

Prof R. Madana Mohana



BIG DATA ANALYTICS

MongoDB

Create & Drop Collection

<https://www.youtube.com/c/RASINENIMADANAMOHANA>

MongoDB: Create Collection

MongoDB: *Create Collection*

The `createCollection()` Method:

MongoDB `db.createCollection(name, options)` is used to create collection.

Syntax:

Basic syntax of `createCollection()` command is as follows:

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

MongoDB: *Create Collection*

The createCollection() Method:

MongoDB db.createCollection(name, options) is used to create collection.

- In the command, **name** is **name of collection** to be created.
- **Options** is a **document** and is used to specify **configuration of collection**.

MongoDB: *Create Collection*

The createCollection() Method:

MongoDB `db.createCollection(name, options)` is used to create collection.

Parameter	Type	Description
Name	String	Name of the collection to be created.
Options	Document	(Optional) Specify options about memory size and indexing.

MongoDB: *Create Collection*

The **createCollection()** Method:

Options parameter is **optional**, so we need to specify only the **name of the collection**.

*Following is the **list of options** we can use:*

MongoDB: *Create Collection*

The createCollection() Method:

Field	Type	Description
capped	Boolean	<p>(Optional) If true, enables a capped collection. Capped collection is a fixed size collection that automatically overwrites its oldest entries when it reaches its maximum size.</p> <p>If you specify true, you need to specify size parameter also.</p>
autoIndexID	Boolean	<p>(Optional) If true, automatically create index on _id field. Default value is false.</p>
size	number	<p>(Optional) Specifies a maximum size in bytes for a capped collection. If capped is true, then you need to specify this field also.</p>
max	number	<p>(Optional) Specifies the maximum number of documents allowed in the capped collection.</p>

MongoDB: *Create Collection*

The createCollection() Method:

While inserting the document, **MongoDB** first checks **size** field of capped collection, then it checks **max** field.

MongoDB: *Create Collection*

The `createCollection()` Method: *Examples*

Basic syntax of `createCollection()` method *without options* is as follows:

```
> use test
```

```
switched to db test
```

```
> db.createCollection("mycollection")
```

```
{ "ok" : 1 }
```

```
>
```

MongoDB: *Create Collection*

The `createCollection()` Method: *Examples*

Basic syntax of `createCollection()` method *without options* is as follows:

```
Command Prompt - mongo
> use test
switched to db test
> db.createCollection("mycollection")
{ "ok" : 1 }
> _
```

MongoDB: *Create Collection*

The `createCollection()` Method: *Examples*

We can check the *created collection* by using the command ***show collections.***

```
> show collections  
mycollection
```

MongoDB: *Create Collection*

The `createCollection()` Method: *Examples*

We can check the *created collection* by using the command *show collections*.

```
Command Prompt - mongo
> use test
switched to db test
> db.createCollection("mycollection")
{ "ok" : 1 }
> show collections
mycollection
>
```

MongoDB: *Create Collection*

The `createCollection()` Method: *Examples*

The following example shows the syntax of `createCollection()` method with few important options:

```
> db.createCollection("mycol", { capped : true, autoIndexID : true,
size : 6142800, max : 10000 } )
{
  "ok" : 0,
  "errmsg" : "BSON field 'create.autoIndexID' is an unknown
field.",
  "code" : 40415,
  "codeName" : "Location40415"
}
>
```

MongoDB: *Create Collection*

The `createCollection()` Method: *Examples*

The following example shows the syntax of `createCollection()` method with few important options:

```
Command Prompt - mongo
> db.createCollection("mycol", { capped : true, autoIndexID : true, size : 6142800, max :
10000 } )
{
  "ok" : 0,
  "errmsg" : "BSON field 'create.autoIndexID' is an unknown field.",
  "code" : 40415,
  "codeName" : "Location40415"
}
>
```

MongoDB: *Create Collection*

The createCollection() Method: *Examples*

- In **MongoDB**, we don't need to **create collection**.
- **MongoDB** creates collection automatically, when we **insert** some document.

MongoDB: *Create Collection*

The createCollection() Method: *Examples*

```
> db.bdacollection.insert({"name": "Big Data Analytics"})
```

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

```
> show collections
```

```
bdacollection
```

```
mycollection
```

```
mycollection1
```

```
>
```


MongoDB: *Create Collection*

The createCollection() Method: *Examples*

Command Prompt - mongo

```
> db.bdacollection.insert({"name": "Big Data Analytics"})
WriteResult({ "nInserted" : 1 })
> show collections
bdacollection
mycollection
mycollection1
>
```

MongoDB: Drop Collection

MongoDB: *Drop Collection*

The drop() Method:

MongoDB's `db.collection.drop()` is used to **drop** a **collection** from the **database**.

Syntax:

Basic syntax of `drop()` command is as follows:

```
db.COLLECTION_NAME.drop()
```

MongoDB: *Drop Collection*

The drop() Method: *Example*

First, check the **available collections** into our database **mydb**.

```
> use mydb
```

```
switched to db mydb
```

```
> db
```

```
mydb
```

```
> show dbs
```

```
admin      0.000GB
```

```
config    0.000GB
```

```
local     0.000GB
```

```
test      0.000GB
```

MongoDB: *Drop Collection*

The drop() Method: *Example*

```
> db.student.insert({"name": "Ram"})
```

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

```
> show dbs
```

```
admin      0.000GB
```

```
config    0.000GB
```

```
local     0.000GB
```

```
mydb     0.000GB
```

```
test      0.000GB
```

MongoDB: *Drop Collection*

The drop() Method: *Example*

```
> use mydb
```

```
switched to db mydb
```

```
> show collections
```

```
student
```

MongoDB: *Drop Collection*

The drop() Method: *Example*

First, check the **available collections** into our database **mydb**.

```
Command Prompt - mongo
> use mydb
switched to db mydb
> db
mydb
> show dbs
admin    0.000GB
config  0.000GB
local   0.000GB
test    0.000GB
> db.student.insert({"name":"Ram"})
WriteResult({ "nInserted" : 1 })
> show dbs
admin    0.000GB
config  0.000GB
local   0.000GB
mydb    0.000GB
test    0.000GB
> use mydb
switched to db mydb
> show collections
student
>
```

MongoDB: *Drop Collection*

The drop() Method: *Example*

Now **drop** the **collection** with the name **student**.

```
> show collections
```

```
student
```

```
> db.student.drop()
```

```
true
```


MongoDB: *Drop Collection*

The drop() Method: *Example*

Now **drop** the **collection** with the name **student**.

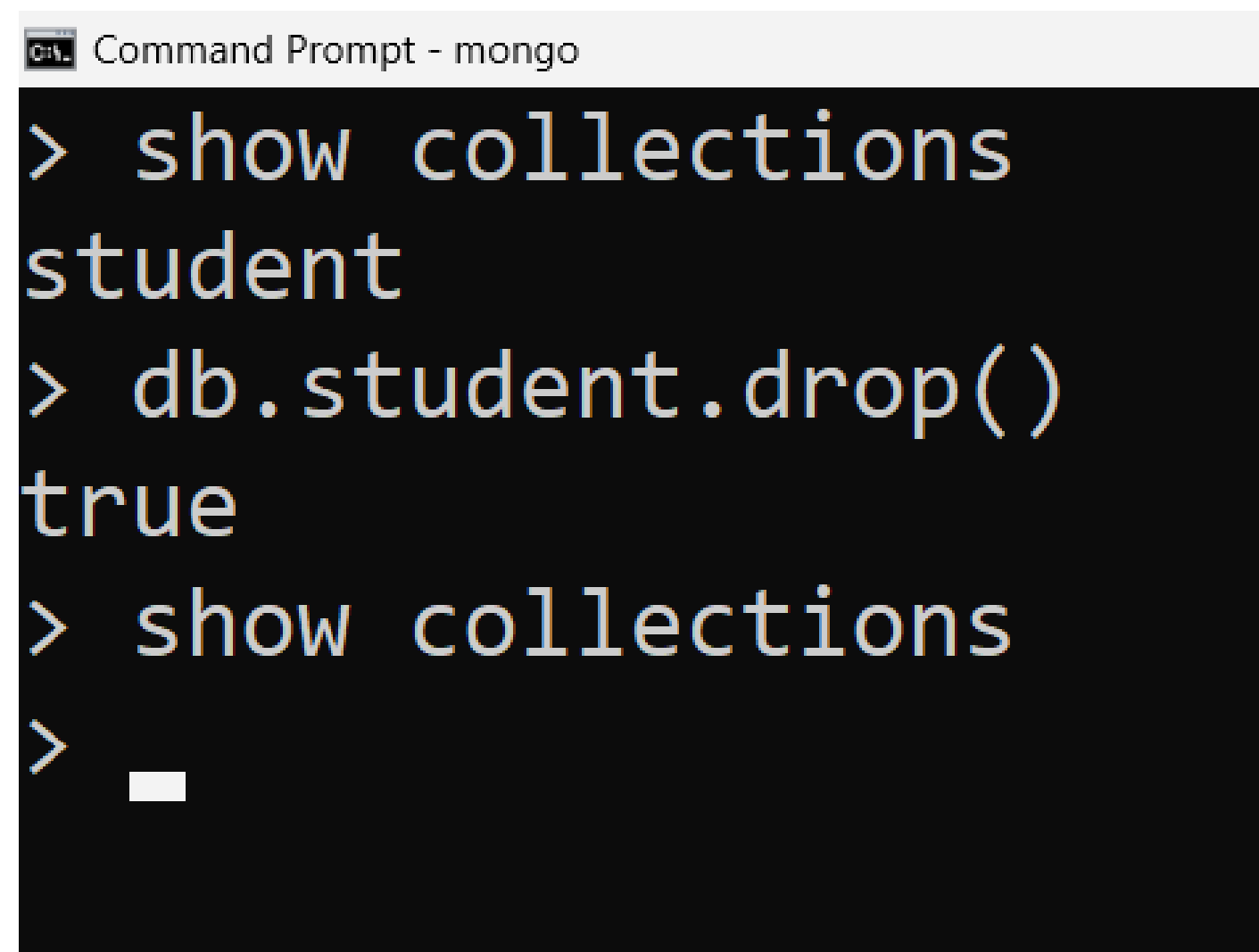
```
Command Prompt - mongo
> show collections
student
> db.student.drop()
true
> _
```

MongoDB: *Drop Collection*

The drop() Method: *Example*

Again check the list of collections into database.

```
> show collections
```



```
Command Prompt - mongo
> show collections
student
> db.student.drop()
true
> show collections
> _
```

MongoDB: *Drop Collection*

The drop() Method: *Example*

drop() method will return **true**, if the selected collection is **dropped successfully**, otherwise it will return **false**.