MongoDB Indexing Tutorial with Example

An **index** in MongoDB is a special data structure that holds the data of few fields of documents on which the index is created. Indexes improve the speed of search operations in database because instead of searching the whole document, the search is performed on the indexes that holds only few fields. On the other hand, having too many indexes can hamper the performance of insert, update and delete operations because of the additional write and additional data space used by indexes.

How to create index in MongoDB

Syntax:

```
db.collection_name.createIndex({field_name: 1 or -1})
```

The value 1 is for ascending order and -1 is for descending order.

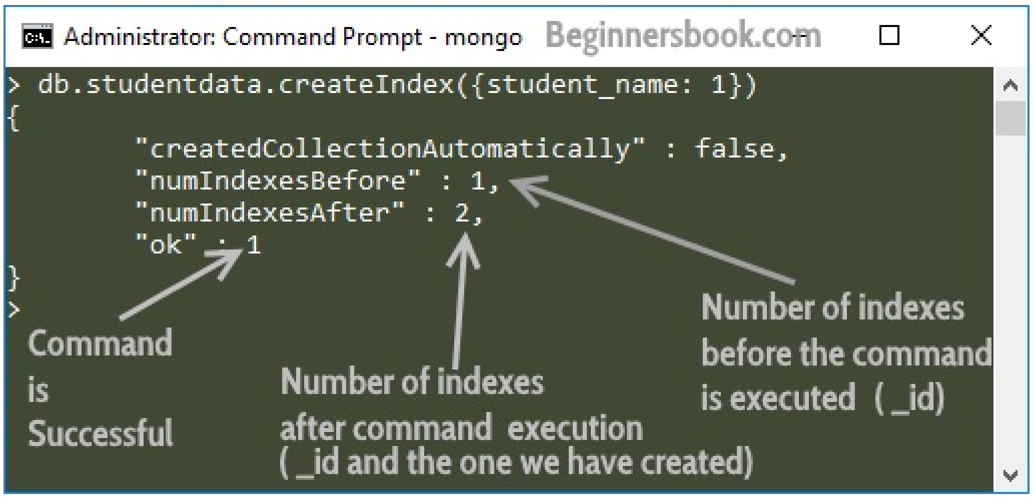
For example, I have a collection `studentdata`. The documents inside this collection have following fields:
student_name, student_id and student_age

Lets say I want to create the index on student_name field in ascending order:

```
db.studentdata.createIndex({student_name: 1})
```

Output:

```
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```



```
> db.studentdata.createIndex({student_name: 1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
>
```

Administrator: Command Prompt - mongo
Beginnersbook.com
Command is Successful
Number of indexes after command execution (_id and the one we have created)
Number of indexes before the command is executed (_id)

We have created the index on student_name which means when someone searches the document based on the student_name, the search will be faster because the index will be used for this search. So this is important to create the index on the field that will be frequently searched in a collection.

MongoDB – Finding the indexes in a collection

We can use getIndexes() method to find all the indexes created on a collection. The syntax for this method is:

```
db.collection_name.getIndexes()
```

So to get the indexes of `studentdata` collection, the command would be:

```
> db.studentdata.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_",
                "ns" : "test.studentdata"
        },
        {
                "v" : 2,
                "key" : {
                        "student_name" : 1
                },
                "name" : "student_name_1",
                "ns" : "test.studentdata"
        }
]
```

The output shows that we have two indexes in this collection. The default index created on _id and the index that we have created on student_name field.

MongoDB – Drop indexes in a collection

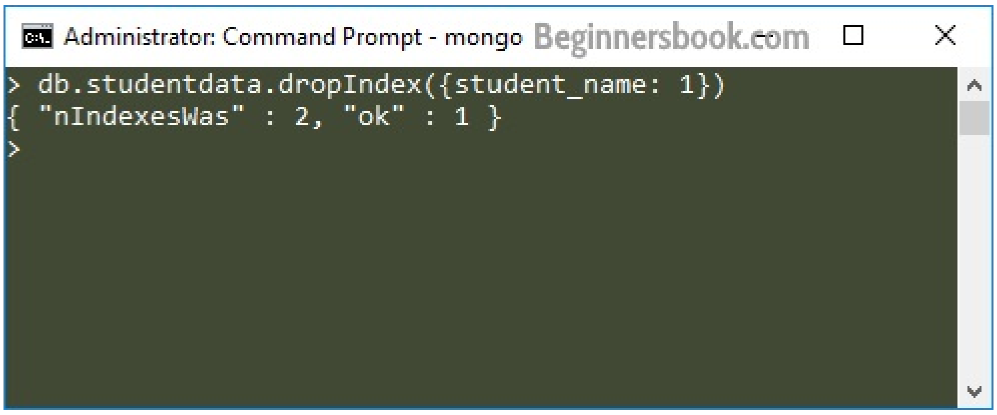
You can either drop a particular index or all the indexes.

Dropping a specific index:
For this purpose the dropIndex() method is used.

```
db.collection_name.dropIndex({index_name: 1})
```

Lets drop the index that we have created on `student_name` field in the collection `studentdata`. The command for this:

```
db.studentdata.dropIndex({student_name: 1})
```



```
> db.studentdata.dropIndex({student_name: 1})
{ "nIndexesWas" : 2, "ok" : 1 }
>
```

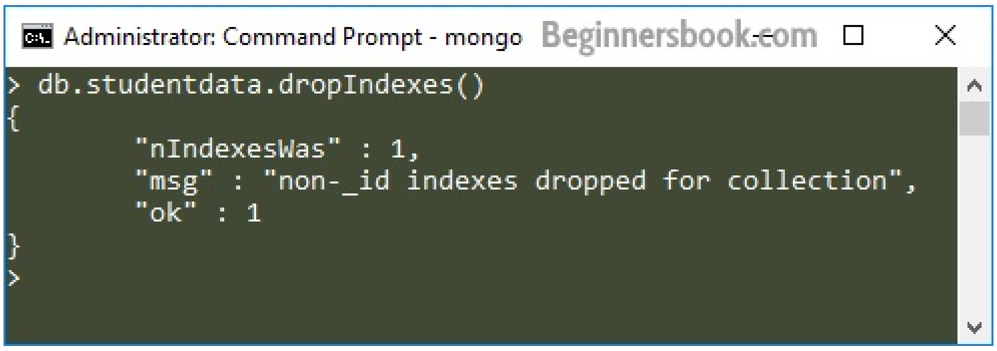
nIndexesWas: It shows how many indexes were there before this command got executed
ok: 1: This means the command is executed successfully.

Dropping all the indexes:
To drop all the indexes of a collection, we use dropIndexes() method.
Syntax of dropIndexs() method:

```
db.collection_name.dropIndexes()
```

Lets say we want to drop all the indexes of `studentdata` collection.

```
db.studentdata.dropIndexes()
```



```
> db.studentdata.dropIndexes()
{
        "nIndexesWas" : 1,
        "msg" : "non-_id indexes dropped for collection",
        "ok" : 1
}
>
```

The message “non-_id indexes dropped for collection” indicates that the default index _id will still remain and cannot be dropped. This means that using this method we can only drop indexes that we have created, we can’t drop the default index created on _id field.