ["joe", "bob", "sam", "tia"]
}
```

```
> db.game.updateOne({"playernames" : {"$ne" : "joe"}},
  {"$push" : {"playernames" : "joe"}}) ... ;
```

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "playernames" : ["joe", "bob", "sam", "tia"]
}
```

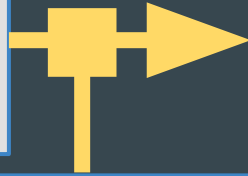


```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "playernames" : ["joe", "bob", "sam", "tia"]
}
```

```
> db.game.updateOne({"game" : "pinball"},
  {"$addToSet" : {"playernames" : "joe"}}) ... ;
```

\$pop and \$pull

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "playernames" : ["joe", "bob", "sam", "tia"]
}
```

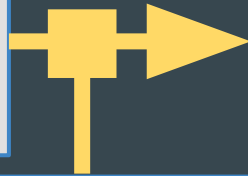


```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "playernames" : ["bob", "sam", "tia"]
}
```

```
> db.game.updateOne({"game" : "pinball"}, {"$pop" : {"playernames" : -1}}) ... ;
```

Determines which end to pop from: 1 pops from the end, -1 pops from the beginning

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "playernames" : ["joe", "bob", "sam", "tia"]
}
```



```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "score" : 60500
  "playernames" : ["joe", "bob", "tia"]
}
```

```
> db.game.updateOne({"game" : "pinball"}, {"$pull" : {"playernames" : "sam"}}) ... ;
```

Updates Using Array Filters

The **arrayFilters** command can be used to only modify elements matching particular criteria.

```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "players" : [
    {"name" : "joe"
     "score" : 44500},
    {"name" : "bob"
     "score" : 96000},
    {"name" : "tia"
     "score" : 101500}
  ]
}
```



```
{
  "_id" : ObjectId(4...2)
  "game" : "pinball"
  "players" : [
    {"name" : "joe"
     "score" : 44500},
    {"name" : "bob"
     "score" : 96000},
    {"name" : "tia"
     "score" : 101500
     "prizewon" : true}
  ]
}
```

```
> db.game.updateOne({"game" : "pinball"},
  {"$set": {"players.$[elem].prizewon" : true}},
  {arrayFilters : [{"elem.score" : {$gte : 100000}]}})
```

Sources

<https://docs.mongodb.com/manual/reference/method/db.collection.replaceOne/>

<https://docs.mongodb.com/manual/reference/operator/update/>

<https://database.guide/how-the-arrayfilters-parameter-works-in-mongodb/>