

# How to Create User & add Role in MongoDB

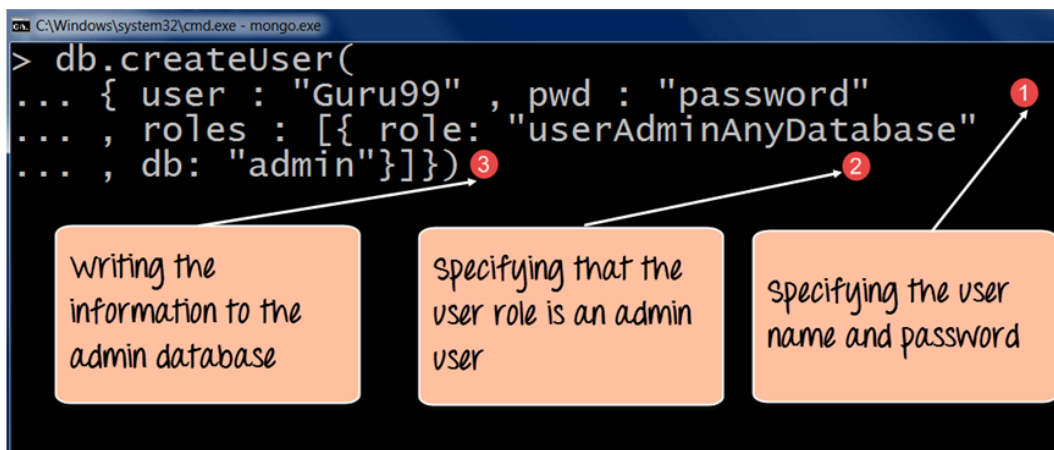
## Details

Last Updated: 03 January 2020

### MongoDB Create Administrator User

Creating a user administrator in MongoDB is done by using the createUser method. The following example shows how this can be done.

```
db.createUser(  
{  
  user: "Guru99",  
  pwd: "password",  
  
  roles:[{role: "userAdminAnyDatabase" , db:"admin"}]})
```



## Code Explanation:

1. The first step is to specify the "username" and "password" which needs to be created.
2. The second step is to assign a role for the user. Since it needs to be a database administrator in which case we have assigned to the "userAdminAnyDatabase" role. This role allows the user to have administrative privileges to all databases in MongoDB.
3. The db parameter specifies the admin database which is a special Meta database within MongoDB which holds the information for this user.

If the command is executed successfully, the following Output will be shown:

## Output:

```
> db.createUser(
... { user : "Guru99" , pwd : "password"
... , roles : [{ role: "userAdminAnyDatabase"
... , db: "admin"}]})
Successfully added user: {
  "user" : "Guru99",
  "roles" : [
    {
      "role" : "userAdminAnyDatabase",
      "db" : "admin"
    }
  ]
}
```

Shows the user "guru99" created

Shows the role which was assigned

The output shows that a user called "Guru99" was created and that user has privileges over all the databases in MongoDB.

## MongoDB Create User for Single Database

To create a user who will manage a single database, we can use the same command as mentioned above but we need to use the "userAdmin" option only.

The following example shows how this can be done;

```
C:\Windows\system32\cmd.exe - mongo.exe
> db.createUser(
.. 1 { user : "Employeeadmin" , pwd : "password"
... , roles : [{role: "userAdmin" 2
... 3 , db: "Employee"}]})
```

We are creating an administrator only in the Employee database

We are using the userAdmin role

```
db.createUser(  
{  
    user: "Employeeadmin",  
  
    pwd: "password",  
  
    roles:[{role: "userAdmin" , db:"Employee"}]})
```

### Code Explanation:

1. The first step is to specify the "username" and "password" which needs to be created.
2. The second step is to assign a role for the user which in this case since it needs to be a database administrator is assigned to the "userAdmin" role. This role allows the user to have administrative privileges only to the database specified in the db option.
3. The db parameter specifies the database to which the user should have administrative privileges on.

If the command is executed successfully, the following Output will be shown:

### Output:

```
C:\Windows\system32\cmd.exe - mongo.exe  
> db.createUser(  
... { user : "Employeeadmin" , pwd : "password"  
... , roles : [{role: "userAdmin"  
... ,db: "Employee"}]})  
Successfully added user: {  
  "user" : "Employeeadmin",  
  "roles" : [  
    {  
      "role" : "userAdmin",  
      "db" : "Employee"  
    }  
  ]  
}
```

Shows the user created

Shows the role which was assigned to the user and to which database

The output shows that a user called "Employeeadmin" was created and that user has privileges only on the "Employee" database.

## Managing users

First understand the roles which you need to define. There is a whole list of role available in MongoDB. For example, there is a "read role" which only allows read only access to databases and then there is the "readwrite" role which provides read and write access to the

database , which means that the user can issue the insert, delete and update commands on collections in that database.

```
db.createUser(  
{  
  user: "Mohan",  
  pwd: "password",  
  roles:[  
    {  
      role: "read" , db:"Marketing"},  
      role: "readwrite" , db:"Sales"}  
    ]  
})
```

A diagram illustrating the MongoDB `db.createUser()` command. The command is shown in a code block. A red rectangular box highlights the `roles` array within the command. An orange rounded rectangular box with a black border contains the text "Specifying the different roles for the user". A blue arrow points from the orange box to the red box, indicating that the roles array is used to specify permissions for the user.

```
db.createUser(  
{  
  user: "Mohan",  
  
  pwd: "password",  
  
  roles:[  
    {  
      role: "read" , db:"Marketing"},  
  
      role: "readWrite" , db:"Sales"}  
    ]  
})
```

The above code snippet shows that a user called Mohan is created, and he is assigned multiple roles in multiple databases. In the above example, he is given read only permission to the "Marketing" database and readWrite permission to the "Sales" database.