

Node Auth Tutorial (JWT)

#6 - Mongoose Hooks



In this Node auth tutorial we'll take a look at how mongoose hooks can be used to fire code at different points when documents are saved to the database.



express

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

```
const userSchema = new mongoose.Schema({
  ...
});
```

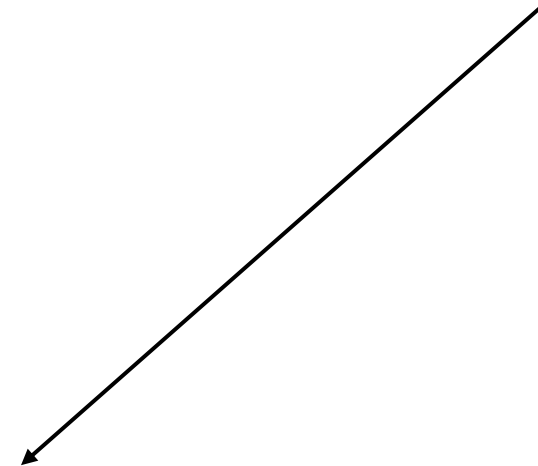
```
// fire a function after doc saved to db
userSchema.post('save', function (doc, next) {
  console.log('new user was created & saved', doc);
  next();
});

// fire a function before doc saved to db
userSchema.pre('save', function (next) {
  console.log('user about to be created & saved', this);
  next();
});
```

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

```
module.exports = User;
```

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": "yoshi@gmail.com",  
3   "password": "321123"  
4 }
```

Body

Cookies

Headers (7)

Test Results



Status: 201 Created

Time: 194 ms

Size: 328 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "email": "yoshi@gmail.com",  
3   "password": "321123",  
4   "_id": "612cf549e4ddecbd2e52449e",  
5   "__v": 0  
6 }
```



POSTMAN

Terminal

```
[nodemon] starting `node app.js`  
user about to be created & saved {  
  email: 'yoshi@gmail.com',  
  password: '321123',  
  _id: new ObjectId("612cf549e4ddecabd2e52449e")  
}  
new user was created & saved {  
  email: 'yoshi@gmail.com',  
  password: '321123',  
  _id: new ObjectId("612cf549e4ddecabd2e52449e"),  
  __v: 0  
}  
█
```