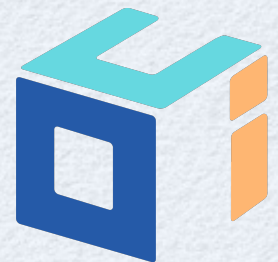


React in Angular

slides at <https://github.com/mvolkmann/talks>

R. Mark Volkmann
Object Computing, Inc.
<http://objectcomputing.com>
Email: mark@objectcomputing.com
Twitter: @mark_volkman
GitHub: mvolkmann



OCI | TRAINING

Overview

- React components can be used in Angular apps!
- Even in ones created with the Angular CLI
 - `npm install -g @angular/cli`
 - `ng new my-app`
 - `cd my-app`
 - `ng serve`
 - browse `localhost:4200`

npm Installs

- Core React library
 - `npm install react`
- React library for DOM rendering
 - `npm install react-dom`
- React library for specifying “prop types”
 - `npm install prop-types`
 - could also use Flow or TypeScript
- Babel for transpiling React code containing JSX
 - `npm install -D babel-cli`
- Babel preset plugin
 - `npm install -D babel-preset-env`
 - “automatically determines the Babel plugins you need based on your supported environments”
- Babel JSX plugin
 - `npm install -D babel-plugin-transform-react-jsx`

Babel Setup

- Create `.babelrc` file at project root

```
{  
  "presets": [  
    ["env", {  
      "targets": {  
        "browsers": ["last 2 versions"]  
      }  
    }]  
  ],  
  "plugins": [  
    "transform-react-jsx"  
  ]  
}
```

`.babelrc`

- Add npm script to run `babel` on all React component source files

```
"react-build": "babel 'src/react/**/*.jsx' -d .",
```

- only runs on `.jsx` files under `src/react`
- creates `.js` files in same directory that can be imported into Angular source files
 - don't check these into version control

Example React Component

- `src/react/hello.jsx`

```
import * as React from 'react';
import PropTypes from 'prop-types';

export class Hello extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}!</h1>;
  }
}

Hello.propTypes = {
  name: PropTypes.string.isRequired
};
```

src/react/react-util.js ...

```
import * as React from 'react';
import ReactDOM from 'react-dom';

/**
 * This creates a custom element that uses a
 * Web Component to wrap a given a React component
 * @param {Object} customElementName the name of the custom element to be created
 * @param {Object} reactComponent the React component to be rendered
 * @param {string[]} propNames an array of prop names used by the React component
 */
export function createCustomElement(
  customElementName,
  reactComponent,
  propNames
) {
  class ReactWebComponent extends HTMLElement {
    constructor() {
      super();
      const shadow = this.attachShadow({mode: 'open'});
      this.mountPoint = document.createElement('span');
      shadow.appendChild(this.mountPoint);
      this.props = {};
    }
  }
```


... src/react/react-util.js

```
static get observedAttributes() {  
  return propNames;  
}  
  
attributeChangedCallback(name, oldValue, newValue) {  
  this.props[name] = newValue;  
  this.render();  
}  
  
connectedCallback() {  
  Array.from(this.attributes).forEach(  
    a => (this.props[a.name] = a.value)  
  );  
  this.render();  
}  
  
render() {  
  const reactElement = React.createElement(reactComponent, this.props);  
  ReactDOM.render(reactElement, this.mountPoint);  
}  
}  
  
customElements.define(customElementName, ReactWebComponent);  
}
```

Web Components ...

- Can wrap a React component in a Web Component that become custom elements
- To enable use of custom elements in an Angular app, add the highlighted parts below in `src/app/app.module.ts`

```
import {BrowserModule} from '@angular/platform-browser';
import {NgModule, CUSTOM_ELEMENTS_SCHEMA} from '@angular/core';

import {AppComponent} from './app.component';

@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule],
  providers: [],
  bootstrap: [AppComponent],
  schemas: [CUSTOM_ELEMENTS_SCHEMA]
})
export class AppModule {}
```


... Web Components

- To create custom elements that use Web Components to render a React components

- create `src/react/index.js`

```
import {createCustomElement} from './react-util';
import {Hello} from './hello';
// Can import more React components.

createCustomElement('x-hello', Hello, ['name']);
// Can create more custom elements.
```

- import this in `src/app/app.module.ts` to make the custom elements available

```
import './react';
```

- To use the custom elements, add lines like the following in some `.html` file where they should be rendered

```
<x-hello name="Tami"></x-hello>
```

Steps to Run

- **npm run react-build**
 - to compile all the React .jsx files to .js files
- **npm start**
 - to start a local Angular web server
- **browse localhost:4200**