

# How can I insert a document into a MongoDB collection only if it does not already exist?

Authored by  
**stats writer**

June 30, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I insert a document into a MongoDB collection only if it does not already exist?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=162361>

To insert a document into a MongoDB collection only if it does not already exist, the following steps can be followed:

1. First, connect to the MongoDB database and select the desired collection.
2. Use the "findOne" method to search for the document that you want to insert.
3. If the "findOne" method returns a null value, it means the document does not exist in the collection.
4. In this case, you can use the "insertOne" method to add the document to the collection.
5. However, if the "findOne" method returns a document, it means the document already exists in the collection.
6. In this case, you can choose to either skip the insertion or update the existing document.
7. To skip the insertion, you can simply move on to the next step in your code.
8. To update the existing document, you can use the "updateOne" or "replaceOne" method.
9. Once the desired action is completed, close the database connection.

By following these steps, you can ensure that a document is only inserted into a MongoDB collection if it does not already exist. This can help prevent duplicate documents and maintain data integrity within the collection.

## MongoDB: Insert if Not Exists

You can use the following syntax to insert a document into a collection in MongoDB only if it doesn't already exist:

```
db.teams.update(  
{  
  team : 'Hornets'  
},  
{  
  $setOnInsert: {team: 'Hornets', points: '58', rebounds:  
'20'}
```

```
},  
{upsert: true}  
)
```

This particular code checks if the field "team" has a value of "Hornets." If this value exists, then nothing will happen.

However, if this value does not exist then it will insert a document with specific values for the "team", "points", and "rebounds" fields.

The following example shows how to use this syntax in practice.

**Example: Insert if Not Exists in MongoDB**

Suppose we have a collection called teams with the following documents:

```
db.teams.insertOne({team: "Mavs", points: 30,  
rebounds: 8})
```

```
db.teams.insertOne({team: "Spurs", points: 35,  
rebounds: 12})
```

```
db.teams.insertOne({team: "Rockets", points: 20,  
rebounds: 7})
```

```
db.teams.insertOne({team: "Warriors", points: 25, rebounds: 5})
```

```
db.teams.insertOne({team: "Cavs", points: 23, rebounds: 9})
```

Suppose we use the following code to attempt to insert a document for the team "Mavs":

```
db.teams.update(  
{  
  team : 'Mavs'  
},  
{  
  $setOnInsert: {team: 'Mavs', points: '58', rebounds: '20'}  
},  
{upsert: true}  
)
```

Since the field "team" already contains information for the "Mavs", none of the documents will be modified.

However, suppose we use the following code to insert a document for the team "Hornets":

```
db.teams.update(  
{
```

```
{  
team : 'Hornets'  
},  
{  
$setOnInsert: {team: 'Hornets', points: '58', rebounds:  
'20'}  
},  
{upsert: true}  
)
```

Since the field "team" does not already contain information for the "Hornets", a new document will be added to the collection with the values that we specified for each field.

Here's what the updated collection looks like:

```
{_id: ObjectId("6203df361e95a9885e1e764a"),  
team: 'Mavs',  
points: 30,  
rebounds: 8 }  
{_id: ObjectId("6203df361e95a9885e1e764b"),  
team: 'Spurs',  
points: 35,
```

```
rebounds: 12 }
{ _id: ObjectId("6203df361e95a9885e1e764c"),
team: 'Rockets',
points: 20,
rebounds: 7 }
{ _id: ObjectId("6203df361e95a9885e1e764d"),
team: 'Warriors',
points: 25,
rebounds: 5 }
{ _id: ObjectId("6203df361e95a9885e1e764e"),
team: 'Cavs',
points: 23,
rebounds: 9 }
{ _id: ObjectId("6203e17de42bfba74fc73325"),
team: 'Hornets',
points: '58',
rebounds: '20' }
```

Notice that a new document has been added for the "Hornets" team.

#### Additional Resources

The following tutorials explain how to perform other common operations in MongoDB: