

Node Auth Tutorial (JWT)

#12 - Logging Users in



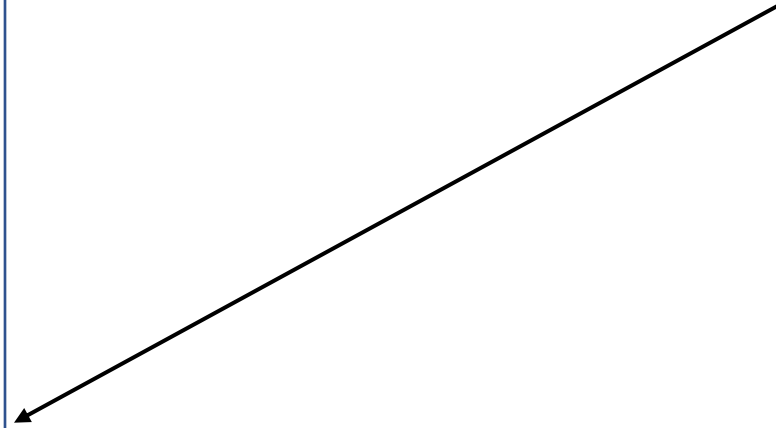
In this Node auth tutorial we'll begin to see how to implement our login process.



express

...

```
<script>
const form = document.querySelector('form');
const emailError = document.querySelector('.email.error');
const passwordError = document.querySelector('.password.error');
form.addEventListener('submit', async (e) => {
  e.preventDefault();
  // reset errors
  emailError.textContent = '';
  passwordError.textContent = '';
  // get values
  const email = form.email.value;
  const password = form.password.value;
  try {
    const res = await fetch('/login', {
      method: 'POST',
      body: JSON.stringify({ email, password }),
      headers: {'Content-Type': 'application/json'}
    });
    const data = await res.json();
    console.log(data);
    if (data.errors) {
      emailError.textContent = data.errors.email;
      passwordError.textContent = data.errors.password;
    }
    if (data.user) {
      location.assign('/');
    }
  }
  catch (err) {
    console.log(err);
  }
});
</script>
```

Insert code

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

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

// fire a function before doc saved to db
userSchema.pre('save', async function(next) {
  ...
});
```

```
// static method to login user
userSchema.statics.login = async function(email, password) {
  const user = await this.findOne({ email });
  if (user) {
    const auth = await bcrypt.compare(password, user.password);
    if (auth) {
      return user;
    }
    throw Error('incorrect password');
  }
  throw Error('incorrect email');
};
```

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

module.exports = User;
```

Insert code



```
const User = require("../models/User");
const jwt = require('jsonwebtoken');

// handle errors
...

// create json web token
...

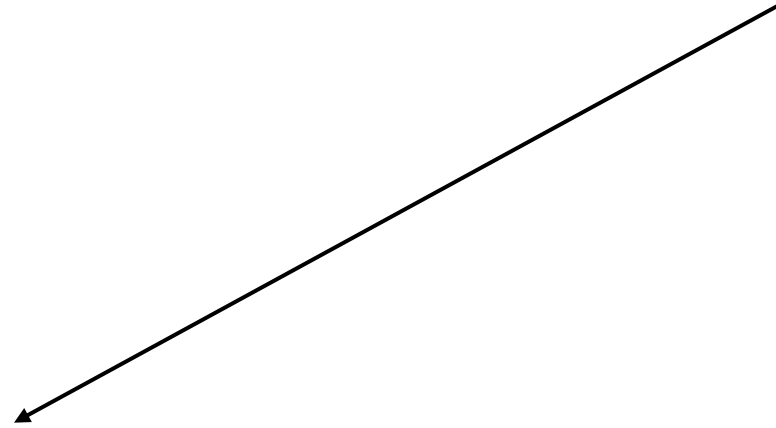
// controller actions
...
```

```
module.exports.login_post = async (req, res) => {
  const { email, password } = req.body;
```

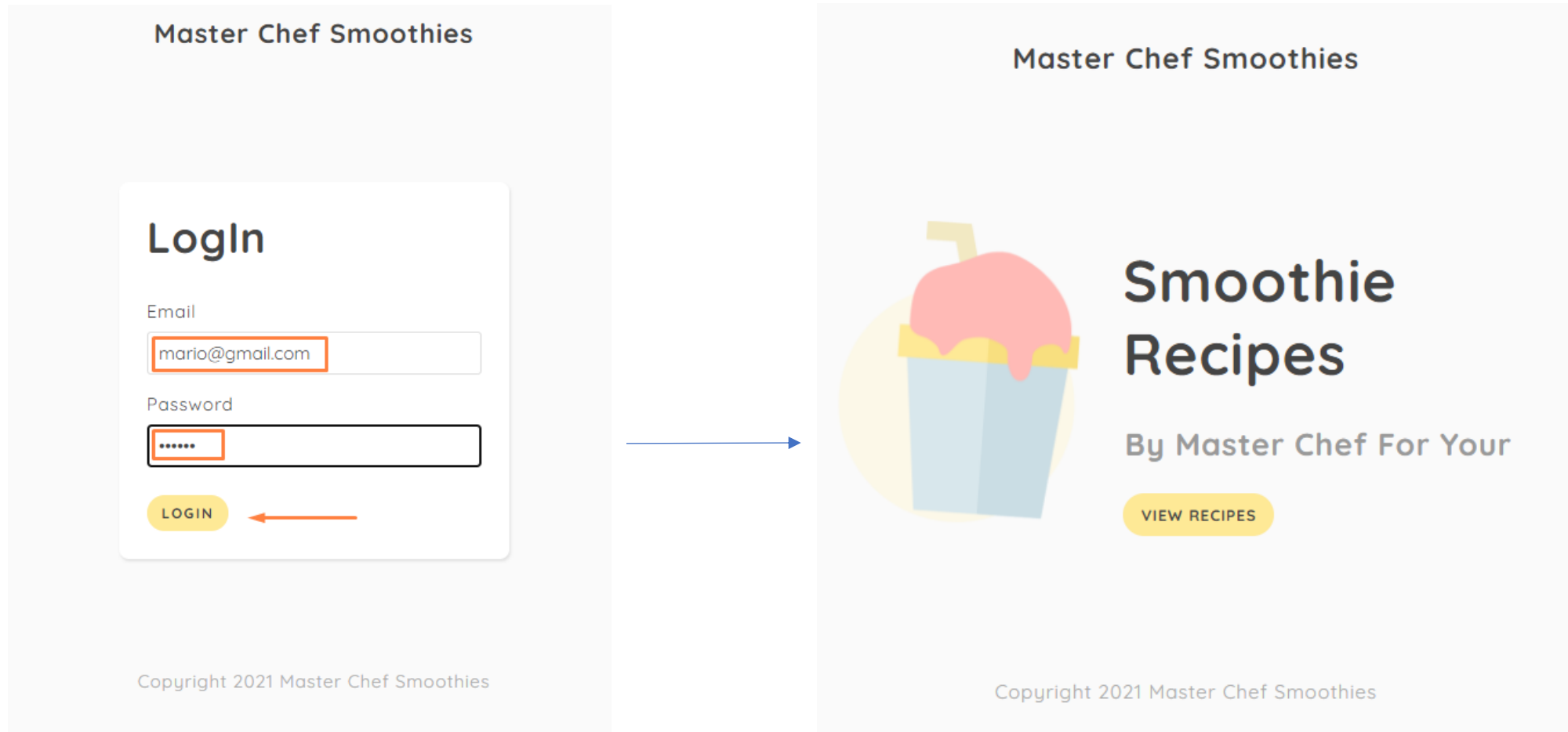
```
  try {
    const user = await User.login(email, password);
    res.status(200).json({ user: user._id });
  } catch (err) {
    res.status(400).json({});
  }
}
```

```
}
```

Insert code



Go to <http://localhost:3000/login>



```
...
// handle errors
const handleErrors = (err) => {
  console.log(err.message, err.code);
  let errors = { email: '', password: '' };

```

```
  // incorrect email
  if (err.message === 'incorrect email') {
    errors.email = 'That email is not registered';
  }

  // incorrect password
  if (err.message === 'incorrect password') {
    errors.password = 'That password is incorrect';
  }

```

```
  ...
  return errors;
}

```

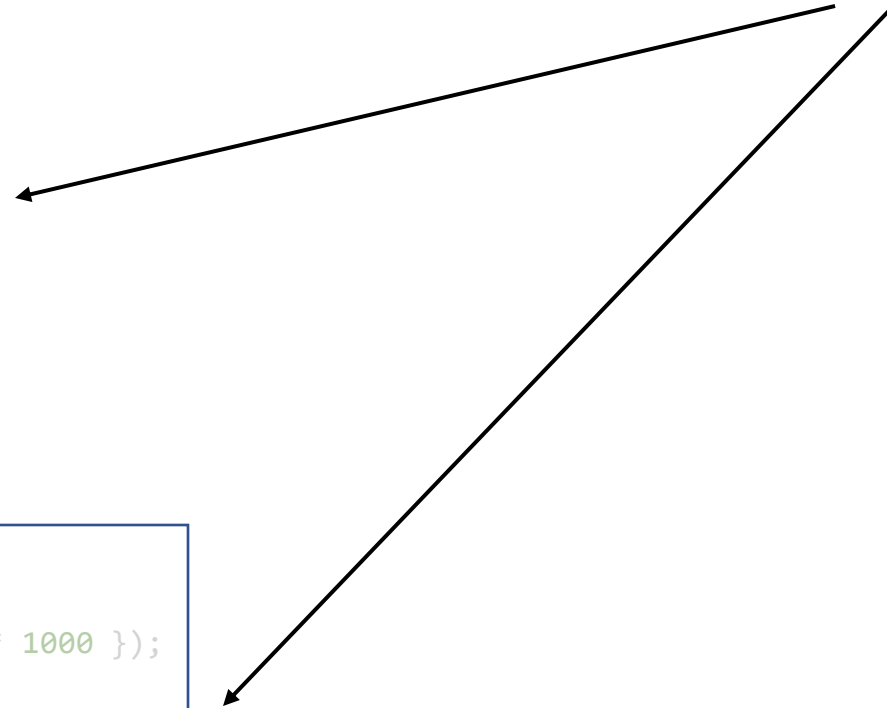
```
...
module.exports.login_post = async (req, res) => {
  const { email, password } = req.body;

```

```
  try {
    const user = await User.login(email, password);
    const token = createToken(user._id);
    res.cookie('jwt', token, { httpOnly: true, maxAge: maxAge * 1000 });
    res.status(200).json({ user: user._id });
  }
  catch (err) {
    const errors = handleErrors(err);
    res.status(400).json({ errors });
  }
}

```

Insert code



Go to <http://localhost:3000/login>

Master Chef Smoothies

Login

Email

mario1@gmail.com

Password

.....

LOGIN

Application

- Manifest Delete
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
 - http://localhost:3000
 - Trust Tokens

Cache

- Cache Storage
- Application Cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging

Frames

Application							
Filter							
Only show cookies with an issue							
Name	Value	Domain	Path	Expires / Ma...	Size	HttpOnly	Se
jwt	eyJhbGciOiJIUz...	localhost	/	2021-09-03T...	174	✓	
isEmployee	true	localhost	/	2021-08-31T...	14	✓	

Select a cookie to preview its value

Go to <http://localhost:3000/login>

Master Chef Smoothies

Login

Email

That email is not registered

Password

LOGIN

Master Chef Smoothies

Login

Email

Password

That password is incorrect

LOGIN