

How to create schema in  
node js “mongoose”

## Create Project

C:\Windows\System32\cmd.exe

```
D:\reactshrenik\Ex8-Express-Js\#7-Express-MongoDb-Create-Schema>mkdir mongo-schema-demo
```

```
D:\reactshrenik\Ex8-Express-Js\#7-Express-MongoDb-Create-Schema>cd mongo-schema-demo
```

```
D:\reactshrenik\Ex8-Express-Js\#7-Express-MongoDb-Create-Schema\mongo-schema-demo>npm init --yes
```

```
Wrote to D:\reactshrenik\Ex8-Express-Js\#7-Express-MongoDb-Create-Schema\mongo-schema-demo\package.json:
```

```
{
  "name": "mongo-schema-demo",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

```
D:\reactshrenik\Ex8-Express-Js\#7-Express-MongoDb-Create-Schema\mongo-schema-demo>
```

## Install Express Js

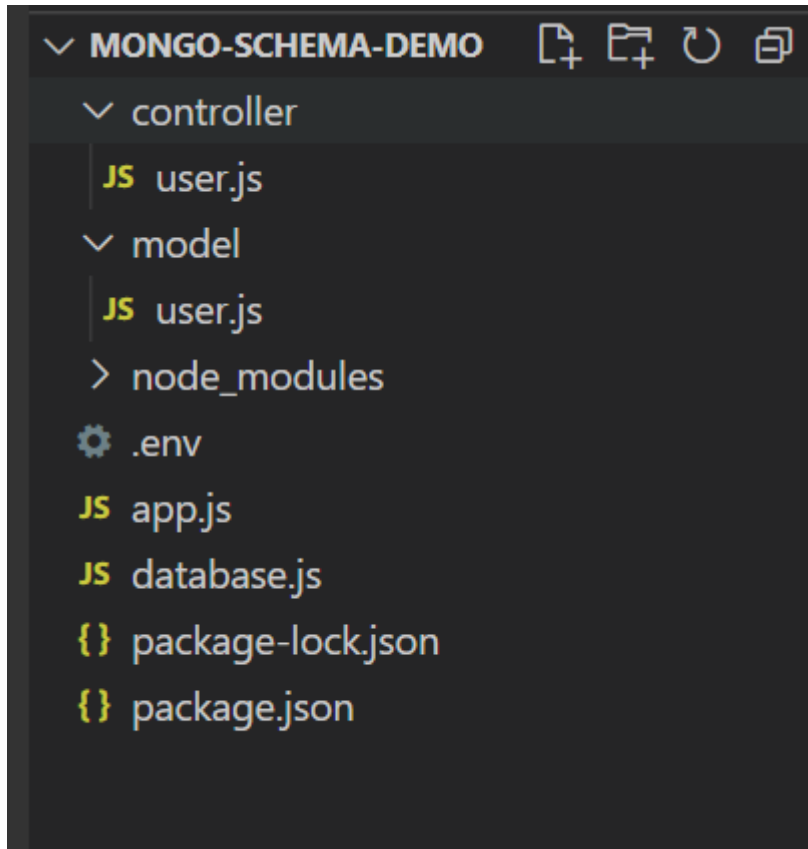
C:\Windows\System32\cmd.exe

```
D:\reactshrenik\Ex8-Express-Js\#7-Express-MongoDb-Create-Schema\mongo-schema-demo>npm i --save express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN mongo-schema-demo@1.0.0 No description
npm WARN mongo-schema-demo@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 4.212s
found 0 vulnerabilities

D:\reactshrenik\Ex8-Express-Js\#7-Express-MongoDb-Create-Schema\mongo-schema-demo>_
```

## Create following Project Structure



Install mongoose, dotenv modules

`npm i --save mongoose dotenv`

## Connect Database using mongoose

```
.env
1 PORT=3002
2 DB_URL=mongodb://localhost:27017/mongooseTest
```

```
JS database.js
JS database.js > connectMongoDb > <function>
1 const mongoose = require('mongoose');
2 const db_url = process.env.DB_URL;
3
4
5 console.log(mongoose.connection.readyState);
6
7 let connectMongoDb = () => {
8   return new Promise((resolve, reject) => {
9     mongoose.connect(db_url, {useNewUrlParser:true,useUnifiedTopology: true});
10
11     mongoose.connection
12       .once("open", () => {
13         console.log("Connected");
14         const conn_status = mongoose.connection.readyState;
15         resolve(conn_status);
16       })
17       .on("error", (error) => {
18         console.log("Error :" + error);
19         reject("Mongo Db Connection Failed");
20       });
21   });
22 }
23
24 module.exports.connectMongoDb = connectMongoDb;
```

Create **user** model

```
JS user.js X
model > JS user.js > ...
1  const mongoose = require("mongoose");
2
3  const userSchema = mongoose.Schema({
4    username:{
5      type:String,
6      required:true
7    },
8    email:{
9      type:String,
10     required:true,
11     unique:true
12   },
13   password:{
14     type:String,
15     required:true
16   },
17   isActive:{
18     type:Boolean,
19     default:false
20   },
21   createdAt:{
22     type>Date,
23     default>Date.now()
24   }
25 });
26
27 mongoose.model("users",userSchema);
28
29 module.exports = mongoose.model("users");
```

Create **user** controller to handle api request and perform insert/update/delete/read operations

```
JS user.js X
controller > JS user.js > ...
1  const router = require("express").Router();
2  const user = require("../model/user");
3
4  router.all("/", (request, response)=>{
5      response.setHeader('Content-Type', 'application/json');
6      response.end(JSON.stringify({"status":"User Controller Working.."},null,3));
7  });
8
9
10 module.exports = router;
```

Import "controller" (line 4)  
and Define middleware  
function to connect database

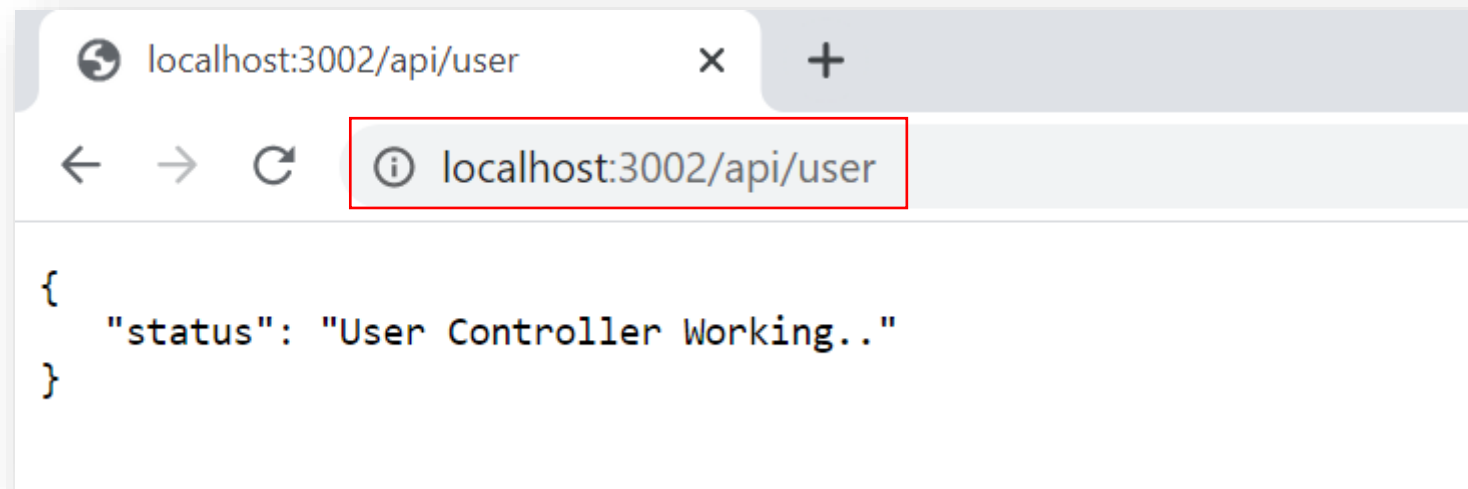
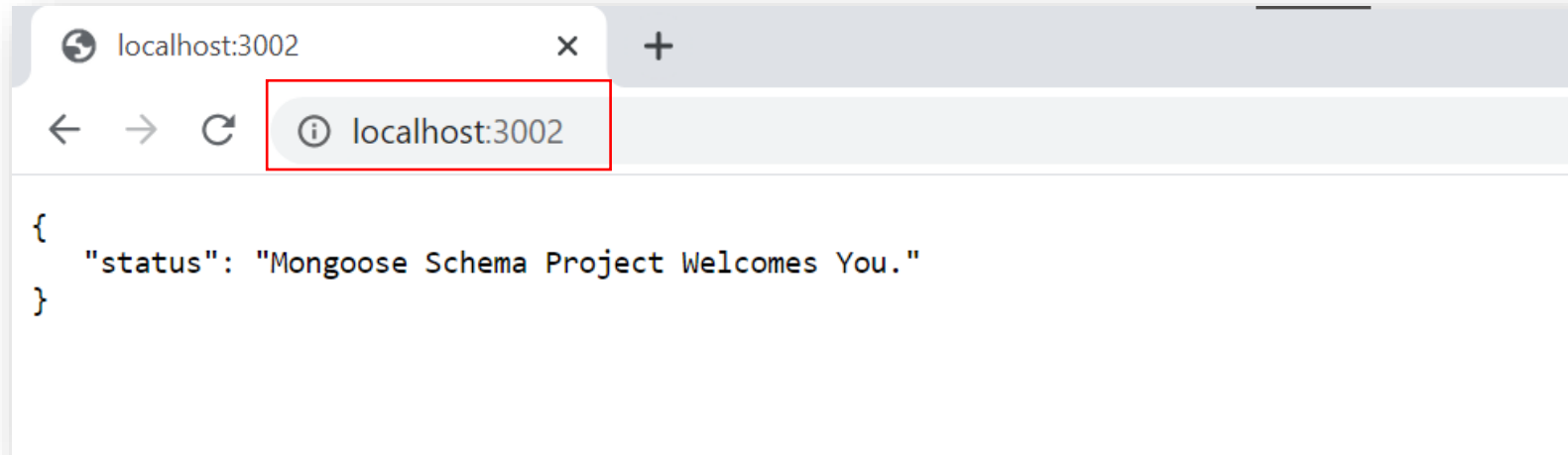
JS app.js X

JS app.js > [🔗] middleware

```
1  const express = require("express");
2  require("dotenv").config();
3  const database = require("../database");
4  const userController = require("../controller/user");
5
6  const app=express();
7  const port=process.env.PORT;
8
9  let middleware = (request, response, next)=>{
10     database.connectMongoDb().then((arg)=>{
11         console.log('Connection Status : ' + arg);
12     }).catch((error)=>{
13         console.log(error);
14     });
15
16     next();
17 }
18
19 app.use(middleware);
20 app.use("/api/user",userController);
21
```



Run Project and check following two urls



**Note:**

After Insert Document new schema and collection get create in mongodb database.

After insert document, open robo 3t or compass to check record