

Authentication for React Apps

@AshishDasnurkar

I work for Auth0

What you will learn today?

- React Basics
- Authenticating users with email/pwd and social logins

What is React?

- A JavaScript library for building user interfaces
- Promotes Component-Based Architecture

React Component

- Essentially React components are JS functions that take data in and return *React elements* describing what should appear on screen

```
class GreetingComponent extends React.Component {  
  render() {  
    return <h3> Yo { this.props.name }, how are you doing?</h3>;  
  }  
}
```

JSX

- Syntax extension to JavaScript
- it's neither string nor HTML

Component lifecycle methods (.. and other important methods)

- lifecycle hooks are
 - `componentWillMount()`
 - **`componentDidMount()`**
 - `componentWillUnmount()`
 - `shouldComponentUpdate()`
 - `componentWillUpdate()`
 - **`render`** and
 - **`setState`**

Props

- They are component attributes
- arbitrary inputs given to component

```
class QueenBorg extends React.Component {  
  render() {  
    return <Drone designation="seven of nine" grid="Unimatrix Zero"/>;  
  }  
}
```

```
class Drone extends React.Component {  
  render() {  
    return <p> We are the Borg. Designation { this.props.designation } Grid {this.props.grid}. </p>;  
  }  
}
```

State

- Internal to the component
- Not necessarily tied with how the component is displayed

```
class Button extends React.Component {  
  constructor() {  
    super();  
    this.state = {  
      count: 0,  
    };  
  }  
  
  updateCount() {  
    this.setState((prevState, props) => {  
      return { count: prevState.count + 1 }  
    });  
  }  
  
  render() {  
    return (  
      <button onClick={() => this.updateCount()} >  
        Clicked {this.state.count} times  
      </button>  
    );  
  }  
}
```


OK ! Enough about React, let's talk Authentication

Browser

Server

1. POST /users/login with username and password

2. Creates a User session

3. Returns a logged in cookie to the browser

4. Do an authenticated request. Sends the cookie.

5. Check the session based on the cookie and authenticate the user

6. Sends response to the client

That used to work well

But

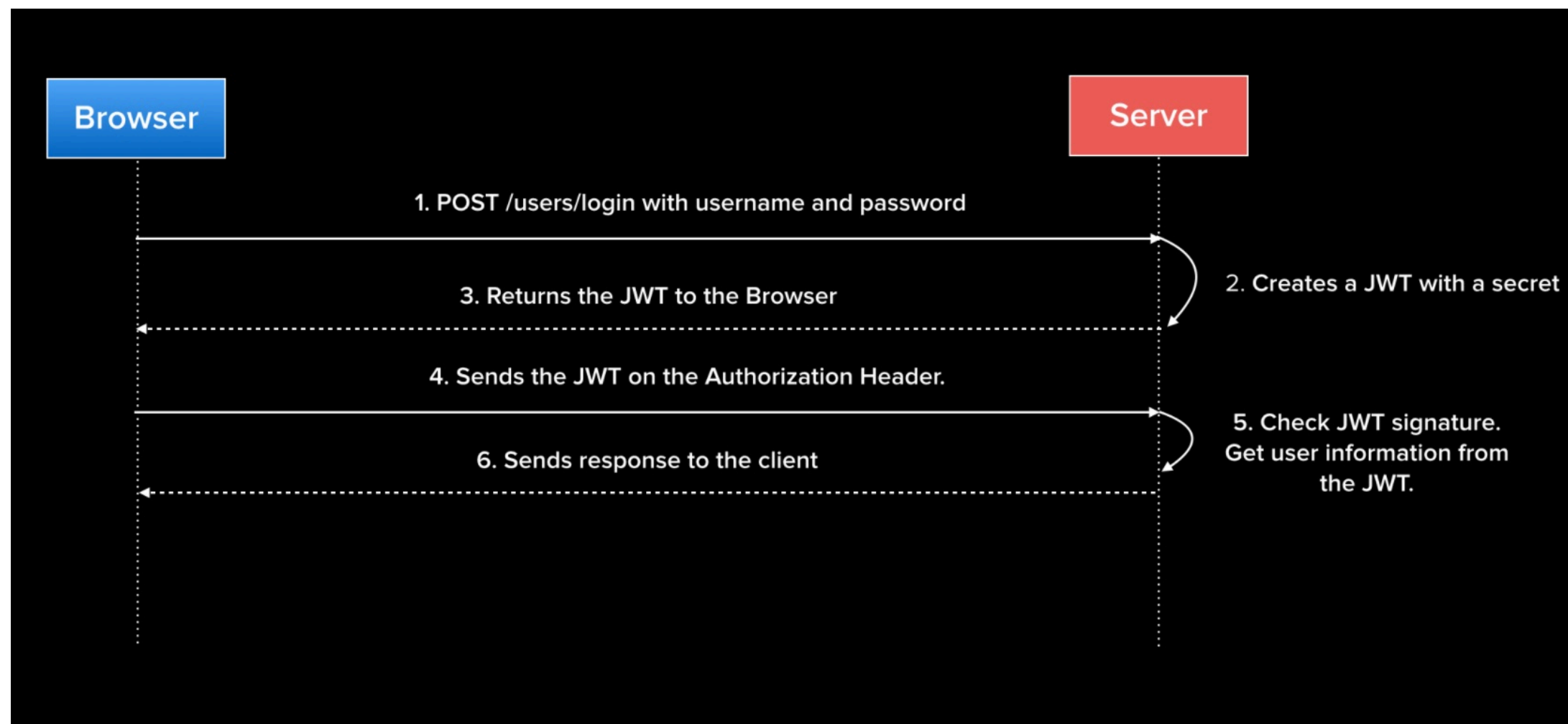
- Cookies don't work well with CORS
- Sessions require stateful servers
- APIs should be stateless

JWTs to the rescue

- JSON Web Token
- Standard
- Secure
- Self Contained
- JSON 🙌🙌

Token Based Authentication

— Open ID Connect (OIDC)



JWT example

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.EkN-D0snsuRjR06BxXemmJDm3HbxrbRzXglbN2S4s0kopdU4IsDxTI8j019W_A4K8ZPJijNLis4EZsHeY559a4DF0d50_OqgHGuERTqYZyuhtF39yxJPAjUESwxk2J5k_4zM30-vtd1Ghyo4IbqKKSy6J9mTniYJPenn5-HIirE
```

Decoded

EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "RS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "sub": "1234567890",  "name": "John Doe",  "admin": true}
```

VERIFY SIGNATURE

```
RSASHA256(  base64UrlEncode(header) + "." +  base64UrlEncode(payload),  -----BEGIN PUBLIC KEY-----  MIGfMA0GCSqGSIB3DQEBAQUAA4GNA  DCBiQKBgQDdlatRjRjogo3WojgGHF  HYLugdUWAY9iR3fy4arWNA1KoS8kV  ,  -----BEGIN RSA PRIVATE KEY---  --  MIICWwIBAAKBgQDdlatRjRjogo3Wo  jgGHFHYLugdUWAY9iR3fy4arWNA1K
```

Demo and Code Walkthrough
