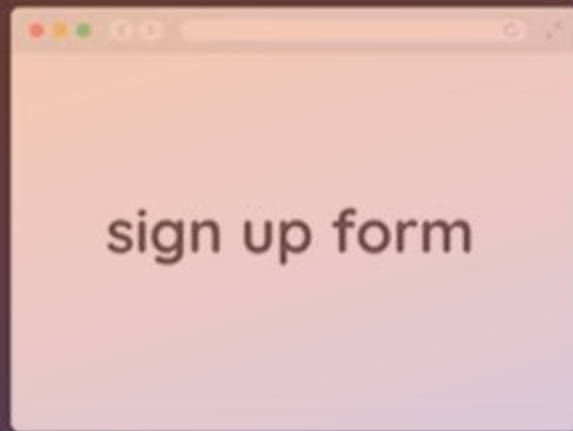


Node Auth Tutorial (JWT)

#11 - New User Signup



JSON Web Tokens

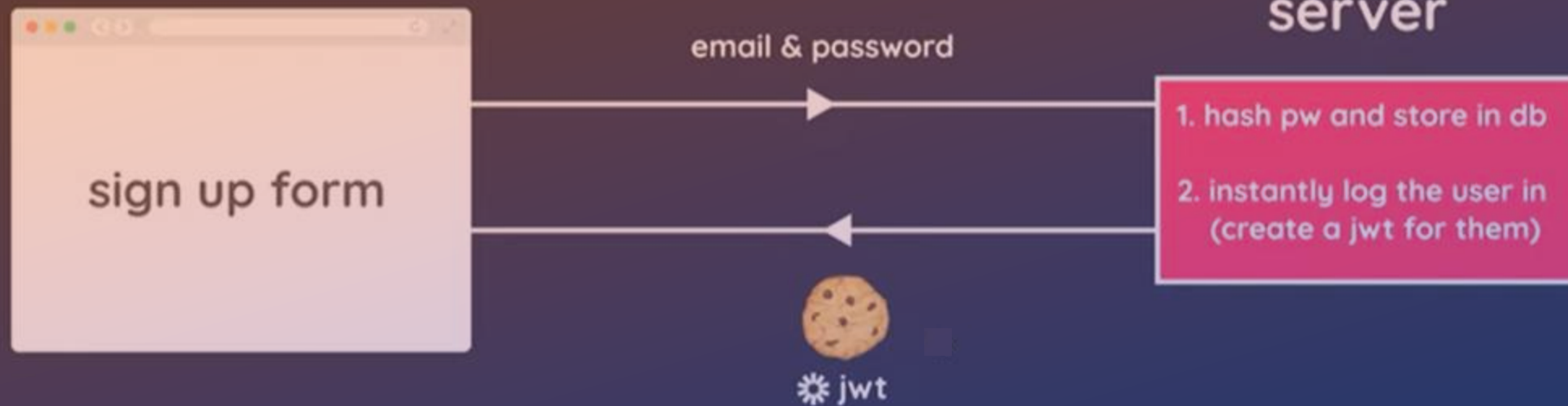


email & password

server

1. hash pw and store in db
2. instantly log the user in (create a jwt for them)

JSON Web Tokens



```
...  
const cookieParser = require('cookie-parser');  
  
const app = express();  
  
// middleware  
...  
app.use(cookieParser());  
  
// view engine  
...
```

Delete code

```
// cookies  
app.get('/set-cookies', (req, res) => {  
  res.cookie('newUser', false);  
  res.cookie('isEmployee', true, { maxAge: 1000 * 60 * 60 * 24, httpOnly: true });  
  res.send('you got the cookies!');  
});  
  
app.get('/read-cookies', (req, res) => {  
  const cookies = req.cookies;  
  console.log(cookies.newUser);  
  res.json(cookies);  
});
```

```
<%- include('partials/header'); -%>
```

```
...
```

```
<%- include('partials/footer'); -%>
```

```
<script>
```

```
const form = document.querySelector('form');
```

```
form.addEventListener('submit', async (e) => {
```

```
  e.preventDefault();
```

```
  // get values
```

```
  const email = form.email.value;
```

```
  const password = form.password.value;
```

```
  try {
```

```
    const res = await fetch('/signup', {
```

```
      method: 'POST',
```

```
      body: JSON.stringify({ email, password }),
```

```
      headers: { 'Content-Type': 'application/json' }
```

```
    });
```

```
  }
```

```
  catch (err) {
```

```
    console.log(err);
```

```
  }
```

```
});
```

```
</script>
```

Insert code



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

Master Chef Smoothies

Sign up

Email

mario@gmail.com

Password

SIGN UP



Copyright 2021 Master Chef Smoothies

- Terminal

```
npm install jsonwebtoken
```



```
const User = require("../models/User");
const jwt = require('jsonwebtoken');
// handle errors
...
// create json web token
const maxAge = 3 * 24 * 60 * 60;
const createToken = (id) => {
  return jwt.sign({ id }, 'secret key', {
    expiresIn: maxAge
  });
};
// controller actions
...
module.exports.signup_post = async (req, res) => {
  const { email, password } = req.body;

  try {
    const user = await User.create({ email, password });
    const token = createToken(user._id);
    res.cookie('jwt', token, { httpOnly: true, maxAge: maxAge * 1000 });
    res.status(201).json({ user: user._id });
  }
  catch(err) {
    const errors = handleErrors(err);
    res.status(400).json({ errors });
  }
}
...
```

Insert code



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

Master Chef Smoothies

Sign up

Email

yohsi@gmail.com

Password

.....

SIGN UP



Application

- Manifest
- 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 / ...	Size	HttpOnly	S...	S...	S...	P...
jwt	eyJhbGciOi...	localhost	/	2021-09-...	174	✓				M...
isEmployee	true	localhost	/	2021-08-...	14	✓				M...

Cookie Value ☐ Show URL decoded

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImRjYXN0ZXQpZm9udC51b250ODY0fQ.polkHmnZaKRz6zTL9Ubi9Zh2gl9Ueu7gn15XUeCVjVY

```
...  
<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();  
    // get values  
    const email = form.email.value;  
    const password = form.password.value;  
    try {  
      const res = await fetch('/signup', {  
        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('/');  
      }  
    }  
  })  
...  
</script>
```

Insert code



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

Master Chef Smoothies

Sign up

Email

luigi

Please enter a valid email

Password

...

Minimum password length is 6 characters

SIGN UP



Elements Console Sources Network Performance Memory Application >> 1 2

top -url:https://localhost:44323/js Default levels 2 Issues: 2

POST http://localhost:3000/signup 400 (Bad Request) signup?email=mario%4...&password=123321:42
signup?email=mario%4...&password=123321:48

```
{errors: {}}
  errors: {email: "Please enter a valid email", password: "Minimum password length is 6 characters"}
  [[Prototype]]: Object
```

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

Master Chef Smoothies

Sign up

Email

luigi@gmail.com

Password

...

Minimum password length is 6 characters

SIGN UP



Elements Console Sources Network Performance Memory Application >> 2 2

top -url:https://localhost:44323/js Default levels 2 Issues: 2

POST http://localhost:3000/signup 400 (Bad Request) signup?email=mario%4...&password=123321:42
signup?email=mario%4...&password=123321:48

{errors: {}}
errors: {email: "Please enter a valid email", password: "Minimum password length is 6 characters"}
[[Prototype]]: Object

POST http://localhost:3000/signup 400 (Bad Request) signup?email=mario%4...&password=123321:42
signup?email=mario%4...&password=123321:48

{errors: {}}
errors: {email: "", password: "Minimum password length is 6 characters"}
[[Prototype]]: Object

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

Master Chef Smoothies

Sign up

Email

mario@gmail.com

that email is already registered

Password

.....

SIGN UP

Elements Console Sources Network Application >> 1 2

top | -url:https://localhost:44 | Default levels | 2 Issues: 2

POST http://localhost:3000/signup 400 (Bad Request) signup?email=mario%40gmail.com&password=123321:42

signup?email=mario%40gmail.com&password=123321:48

{errors: {}} ⓘ

- errors: {email: "that email is already registered", password: ""}
- [[Prototype]]: Object