

How to insert record in  
mongodb using mongoose?

## Install **express-validator** module to validate request.body key:values

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  powershell + v

PS D:\reactshrenik\Ex8-Express-Js\#7-Express-MongoDb-Create-Schema\mongo-schema-demo> npm i --save express-validator
npm WARN mongo-schema-demo@1.0.0 No description
npm WARN mongo-schema-demo@1.0.0 No repository field.

+ express-validator@6.11.1
added 3 packages from 8 contributors and audited 83 packages in 4.912s
found 0 vulnerabilities

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

### Import module (Line 3)

```
JS user.js  X
controller > JS user.js > ...
1  const router = require("express").Router();
2  const user = require("../model/user");
3  const { check, validationResult } = require("express-validator");
```

Install **body-parser** module to work with form data sent to server from browser

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  powershell + ^ X

PS D:\reactshrenik\Ex8-Express-Js\#7-Express-MongoDb-Create-Schema\mongo-schema-demo> npm i --save body-parser
npm WARN mongo-schema-demo@1.0.0 No description
npm WARN mongo-schema-demo@1.0.0 No repository field.

+ body-parser@1.19.0
updated 1 package and audited 84 packages in 3.327s
found 0 vulnerabilities

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

Import module (Line 4) and setup middleware

```
4  const bodyParser = require("body-parser");
5
6  //middleware setup
7  router.use(bodyParser.json());
8  router.use(bodyParser.urlencoded({ extended:true }));
9
```

## Writer router.post() method to perform insert document

```
17 router.post("/create",
18   [
19     check("username").not().isEmpty().trim().escape(),
20     check("password").not().isEmpty().trim().escape(),
21     check("email").isEmail().normalizeEmail()
22   ],
23   (request, response)=>{
24     response.setHeader('Content-Type', 'application/json');
25
26     console.log(request['username']);
27
28     const errors = validationResult(request);
29     console.log(errors);
30
31     if(!errors.isEmpty())
32     {
33       response.status(422).end(JSON.stringify({status:false, message:"Form Validation Error", errors:errors.array()}),
34     }
35
36     response.end(JSON.stringify({status:true, message:"User data is correct"},null,3));
37   });
38
39 module.exports = router;
```

Call create() function and write callback() to handle response with error status

```
36 //response.end(JSON.stringify({status:true, message:"User data is correct"},null,3));
37
38 user.create(
39     {
40         username:request.body.username,
41         password:request.body.password,
42         email:request.body.email
43     },
44     (error, result)=>{
45         if(error){
46             response.status(500).end(JSON.stringify({status:false, message:"Failed to insert record", errors:errors
47         }
48         else{
49             response.end(JSON.stringify({status:true, message:"Insert Successfull"},null,3));
50         }
51     }
52 )
53 });
```

## Using “postman” send form-data using api and check insert successful or not

The screenshot displays the Postman application interface. At the top, the request is identified as a POST to `localhost:3002/api/user/create`. The environment is set to "No Environment". The request is titled "Untitled Request".

The request configuration shows the method as POST and the URL as `localhost:3002/api/user/create`. The "Body" tab is selected, and the data type is set to "x-www-form-urlencoded". The form data includes:

Key	Value	Description
username	shrenik	
password	123456	
email	emailtoshrenik@gmail.com	

The response is shown at the bottom, with a status of 200 OK, a time of 132 ms, and a size of 212 B. The response body is displayed in "Pretty" format as JSON:

```
1 {
2   "status": true,
3   "message": "Insert Successfull"
4 }
```

## Screenshot of robo 3T

New release available. Find out [what's new in Robo 3T](#) - [Download here](#).

shrenik77 (4)

- > System
- > config
- ▼ mongooseTest
  - ▼ Collections (1)
    - ▼ users
      - > Indexes
      - > Functions
      - > Users

db.getCollection('users').find({})

shrenik77 localhost:27017 mongooseTest

```
db.getCollection('users').find({})
```

users 0.001 sec. 0 50

	_id	isActive	createdOn	username	password	email	_v
1	ObjectId("...	false	2021-05-1...	shrenik	123456	emailtoshr...	0