

MongoDB Update with upsert = true to Avoid Inserting Duplicates

MongoDB does not have “Transaction”. It only guarantees a document level atomicity. So no concurrency control algorithms. One way to avoid concurrency problems is through update with upsert true on an unique index

Use Unique Indexes

Warning:

To avoid inserting the same document more than once, only use upsert: true if the query field is uniquely indexed.

Given a collection named people where no documents have a name field that holds the value Andy. Consider when multiple clients issue the following update with upsert: true at the same time:

```
db.people.update(  
  { name: "Andy" },  
  {  
    name: "Andy",  
    rating: 1,  
    score: 1  
  },  
  { upsert: true }  
)
```

If all update() operations complete the query portion before any client successfully inserts data, and there is no unique index on the name field, then each update operation may result in an insert.

To prevent MongoDB from inserting the same document more than once, create a unique index on the name field. With a unique index, if multiple applications issue the same update with upsert: true, exactly one update() would successfully insert a new document.

The remaining operations would either:

- update the newly inserted document, or
- fail when they attempted to insert a duplicate.

If the operation fails because of a duplicate index key error, applications may retry the operation which will succeed as an update operation.

upsert:true with a Dotted _id Query

When you execute an `update()` with `upsert: true` and the query matches no existing document, MongoDB will refuse to insert a new document if the query specifies conditions on the `_id` field using dot notation.

This restriction ensures that the order of fields embedded in the `_id` document is well-defined and not bound to the order specified in the query

If you attempt to insert a document in this way, MongoDB will raise an error.

For example, consider the following update operation. Since the update operation specifies `upsert:true` and the query specifies conditions on the `_id` field using dot notation, then the update will result in an error when constructing the document to insert.

```
db.collection.update( { "_id.name": "Robert Frost", "_id.uid": 0 },  
  { "categories": ["poet", "playwright"] },  
  { upsert: true } )
```