

# Node Auth Tutorial (JWT)

## #5 - Mongoose Validation



- Terminal

```
npm install validator
```



```
const mongoose = require('mongoose');  
const { isEmail } = require('validator');
```

```
const userSchema = new mongoose.Schema({  
  email: {  
    type: String,  
    required: [true, 'Please enter an email'],  
    unique: true,  
    lowercase: true,  
    validate: [isEmail, 'Please enter a valid email']  
  },  
  password: {  
    type: String,  
    required: [true, 'Please enter a password'],  
    minlength: [6, 'Minimum password length is 6 characters'],  
  }  
});
```

```
const User = mongoose.model('user', userSchema);  
  
module.exports = User;
```

**Insert code**



```
const User = require("../models/User");  
// handle errors  
const handleError = (err) => {  
  console.log(err.message, err.code);  
  let errors = { email: '', password: '' };  
  // duplicate email error  
  if (err.code === 11000) {  
    errors.email = 'that email is already registered';  
    return errors;  
  }  
  // validation errors  
  if (err.message.includes('user validation failed')) {  
    Object.values(err.errors).forEach(({ properties }) => {  
      errors[properties.path] = properties.message;  
    });  
  }  
  return errors;  
}
```

~~// controller actions~~

```
...  
module.exports.signup_post = async (req, res) => {  
  const { email, password } = req.body;  
  try {  
    const user = await User.create({ email, password });  
    res.status(201).json(user);  
  }  
  catch(err) {  
    const errors = handleError(err);  
    res.status(400).json({ errors });  
  }  
}
```

**Insert code**



POST

http://localhost:3000/signup

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {  
2   "email": "mario",  
3   "password": "321"  
4 }
```

Body

Cookies

Headers (7)

Test Results



Status: 400 Bad Request

Time: 16 ms

Size: 347 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "errors": {  
3     "email": "Please enter a valid email",  
4     "password": "Minimum password length is 6 characters"  
5   }  
6 }
```



POSTMAN

POST

http://localhost:3000/signup

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {  
2   "email": "mario@gmail.com",  
3   "password": "321"  
4 }
```

Body

Cookies

Headers (7)

Test Results



Status: 400 Bad Request

Time: 14 ms

Size: 320 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "errors": {  
3     "email": "",  
4     "password": "Minimum password length is 6 characters"  
5   }  
6 }
```



POSTMAN

POST

http://localhost:3000/signup

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {  
2   "email": "mario@gmail.com",  
3   "password": "321123"  
4 }
```

Body

Cookies

Headers (7)

Test Results



Status: 201 Created

Time: 172 ms

Size: 328 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "email": "mario@gmail.com",  
3   "password": "321123",  
4   "_id": "612cf0d42f0e7155bef40b04",  
5   "__v": 0  
6 }
```



POSTMAN

POST

http://localhost:3000/signup

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {  
2   ...."email": "mario@gmail.com",  
3   ...."password": "321123"  
4 }
```

Body

Cookies

Headers (7)

Test Results



Status: 400 Bad Request

Time: 171 ms

Size: 313 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "errors": {  
3     "email": "that email is already registered",  
4     "password": ""  
5   }  
6 }
```



POSTMAN