

Create Collection in MongoDB

We know that the data in MongoDB is stored in form of documents. These documents are stored in Collection and Collection is stored in Database. In the Previous tutorial we learned [How to create a Database in MongoDB](#). In this guide, I will show you two ways to create a collection in MongoDB.

Method 1: Creating the Collection in MongoDB on the fly

The cool thing about MongoDB is that you need not to create collection before you insert document in it. With a single command you can insert a document in the collection and the MongoDB creates that collection on the fly.

Syntax: **db.collection_name.insert({key:value, key:value...})**

For example:

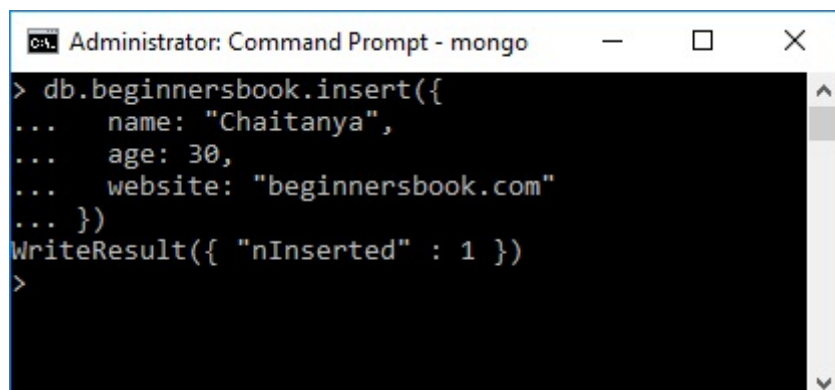
We don't have a collection `beginnersbook` in the database `beginnersbookdb`. This command will create the collection named "beginnersbook" on the fly and insert a document in it with the specified key and value pairs.

```
> use beginnersbookdb
switched to db beginnersbookdb
```

```
db.beginnersbook.insert({
  name: "Chaitanya",
  age: 30,
  website: "beginnersbook.com"
})
```

You would see this response in the command prompt.

```
WriteResult({ "nInserted" : 1 })
```

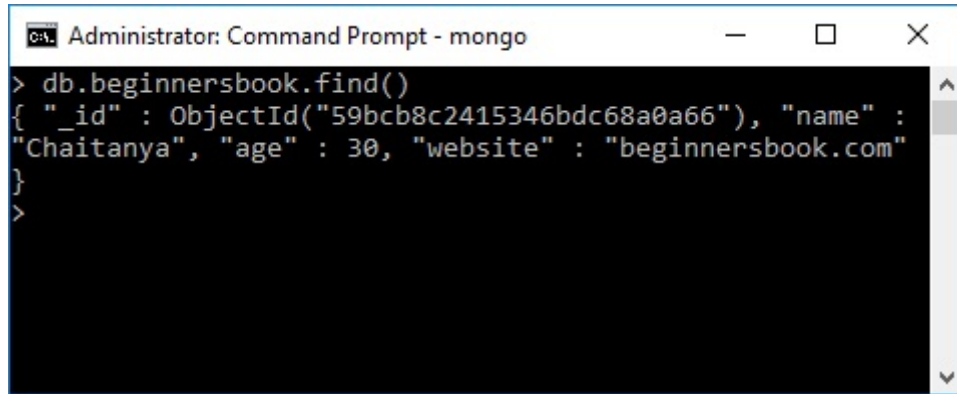


```
Administrator: Command Prompt - mongo
> db.beginnersbook.insert({
...   name: "Chaitanya",
...   age: 30,
...   website: "beginnersbook.com"
... })
WriteResult({ "nInserted" : 1 })
>
```

To check whether the document is successfully inserted, type the following command. It shows all the documents in the given collection.

Syntax: **db.collection_name.find()**

```
> db.beginnersbook.find()
{ "_id" : ObjectId("59bcb8c2415346bdc68a0a66"), "name" : "Chaitanya",
  "age" : 30, "website" : "beginnersbook.com" }
```



```
Administrator: Command Prompt - mongo
> db.beginnersbook.find()
{ "_id" : ObjectId("59bcb8c2415346bdc68a0a66"), "name" :
"Chaitanya", "age" : 30, "website" : "beginnersbook.com"
}
```

To check whether the collection is created successfully, use the following command.

```
show collections
```

This command shows the list of all the collections in the currently selected database.

```
> show collections
beginnersbook
```

Method 2: Creating collection with options before inserting the documents

We can also create collection before we actually insert data in it. This method provides you the options that you can set while creating a collection.

Syntax:

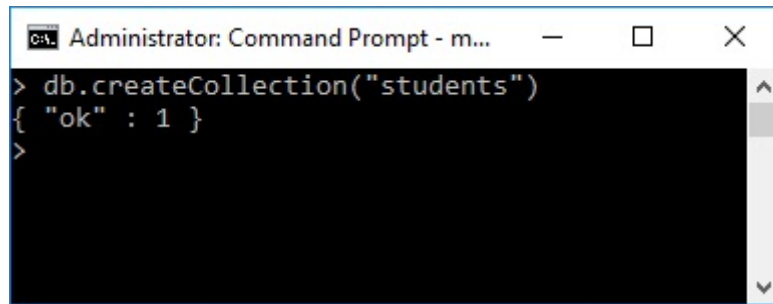
```
db.createCollection(name, options)
```

name is the collection name

and **options** is an optional field that we can use to specify certain parameters such as size, max number of documents etc. in the collection.

First lets see how this command is used for creating collection without any parameters:

```
> db.createCollection("students")
{ "ok" : 1 }
```

A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt - m...". The window has a black background with white text. The command prompt shows the command `> db.createCollection("students")` followed by the output `{ "ok" : 1 }` and a new prompt `>`.

Lets see the options that we can provide while creating a collection:

capped: type: boolean.

This parameter takes only true and false. This specifies a cap on the max entries a collection can have. Once the collection reaches that limit, it starts overwriting old entries. The point to note here is that when you set the capped option to true you also have to specify the size parameter.

size: type: number.

This specifies the max size of collection (capped collection) in bytes.

max: type: number.

This specifies the max number of documents a collection can hold.

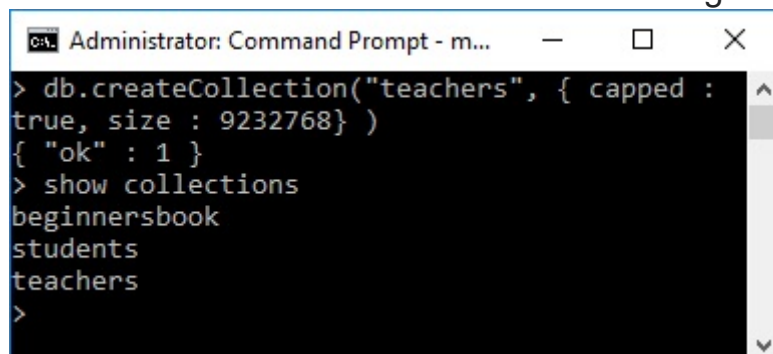
autoIndexId: type: boolean

The default value of this parameter is false. If you set it true then it automatically creates index field `_id` for each document. We will learn about index in the MongoDB indexing tutorial.

Lets see an example of capped collection:

```
db.createCollection("teachers", { capped : true, size : 9232768 } )
{ "ok" : 1 }
```

This command will create a collection named "teachers" with the max size of 9232768 bytes. Once this collection reaches that limit it will start overwriting old entries.

A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt - m...". The window has a black background with white text. The command prompt shows the command `> db.createCollection("teachers", { capped : true, size : 9232768 })` followed by the output `{ "ok" : 1 }`. Then, the command `> show collections` is entered, and the output lists three collections: `beginnersbook`, `students`, and `teachers`. A new prompt `>` is shown at the bottom.