



MongoDB insertOne

Summary: in this tutorial, you will learn how to use the MongoDB `insertOne()` method to insert a single document into a collection.

Introduction to MongoDB `insertOne()` method

The `insertOne()` method allows you to insert a single document into a collection.

The `insertOne()` method has the following syntax:

```
db.collection.insertOne(  
    <document>,  
    { writeConcern: <document>}  
)
```

The `insertOne()` method accepts two arguments:

- `document` is a document that you want to insert into the `collection`. The `document` argument is required.
- `writeConcern` is an optional argument that describes the level of acknowledgment requested from MongoDB for insert operation to a standalone MongoDB server or to shared clusters. We'll discuss the `writeConcern` another tutorial.

The `insertOne()` method returns a document that contains the following fields:

- `acknowledged` is a boolean value. It is set to true if the insert executed with write concern or false if the write concern was disabled.
- `insertedId` stores the value of `_id` field of the inserted document.

Note that if the `collection` does not exist, the `insertOne()` method will also create the collection and insert the `document` into it.

If you don't specify the `_id` field in the document, MongoDB will add the `_id` field and generate a unique `ObjectId` for it before insert.

If you explicitly specify a value for the `_id` field, you need to ensure that it is unique in the collection. Otherwise, you will get a duplicate key error.

To insert multiple documents into a collection, you use the `insertMany()` (<https://mongodbtutorial.org/mongodb-crud/mongodb-insertmany/>) method.

MongoDB insertOne() method examples

First, you need to launch the mongo shell and connect it to the `bookdb` database:

```
mongosh bookdb
```

1) Insert a document without an `_id` field example

The following example uses the `insertOne()` method to insert a new document into the `books` collection:

```
db.books.insertOne({
  title: 'MongoDB insertOne',
  isbn: '0-7617-6154-3'
});
```

Output:

```
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5f31cf00902f22de3464ddc4")
}
```

In this example, we passed a document to the `insertOne()` method without specifying the `_id` field. Therefore, MongoDB automatically added the `_id` field and assigned it a unique `ObjectId` value.

Note that you will see a different `ObjectId` value from this example because `ObjectId` values are specific to machine and time when the `insertOne()` method executes.

To select the document that you have inserted, you can use the `find()` (<https://mongodbtutorial1.org/mongodb-crud/mongodb-find/>) method like this:

```
db.books.find()
```

Output:

```
[
  {
    _id: ObjectId("621489fcf514a446bf1a98ea"),
    title: 'MongoDB insertOne',
    isbn: '0-7617-6154-3'
  }
]
```

2) Insert a document with an `_id` field example

The following example uses the `insertOne()` method to insert a document that has an `_id` field into the `books` collection:

```
db.books.insertOne({
  _id: 1,
  title: "Mastering Big Data",
  isbn: "0-9270-4986-4"
});
```

Output:

```
{ "acknowledged" : true, "insertedId" : 1 }
```

The following example attempts to insert another document whose `_id` field already exists into the `books` collection:

```
db.books.insertOne({  
  _id: 1,  
  title: "MongoDB for JS Developers",  
  isbn: "0-4925-3790-9"  
});
```

Since the `_id: 1` already exists, MongoDB threw the following exception:

```
WriteError({  
  "index" : 0,  
  "code" : 11000,  
  "errmsg" : "E11000 duplicate key error collection: bookstore.books index:  
  "op" : {  
    "_id" : 1,  
    "title" : "MongoDB for JS Developers",  
    "isbn" : "0-4925-3790-9"  
  }  
})
```



Summary

- Use `db.collection.insertOne()` method to insert a single document into a collection.
- If you explicitly provide a value for the `_id` field, you must ensure that the value is unique within the collection or you will get a duplicate key error.