

Day 12: Web user interfaces with React

CODE BOOTCAMP VAASA 2021
JANI JÄRVINEN, INTERTECHNO TRAINING OY



The React user interface library

What is React?

- React is a JavaScript based user interface framework for building web user interfaces (UIs)
- Originally from Facebook/Instagram (aka Meta), today React is open-source
 - Facebook started using React in its timeline feature in 2012, React was open-sourced in 2013
- React focuses on the user interface, and thus implements the “View” in the MVC model
 - React does not offer a complete front-end library like for example Angular from Google
- In React, application are written based on components, which can be used multiple times
- In turn, complete user interfaces are generated from nested components
- For more information
 - <https://reactjs.org/>

A Hello Word page in React

```
<html>
<div id="myReactApp"></div>
<script type="text/jsx">
  class Greeter extends React.Component {
    render() {
      return <h1>{this.props.greeting}</h1>
    }
  }
  ReactDOM.render(<Greeter greeting="Hello World!" />, document.getElementById("myReactApp"));
</script>
```

The base template for React applications

- The “Create-React-App” package in NPM
- Installation
 1. `npx create-react-app my-app`
 2. `cd my-app`
 3. `npm start`
- These commands will create a complete React application that you can start developing
 - Includes React, JSX, ES6 and live development (using Webpackin)
- For details
 - <https://github.com/facebook/create-react-app>

Node Package Manager

- Node.js's package management system is called NPM
- NPM can download packages from the Internet and install them and their dependencies
- Everyone can create their own packages and upload them to the NPM global registry
- Compare NPM with Visual Studio's NuGet package manager, both do similar things
- Start with these
 - The command "npm install"
 - The package configuration file "package.json"
- For more details
 - <https://docs.npmjs.com/>

An alternative: Yarn

- NPM is the most well known JavaScript package management tool, but Facebook's alternative called Yarn is another common option
- Yarn attempts to fix NPM's missing pieces
 - Improve the reliability
 - Better performance
 - Offline installation possibility
- Yarn ja NPM are partially compatible
 - The basic commands are the same, and they both use the same settings file "package.json"
- For more information
 - <https://yarnpkg.com/lang/en/>

React major features

- Virtual DOM
 - React will generate an internal tree model that keeps track of the user interface
 - Allows fast user interface updates on the screen
- One-way data flow
 - React limits the ability of components to communicate so that component properties are immutable
 - Phrase to remember: "properties flow down; actions flow up"
- JSX syntax
 - JSX is JavaScript + XML: it combines JavaScript code and XML to form HTML output
 - JSX not a mandatory part of Reach, but a highly useful one and thus widely used
- React usually works in the browser, but can also be rendered on the server
 - This helps the performance of complex applications
 - Backend code can support this server-side rendering (SSR)

Often-used tools

- NPM and NPX

- NPM = Node Package Manager, allows working with React
- NPX is a new NPM extension that allows more functionality than NPM

- Babel

- A JavaScript compiler that produces browser-compatible code from latest JavaScript language versions
- Supports compiling JSX files
- <https://babeljs.io/>

- Webpack

- Allows creating final, executable React applications along with Babel, especially JSX and TSX files
- Allows minimizing and bundling for files (optimizations)
- <https://webpack.js.org/>

Component functionality

- Each React component must implement at minimum the *render* method, but there are other useful methods as well
- Handling the *lifecycle* of a component
 - `componentWillMount`
 - `componentDidMount`
 - `getInitialState`
- Updating the state
 - `setState`
- React components can be defined either as functions or classes
 - Classes are a modern functionality of the JavaScript language
 - Classes can also be used with TypeScript

Updating the state

- In React, the component properties (*props*) are non-changeable (immutable)
- On the other hand, *state* can change during the lifetime of the component
- Updating the state is done using the `setState` method also automatically updates the user interface
 - React only updates the screen for those parts of the component HTML that has actually changed



Interactivity in React

- Dynamic user interfaces

- For example, table components can render an externally-controller number of rows
- Collections are rendered in React using curly braces: { }

- Events

- Allows React components to react to events, such as onClick or onKeyDown
- Compare this to event handling for example using jQuery

- Forms

- Added form input fields in React is done in a different manner than in HTML, because the content of the input field must be stored in the state
- Normally, React components manage their own state; in input fields, it is the user

Libraries related to React

■ Facebook Flux

- An architecture and a library that helps in developing complex React applications
- Provides a model to update the application's internal state only through events
- <https://facebook.github.io/flux/>

■ Redux

- A library inspired by Flux, but simpler to use
- <https://redux.js.org/>

■ React Native

- A framework for building mobile applications for Android and iOS, based on React
- <https://reactnative.dev/>

Practice: your own frontend to the demo

1. Use the "Create-React-App" command to create yourself a React application
2. Add the Fetch library to your project, and use it to call some backend system
3. Investigate the event model of React and find out in which order the following methods are called:
 - The componentDidMount method
 - The constructor of the React component
 - The render method
4. Connect your ASP.NET backend to your React application using the Web API
5. Implement new React components and test their nesting (component inside another)

TypeScript

The TypeScript language

- A modern JavaScript related language originally from Microsoft
- TypeScript allows using types in variable declarations, hence the name
- Shortly "TS", the compiler is named "TSC" (compare to "CSC" which is the C# compiler)
- TypeScript is a superset of the JavaScript language
 - Almost everything in JavaScript is also automatically TypeScript
- The TypeScript compiler will generate JavaScript code that can run in the browser, for example
 - The official term is a "transpiler" not a "compiler" because binary code is not generated, just another form of source code
- Details
 - <http://www.typescriptlang.org/>

TypeScript typing

- TypeScript supports many of the same data types that for example C# supports
 - However, TypeScript also brings many types that are not present in JavaScript itself, such as enums
- Important to remember: both JavaScript and TypeScript use what is called *duck typing*: if two objects have the same properties, they are compatible with each other
- Basic types
 - Boolean, Number, String, Array, Enum
 - Any, Void, Null, Undefined
 - Object
- More exotic types
 - Tuple
 - Never