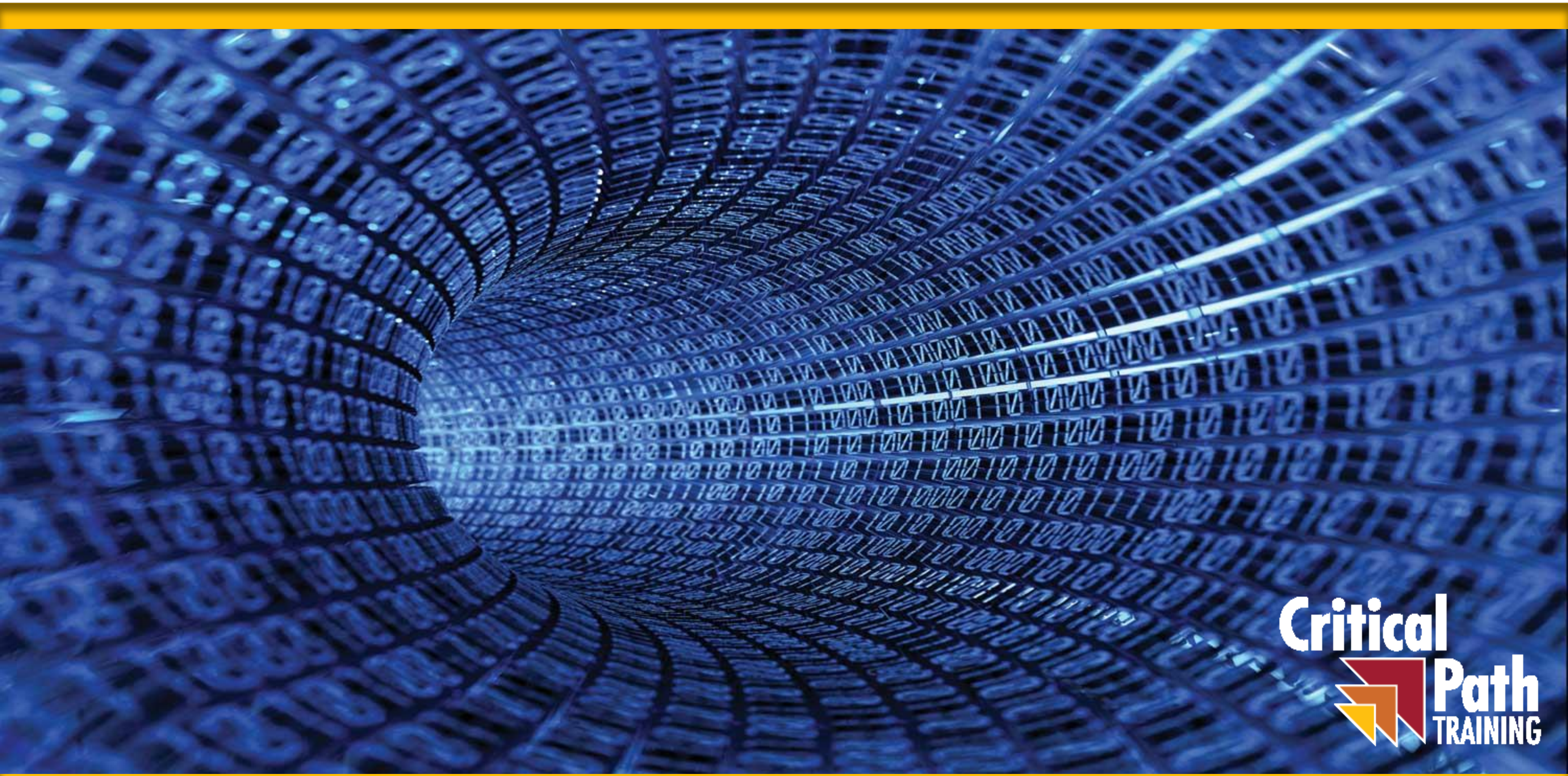
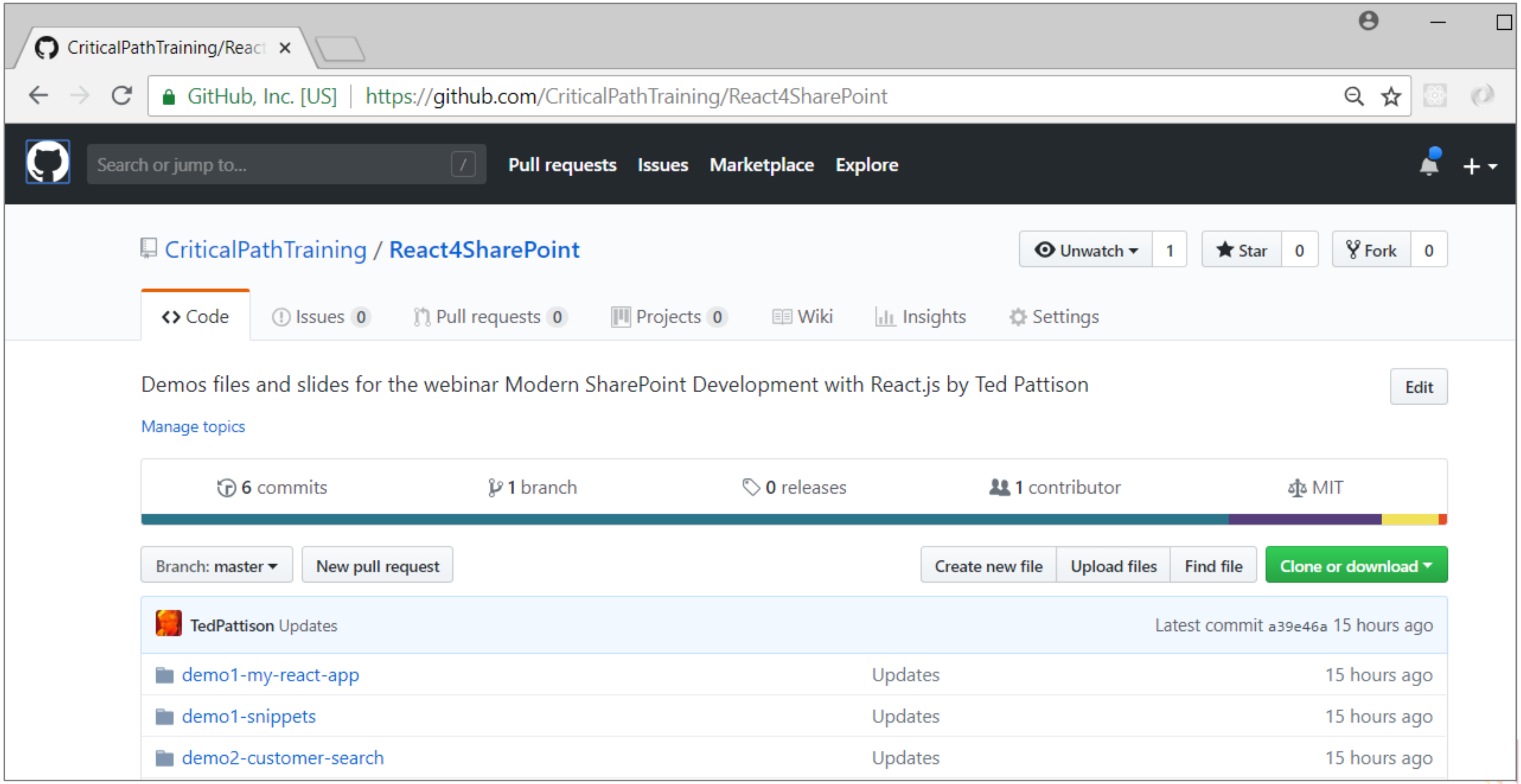
Modern SharePoint Development with React.js



GitHub Repo

- All sample code and slides available for download
  - https://github.com/CriticalPathTraining/React4SharePoint



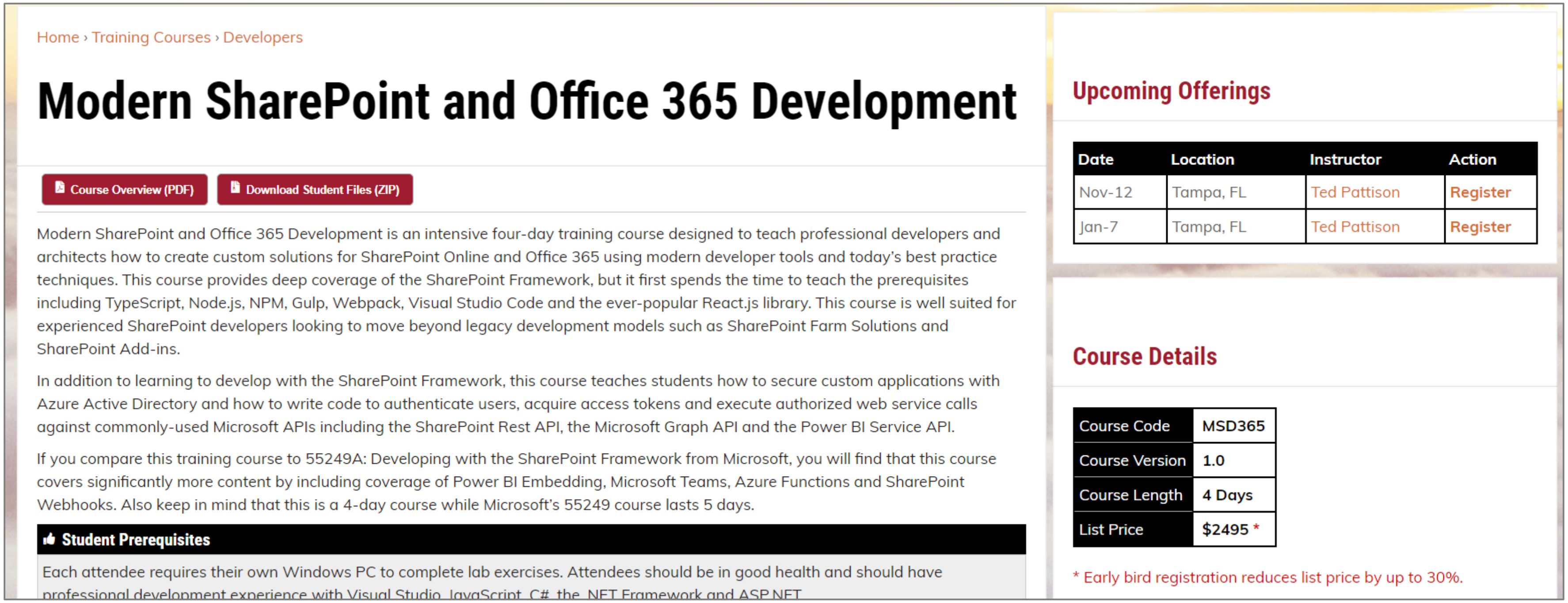
Demos files and slides for the webinar Modern SharePoint Development with React.js by Ted Pattison

TedPattison Updates — Latest commit a39e46a 15 hours ago

demo1-my-react-app	Updates	15 hours ago
demo1-snippets	Updates	15 hours ago
demo2-customer-search	Updates	15 hours ago

Effective SharePoint Framework Training

- MSD365: Modern SharePoint and Office 365 Development
  - 4-day of training with lots of hands-on labs
  - Learn how to develop with SPFx the right way right from the start
  - https://www.criticalpathtraining.com/courses/sharepoint/modern-sharepoint-office-365-development/



Home › Training Courses › Developers

Modern SharePoint and Office 365 Development

Modern SharePoint and Office 365 Development is an intensive four-day training course designed to teach professional developers and architects how to create custom solutions for SharePoint Online and Office 365 using modern developer tools and today's best practice techniques. This course provides deep coverage of the SharePoint Framework, but it first spends the time to teach the prerequisites including TypeScript, Node.js, NPM, Gulp, Webpack, Visual Studio Code and the ever-popular React.js library. This course is well suited for experienced SharePoint developers looking to move beyond legacy development models such as SharePoint Farm Solutions and SharePoint Add-ins.

In addition to learning to develop with the SharePoint Framework, this course teaches students how to secure custom applications with Azure Active Directory and how to write code to authenticate users, acquire access tokens and execute authorized web service calls against commonly-used Microsoft APIs including the SharePoint Rest API, the Microsoft Graph API and the Power BI Service API.

If you compare this training course to 55249A: Developing with the SharePoint Framework from Microsoft, you will find that this course covers significantly more content by including coverage of Power BI Embedding, Microsoft Teams, Azure Functions and SharePoint Webhooks. Also keep in mind that this is a 4-day course while Microsoft's 55249 course lasts 5 days.

Student Prerequisites

Each attendee requires their own Windows PC to complete lab exercises. Attendees should be in good health and should have professional development experience with Visual Studio, JavaScript, C#, the .NET Framework and ASP.NET

Upcoming Offerings

Date	Location	Instructor	Action
Nov-12	Tampa, FL	Ted Pattison	Register
Jan-7	Tampa, FL	Ted Pattison	Register

Course Details

Course Code	MSD365
Course Version	1.0
Course Length	4 Days
List Price	$2495 *

* Early bird registration reduces list price by up to 30%.

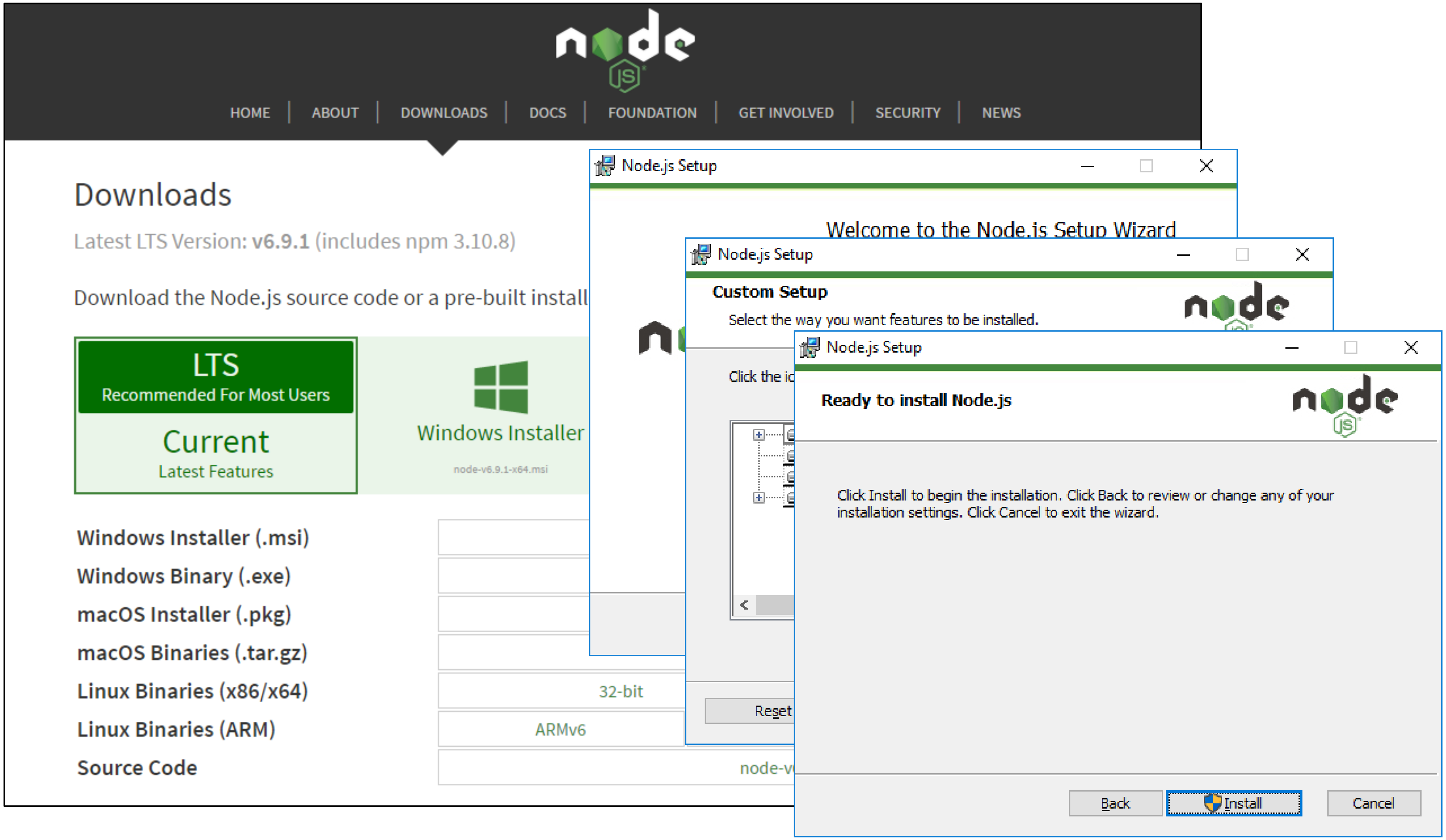
Agenda

➢ Developing with Node.js, TypeScript and Webpack

- Learning React.js Fundamentals
- Using the Office UI Fabric React Component Library
- Developing React Webparts with SharePoint Framework
- Calling the Microsoft Graph API from React Webparts

Installing node.js

- https://nodejs.org/en/download/



Downloads

Latest LTS Version: **v6.9.1** (includes npm 3.10.8)

Download the Node.js source code or a pre-built install

LTS
Recommended For Most Users

Current
Latest Features

Windows Installer
node-v6.9.1-x64.msi

Windows Installer (.msi)
Windows Binary (.exe)
macOS Installer (.pkg)
macOS Binaries (.tar.gz)
Linux Binaries (x86/x64) 32-bit
Linux Binaries (ARM) ARMv6
Source Code node-v

Node.js Setup

Welcome to the Node.js Setup Wizard

Node.js Setup

Custom Setup

Select the way you want features to be installed.

Click the ic

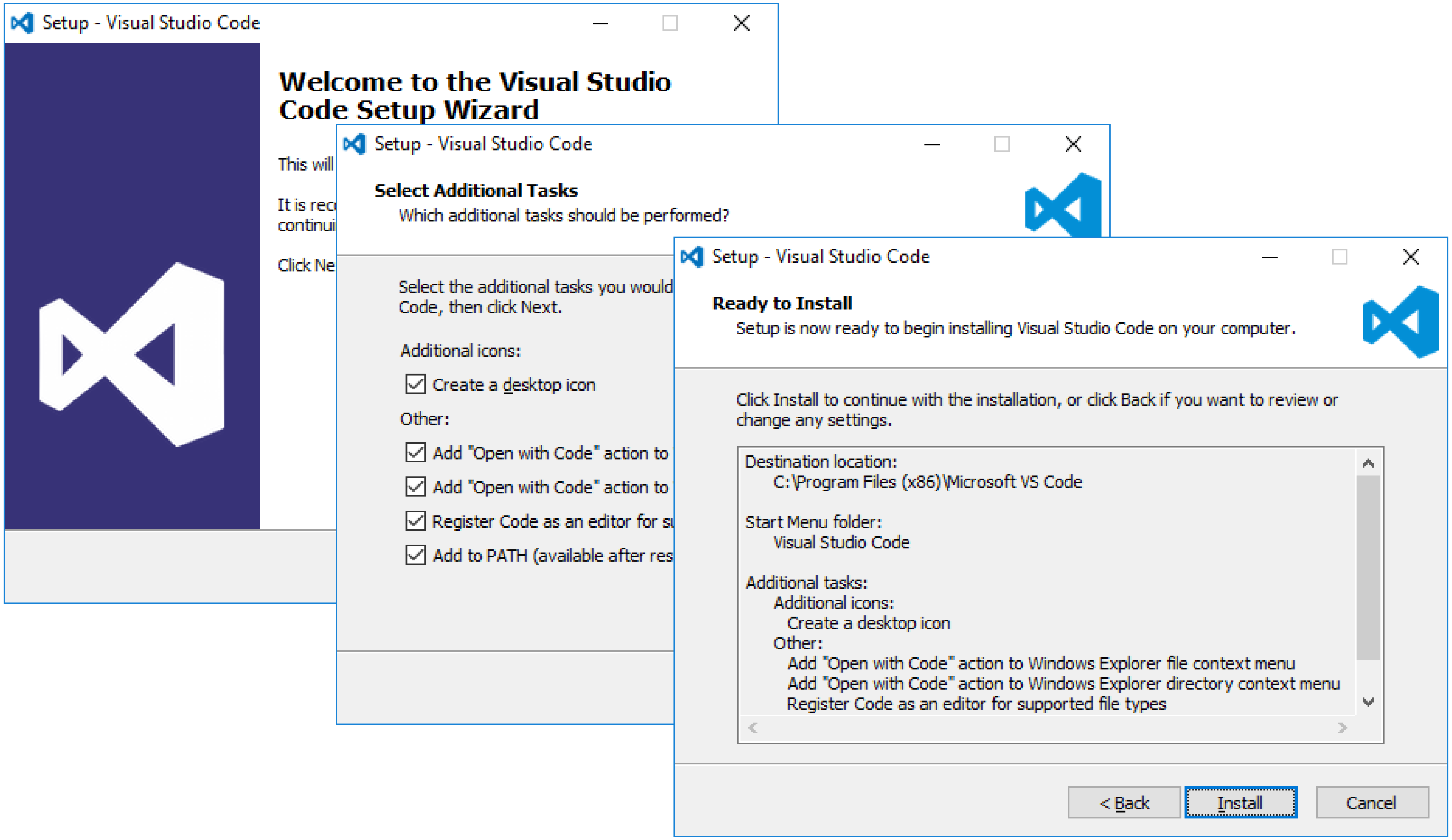
Node.js Setup

Ready to install Node.js

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Install Visual Studio Code

- http://code.visualstudio.com/



Welcome to the Visual Studio Code Setup Wizard

This will

It is rec
continui

Click Ne

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would
Code, then click Next.

Additional icons:

☑ Create a desktop icon

Other:

☑ Add "Open with Code" action to

☑ Add "Open with Code" action to

☑ Register Code as an editor for su

☑ Add to PATH (available after res

Ready to Install

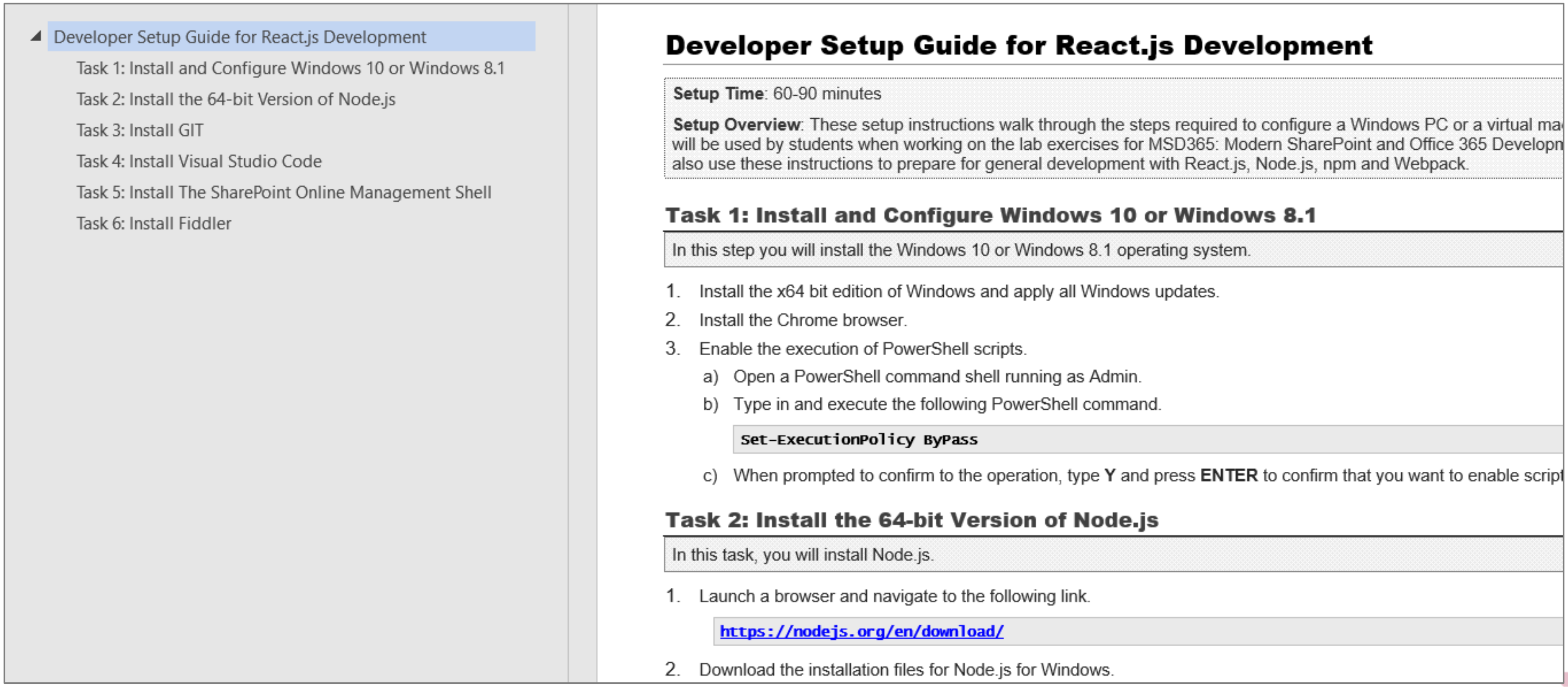
Setup is now ready to begin installing Visual Studio Code on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Program Files (x86)\Microsoft VS Code

Start Menu folder:
    Visual Studio Code

Additional tasks:
    Additional icons:
      Create a desktop icon
    Other:
      Add "Open with Code" action to Windows Explorer file context menu
      Add "Open with Code" action to Windows Explorer directory context menu
      Register Code as an editor for supported file types
```

Modern React Developer Setup Guide

- Use the Critical Path setup guide to get started
  - Used to configure Windows PC for react development
  - https://github.com/CriticalPathTraining/React4SharePoint/raw/master/Setup.pdf



◢ Developer Setup Guide for React.js Development

- Task 1: Install and Configure Windows 10 or Windows 8.1
- Task 2: Install the 64-bit Version of Node.js
- Task 3: Install GIT
- Task 4: Install Visual Studio Code
- Task 5: Install The SharePoint Online Management Shell
- Task 6: Install Fiddler

Developer Setup Guide for React.js Development

Setup Time: 60-90 minutes

Setup Overview: These setup instructions walk through the steps required to configure a Windows PC or a virtual ma will be used by students when working on the lab exercises for MSD365: Modern SharePoint and Office 365 Developn also use these instructions to prepare for general development with React.js, Node.js, npm and Webpack.

Task 1: Install and Configure Windows 10 or Windows 8.1

In this step you will install the Windows 10 or Windows 8.1 operating system.

1. Install the x64 bit edition of Windows and apply all Windows updates.
2. Install the Chrome browser.
3. Enable the execution of PowerShell scripts.
   a) Open a PowerShell command shell running as Admin.
   b) Type in and execute the following PowerShell command.

```
Set-ExecutionPolicy ByPass
```

   c) When prompted to confirm to the operation, type **Y** and press **ENTER** to confirm that you want to enable script

Task 2: Install the 64-bit Version of Node.js

In this task, you will install Node.js.

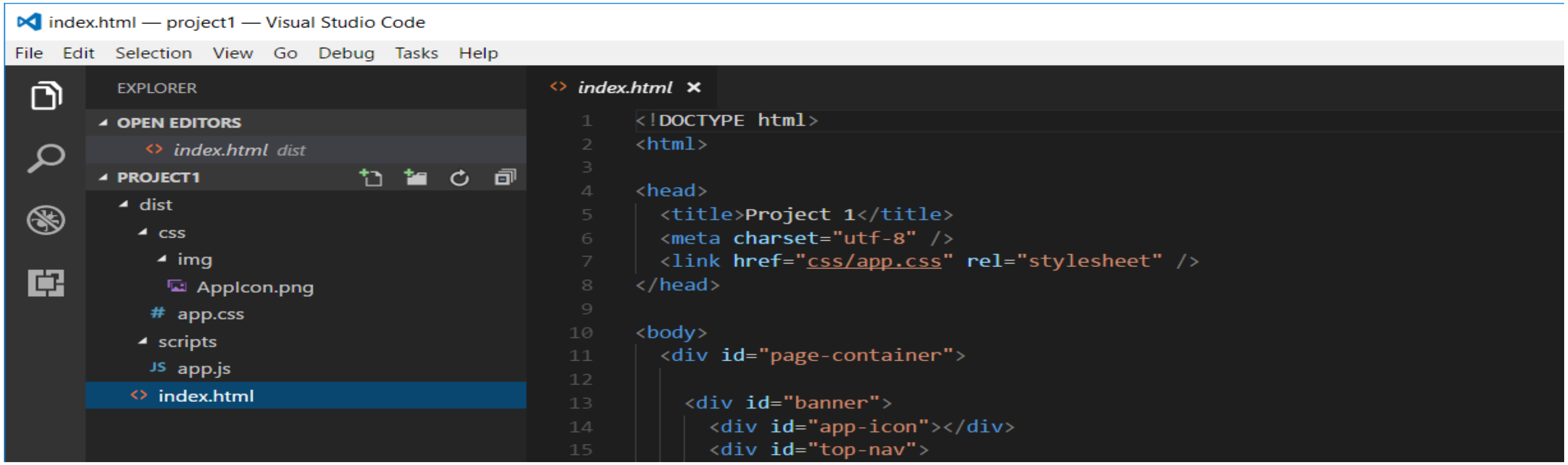
1. Launch a browser and navigate to the following link.

```
https://nodejs.org/en/download/
```

2. Download the installation files for Node.js for Windows.

Developing with Visual Studio Code

- Node.js is agnostic when it comes to developer IDE
  - There are many different IDEs that people use with Node.js
  - This course will be using Visual Studio Code

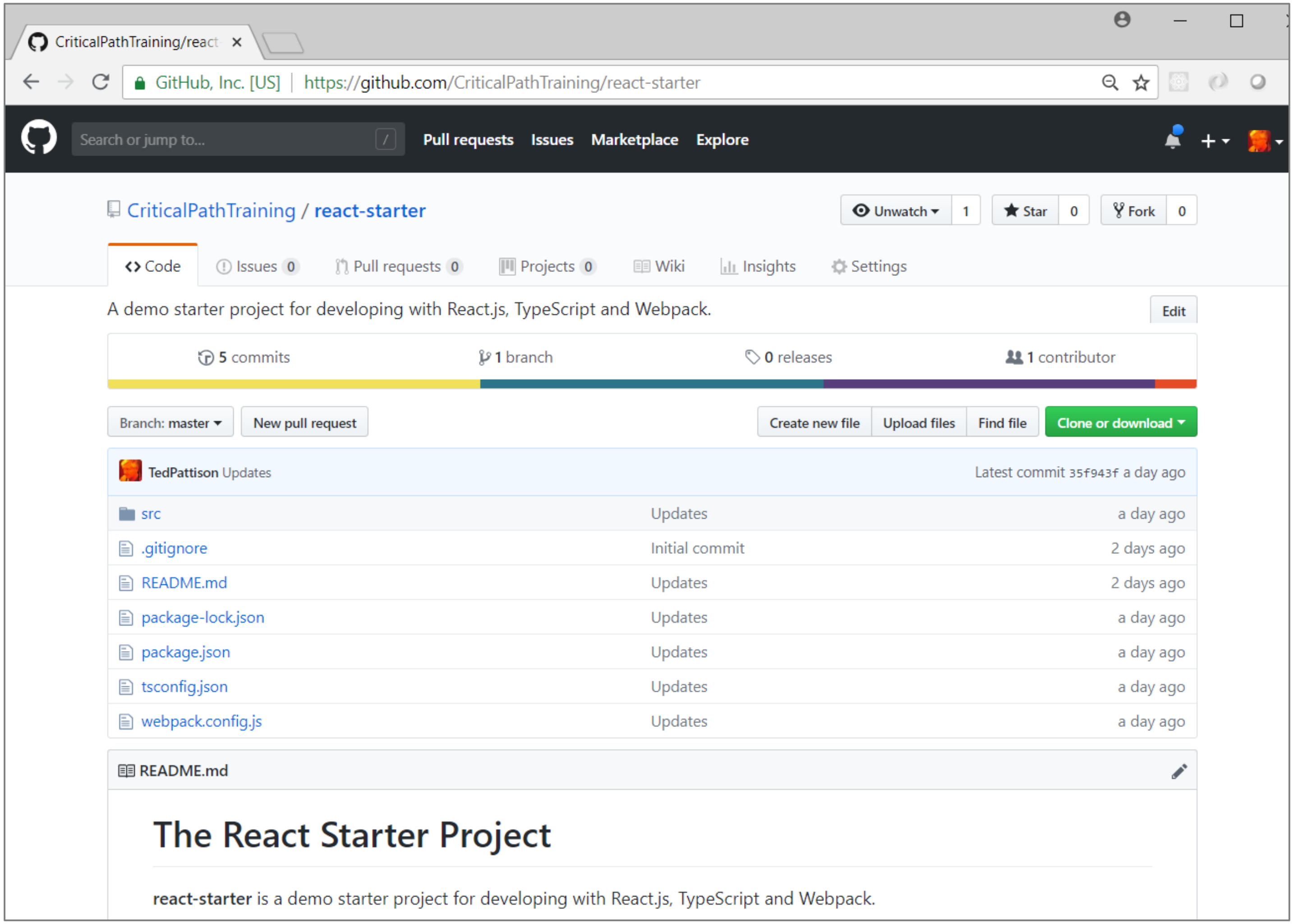


```
<!DOCTYPE html>
<html>

<head>
  <title>Project 1</title>
  <meta charset="utf-8" />
  <link href="css/app.css" rel="stylesheet" />
</head>

<body>
  <div id="page-container">

    <div id="banner">
      <div id="app-icon"></div>
      <div id="top-nav">
```

- Visual Studio is not a good fit for Node.js development
  - Visual Studio solution & project files incompatible with Node.js

The React Starter Project

https://github.com/CriticalPathTraining/react-starter



A demo starter project for developing with React.js, TypeScript and Webpack.

TedPattison Updates — Latest commit 35f943f a day ago

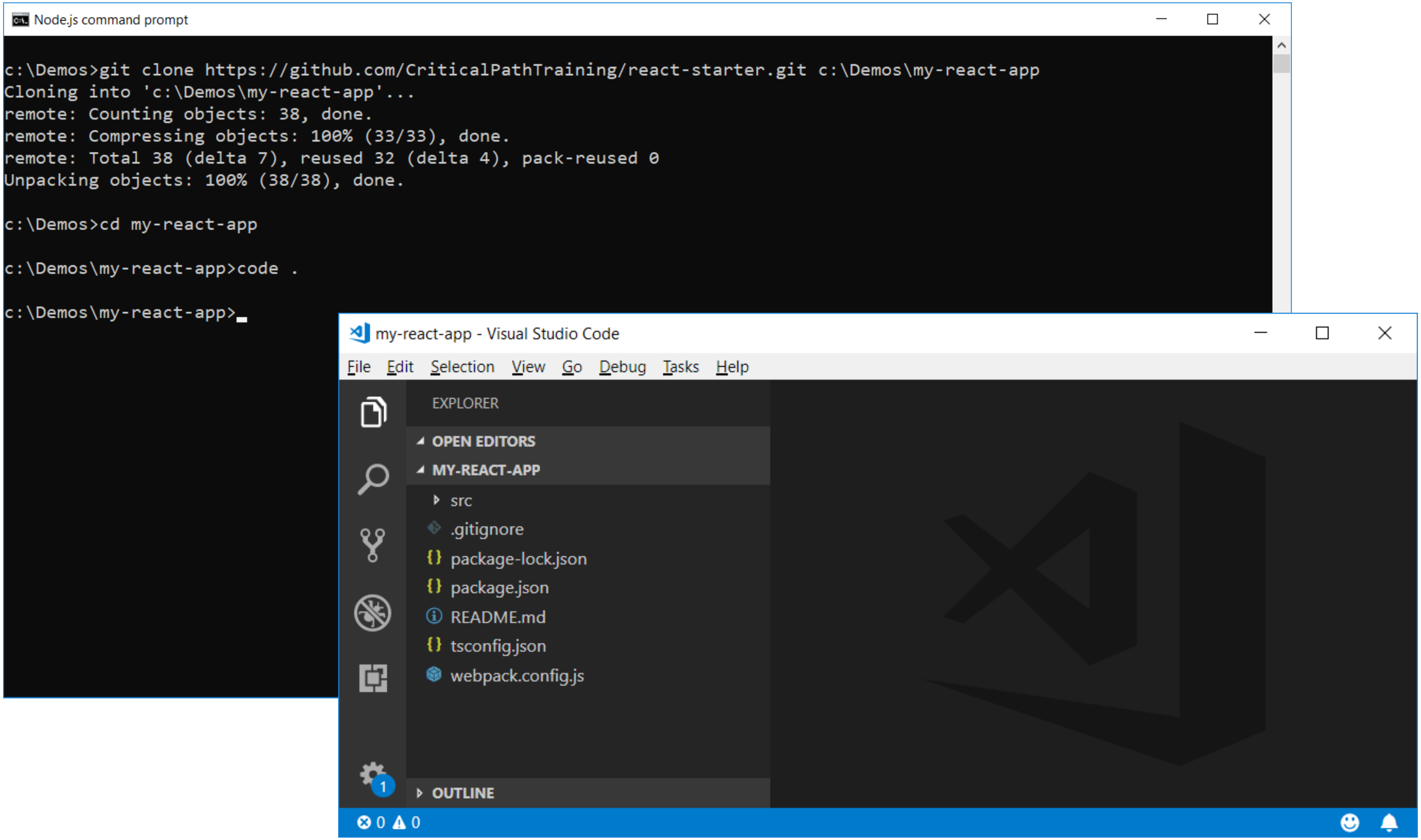
src	Updates	a day ago
.gitignore	Initial commit	2 days ago
README.md	Updates	2 days ago
package-lock.json	Updates	a day ago
package.json	Updates	a day ago
tsconfig.json	Updates	a day ago
webpack.config.js	Updates	a day ago

README.md

The React Starter Project

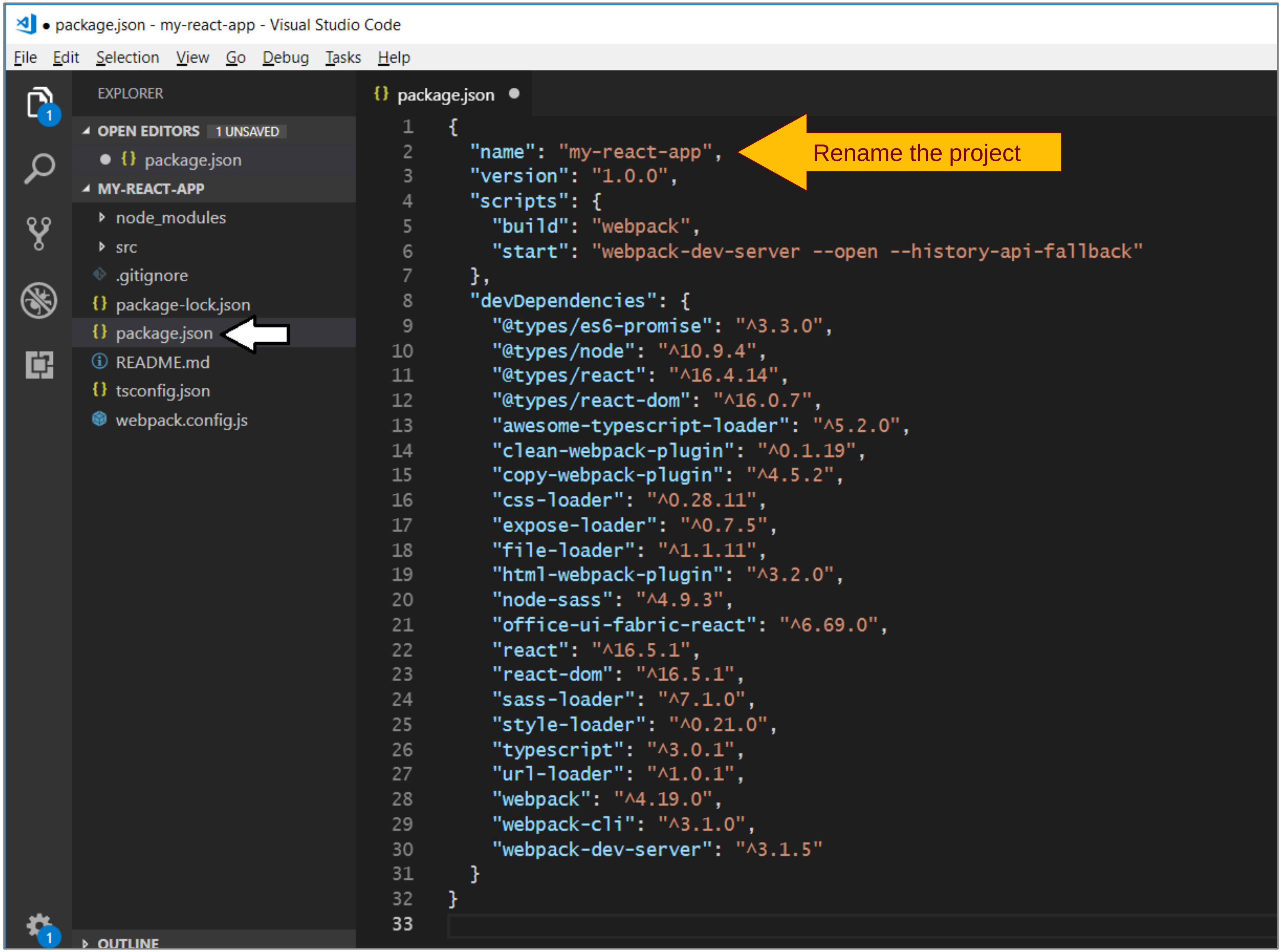
react-starter is a demo starter project for developing with React.js, TypeScript and Webpack.

Cloning the Starter Project



```
c:\Demos>git clone https://github.com/CriticalPathTraining/react-starter.git c:\Demos\my-react-app
Cloning into 'c:\Demos\my-react-app'...
remote: Counting objects: 38, done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 38 (delta 7), reused 32 (delta 4), pack-reused 0
Unpacking objects: 100% (38/38), done.

c:\Demos>cd my-react-app

c:\Demos\my-react-app>code .

c:\Demos\my-react-app>_
```

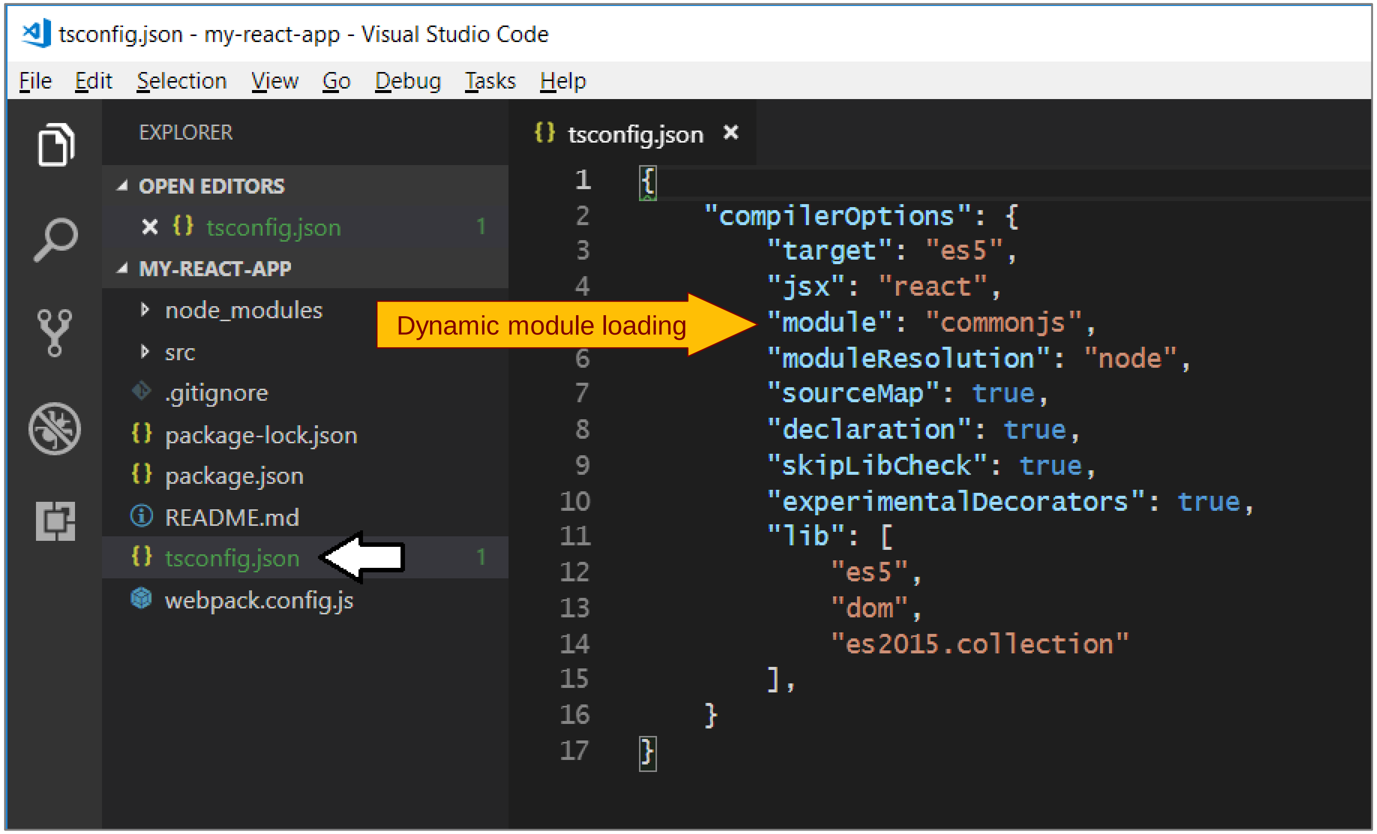
Starter Project - package.json



```json
{
  "name": "my-react-app",
  "version": "1.0.0",
  "scripts": {
    "build": "webpack",
    "start": "webpack-dev-server --open --history-api-fallback"
  },
  "devDependencies": {
    "@types/es6-promise": "^3.3.0",
    "@types/node": "^10.9.4",
    "@types/react": "^16.4.14",
    "@types/react-dom": "^16.0.7",
    "awesome-typescript-loader": "^5.2.0",
    "clean-webpack-plugin": "^0.1.19",
    "copy-webpack-plugin": "^4.5.2",
    "css-loader": "^0.28.11",
    "expose-loader": "^0.7.5",
    "file-loader": "^1.1.11",
    "html-webpack-plugin": "^3.2.0",
    "node-sass": "^4.9.3",
    "office-ui-fabric-react": "^6.69.0",
    "react": "^16.5.1",
    "react-dom": "^16.5.1",
    "sass-loader": "^7.1.0",
    "style-loader": "^0.21.0",
    "typescript": "^3.0.1",
    "url-loader": "^1.0.1",
    "webpack": "^4.19.0",
    "webpack-cli": "^3.1.0",
    "webpack-dev-server": "^3.1.5"
  }
}
```

Rename the project

Starter Project tsconfig.json



Dynamic module loading

```
{
    "compilerOptions": {
        "target": "es5",
        "jsx": "react",
        "module": "commonjs",
        "moduleResolution": "node",
        "sourceMap": true,
        "declaration": true,
        "skipLibCheck": true,
        "experimentalDecorators": true,
        "lib": [
            "es5",
            "dom",
            "es2015.collection"
        ],
    }
}
```

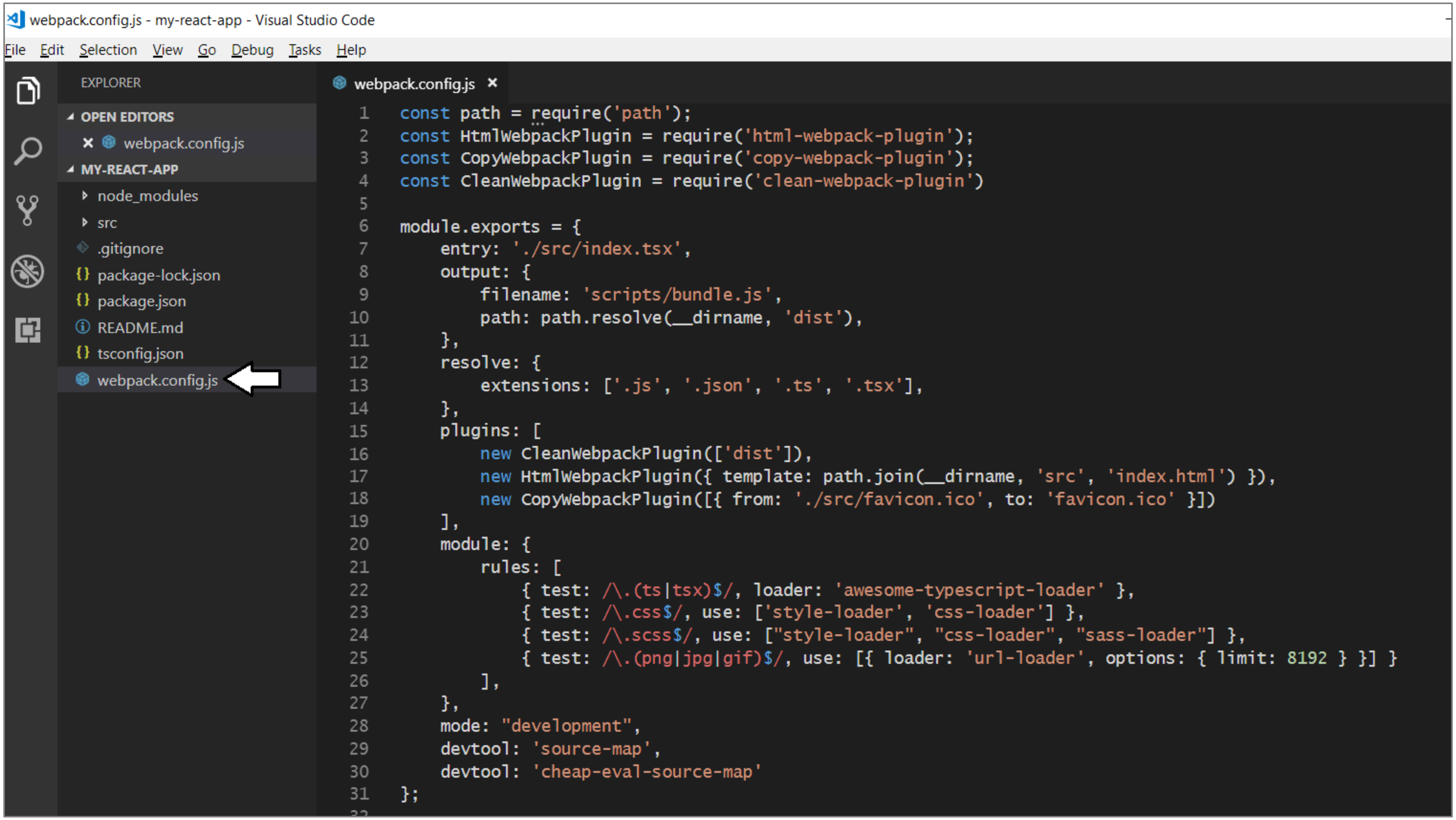
WebPack

- WebPack serves as a bundling utility
  - Bundles many js/ts files into a single file
  - Can handle dynamic module loading
  - Provides a dev server for testing and debugging

- When using Webpack 4
  - Install packages for webpack and webpack-cli

```
npm install webpack webpack-cli --save-dev
```

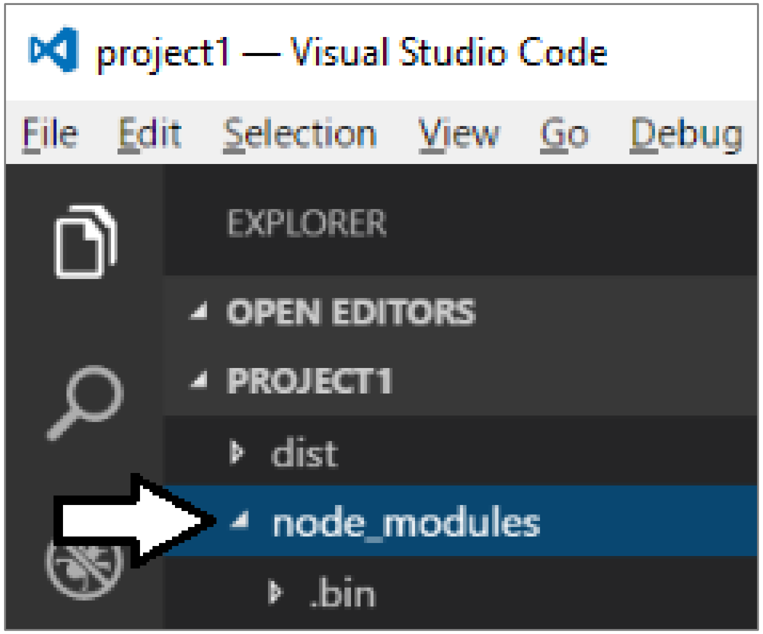
Starter Project - webpack.config.js



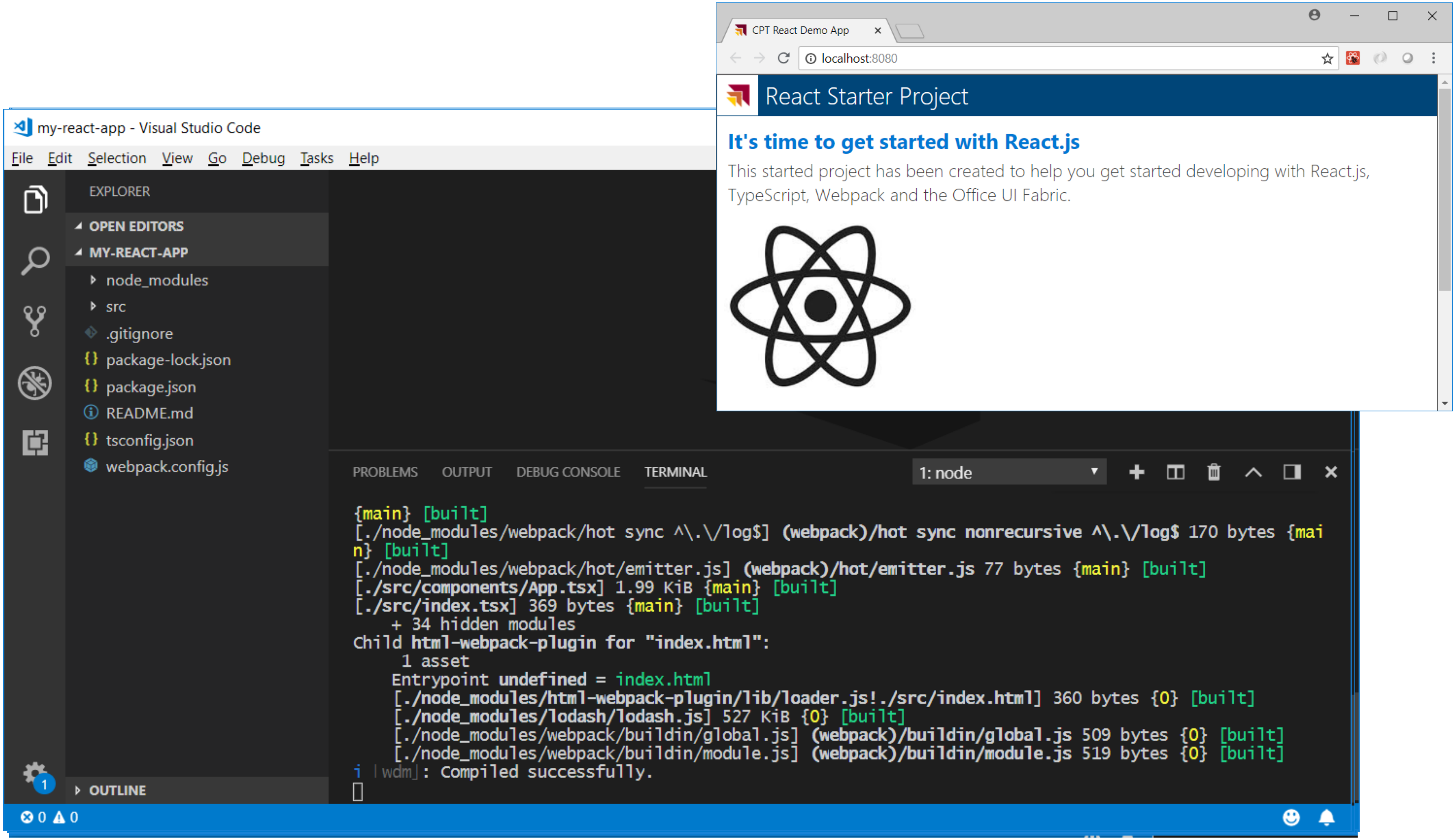
```
const path = require('path');
const HtmlWebpackPlugin = require('html-webpack-plugin');
const CopyWebpackPlugin = require('copy-webpack-plugin');
const CleanWebpackPlugin = require('clean-webpack-plugin')

module.exports = {
    entry: './src/index.tsx',
    output: {
        filename: 'scripts/bundle.js',
        path: path.resolve(__dirname, 'dist'),
    },
    resolve: {
        extensions: ['.js', '.json', '.ts', '.tsx'],
    },
    plugins: [
        new CleanWebpackPlugin(['dist']),
        new HtmlWebpackPlugin({ template: path.join(__dirname, 'src', 'index.html') }),
        new CopyWebpackPlugin([{ from: './src/favicon.ico', to: 'favicon.ico' }])
    ],
    module: {
        rules: [
            { test: /\.(ts|tsx)$/, loader: 'awesome-typescript-loader' },
            { test: /\.css$/, use: ['style-loader', 'css-loader'] },
            { test: /\.scss$/, use: ["style-loader", "css-loader", "sass-loader"] },
            { test: /\.(png|jpg|gif)$/, use: [{ loader: 'url-loader', options: { limit: 8192 } }] }
        ],
    },
    mode: "development",
    devtool: 'source-map',
    devtool: 'cheap-eval-source-map'
};
```

node_modules folder

- Package files copied into **node_modules** folder
  - This folder often contain 100s of packages for a project
  - Contents of folder not saved into source control
  - Contents can be restored with **npm install** command



Executing npm Commands in Visual Studio Code



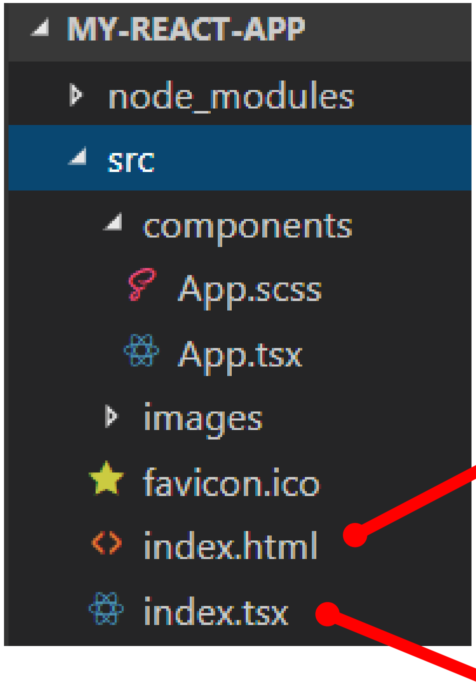
React Starter Project

It's time to get started with React.js

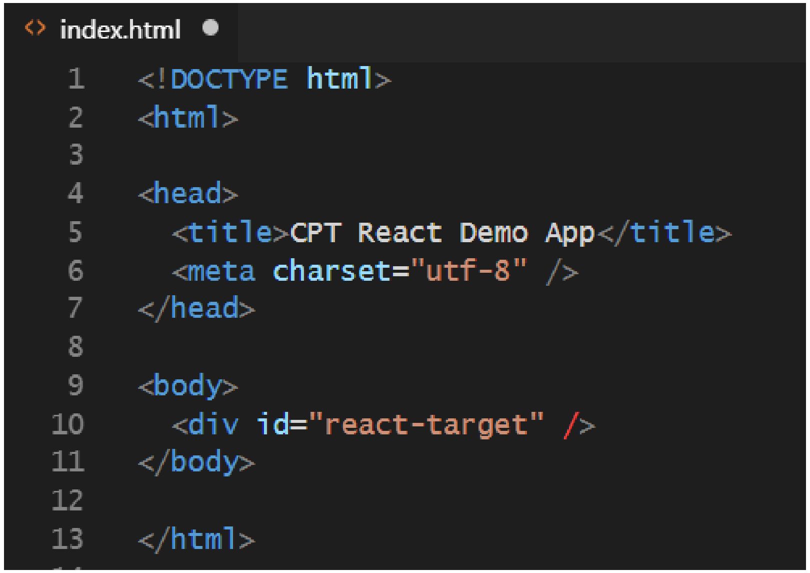
This started project has been created to help you get started developing with React.js, TypeScript, Webpack and the Office UI Fabric.

```
{main} [built]
[./node_modules/webpack/hot sync ^\.\/log$] (webpack)/hot sync nonrecursive ^\.\/log$ 170 bytes {mai
n} [built]
[./node_modules/webpack/hot/emitter.js] (webpack)/hot/emitter.js 77 bytes {main} [built]
[./src/components/App.tsx] 1.99 KiB {main} [built]
[./src/index.tsx] 369 bytes {main} [built]
    + 34 hidden modules
Child html-webpack-plugin for "index.html":
     1 asset
    Entrypoint undefined = index.html
    [./node_modules/html-webpack-plugin/lib/loader.js!./src/index.html] 360 bytes {0} [built]
    [./node_modules/lodash/lodash.js] 527 KiB {0} [built]
    [./node_modules/webpack/buildin/global.js] (webpack)/buildin/global.js 509 bytes {0} [built]
    [./node_modules/webpack/buildin/module.js] (webpack)/buildin/module.js 519 bytes {0} [built]
i ｢wdm｣: Compiled successfully.
```

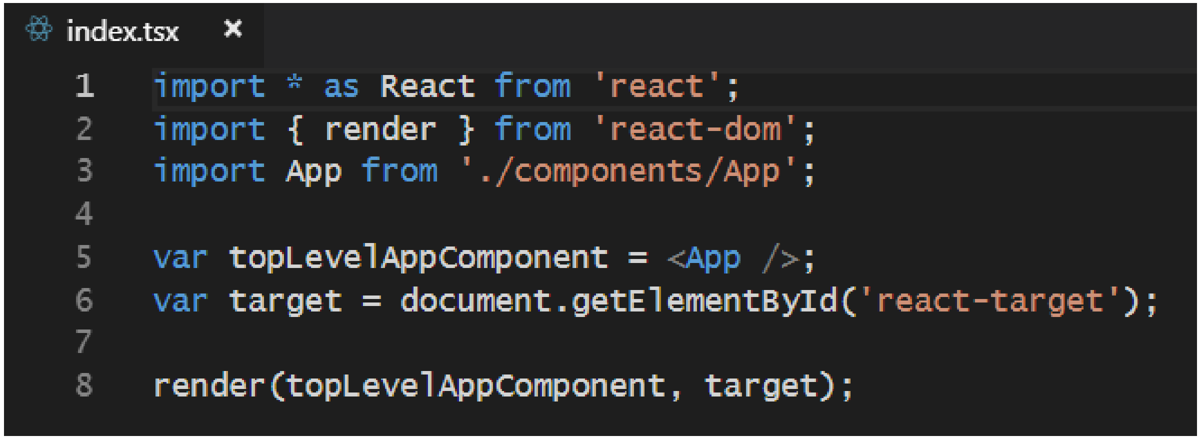
Starter Project Structure



```
MY-REACT-APP
  node_modules
  src
    components
      App.scss
      App.tsx
    images
    favicon.ico
    index.html
    index.tsx
```



```html
<!DOCTYPE html>
<html>

<head>
  <title>CPT React Demo App</title>
  <meta charset="utf-8" />
</head>

<body>
  <div id="react-target" />
</body>

</html>
```



```tsx
import * as React from 'react';
import { render } from 'react-dom';
import App from './components/App';

var topLevelAppComponent = <App />;
var target = document.getElementById('react-target');

render(topLevelAppComponent, target);
```

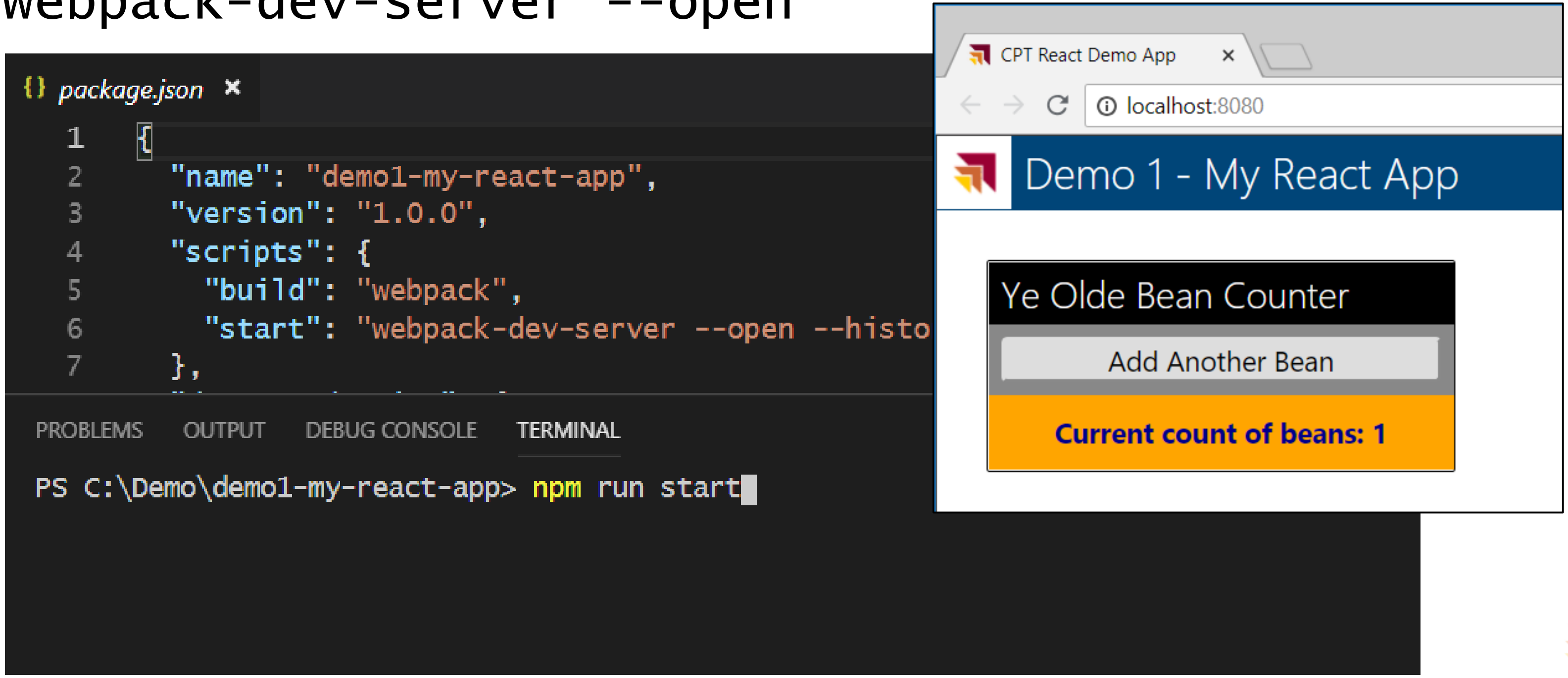
Webpack Dev Server

- Webpack provides its own development server
  - Install the webpack dev server package

    ```
    npm install webpack-dev-server --save-dev
    ```

  - Run your project using the webpack dev server CLI

    ```
    webpack-dev-server --open
    ```



package.json

```
{
  "name": "demo1-my-react-app",
  "version": "1.0.0",
  "scripts": {
    "build": "webpack",
    "start": "webpack-dev-server --open --histo
  },
```

```
PS C:\Demo\demo1-my-react-app> npm run start
```

CPT React Demo App

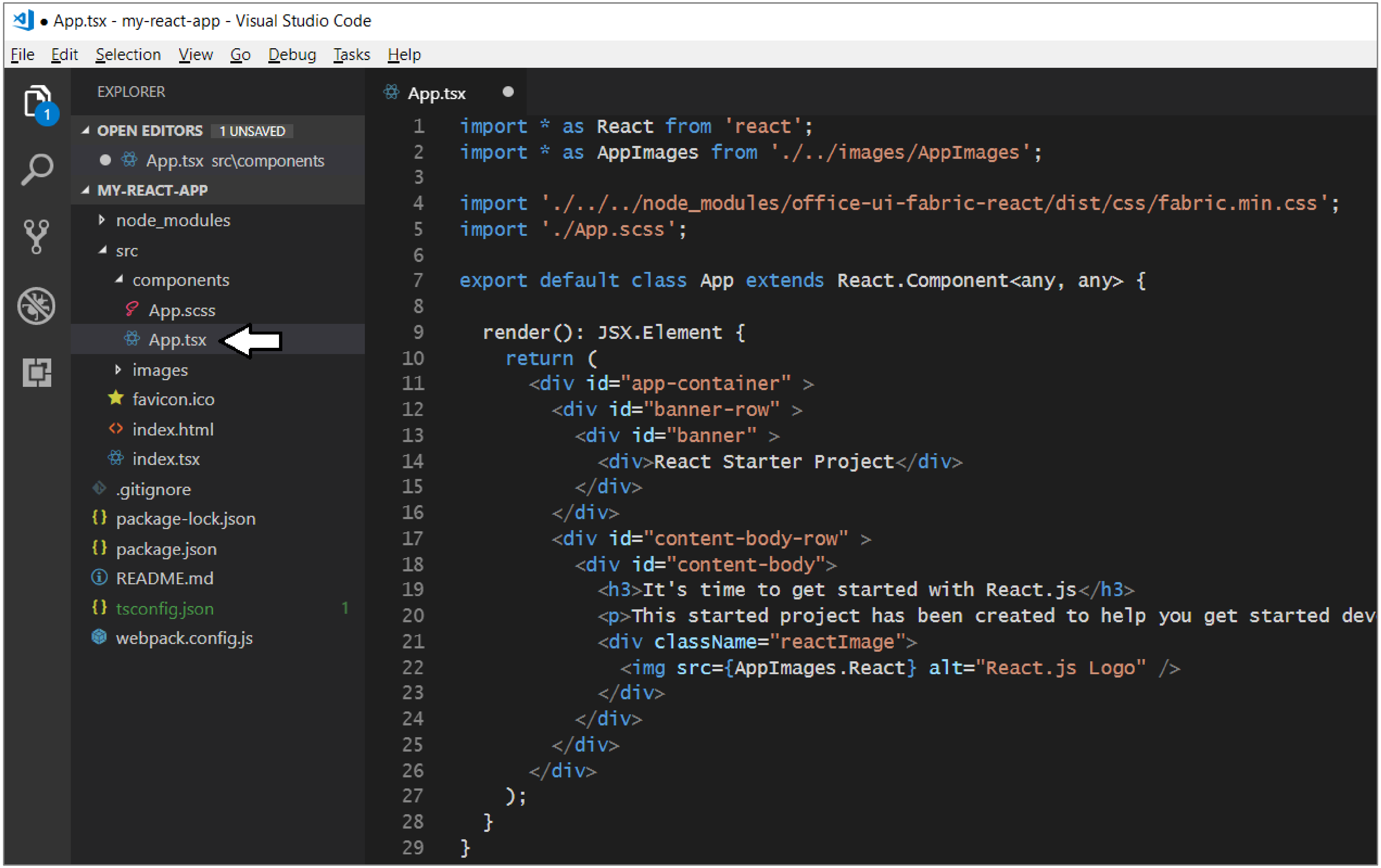
localhost:8080

Demo 1 - My React App

Ye Olde Bean Counter

Current count of beans: 1

The Top-level App Component



```tsx
import * as React from 'react';
import * as AppImages from './../images/AppImages';

import './../../node_modules/office-ui-fabric-react/dist/css/fabric.min.css';
import './App.scss';

export default class App extends React.Component<any, any> {

  render(): JSX.Element {
    return (
      <div id="app-container" >
        <div id="banner-row" >
          <div id="banner" >
            <div>React Starter Project</div>
          </div>
        </div>
        <div id="content-body-row" >
          <div id="content-body">
            <h3>It's time to get started with React.js</h3>
            <p>This started project has been created to help you get started dev
            <div className="reactImage">
              <img src={AppImages.React} alt="React.js Logo" />
            </div>
          </div>
        </div>
      </div>
    );
  }
}
```

Agenda

- ✓ Developing with Node.js, TypeScript and Webpack
- ➢ Learning React.js Fundamentals
- Using the Office UI Fabric React Component Library
- Developing React Webparts with SharePoint Framework
- Calling the Microsoft Graph API from React Webparts

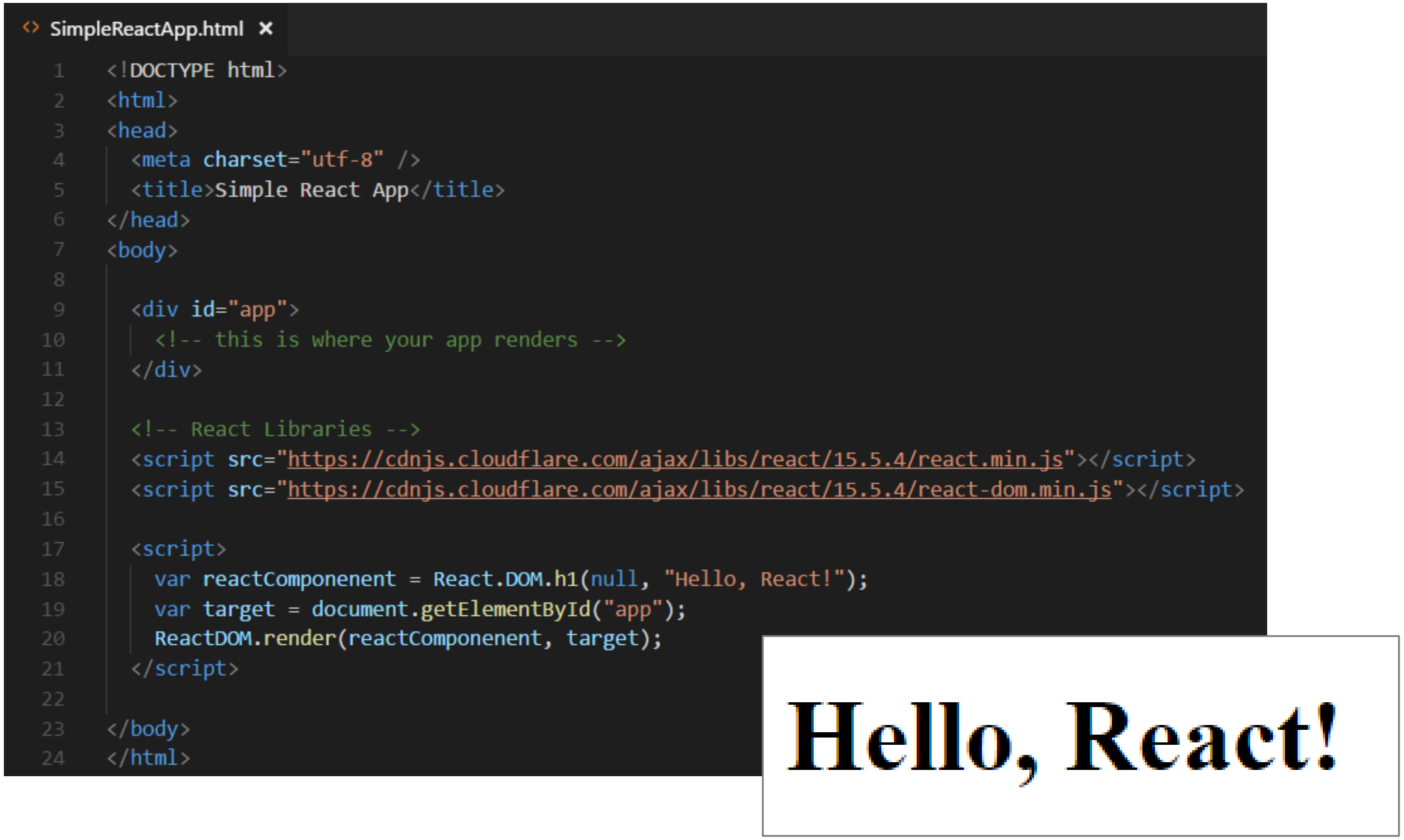
Introducing React

- React is a library for building UI experiences
  - Not as all-encompassing as a framework like Angular
  - Focused on building HTML-based user experiences
  - Based on reusable component-based architecture
  - Components *react* to state changes by updating UI
  - React uses shadow DOM for efficient event handling

- React was originally designed for Facebook
  - Also a good fit for building SPFx web parts

Hello World with React.js and JavaScript

- Obtain the React library with npm or from a CDN
  - **npm install react --save**
  - **npm install react-dom --save**



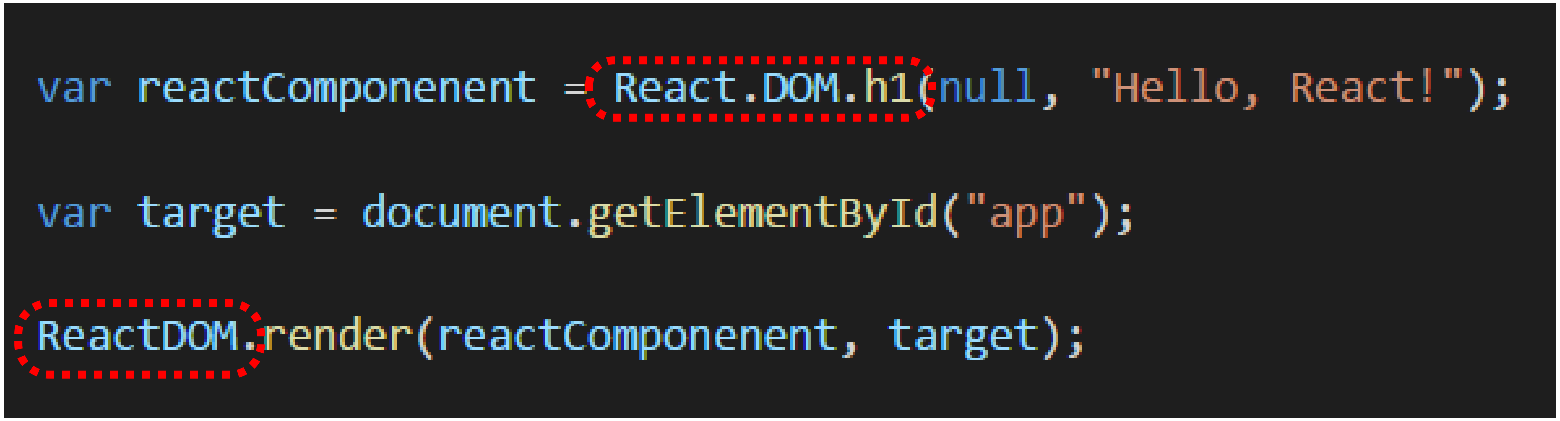
SimpleReactApp.html

```html
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Simple React App</title>
</head>
<body>

  <div id="app">
    <!-- this is where your app renders -->
  </div>

  <!-- React Libraries -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react/15.5.4/react.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react/15.5.4/react-dom.min.js"></script>

  <script>
    var reactComponenent = React.DOM.h1(null, "Hello, React!");
    var target = document.getElementById("app");
    ReactDOM.render(reactComponenent, target);
  </script>

</body>
</html>
```

```
Hello, React!
```

React versus ReactDOM

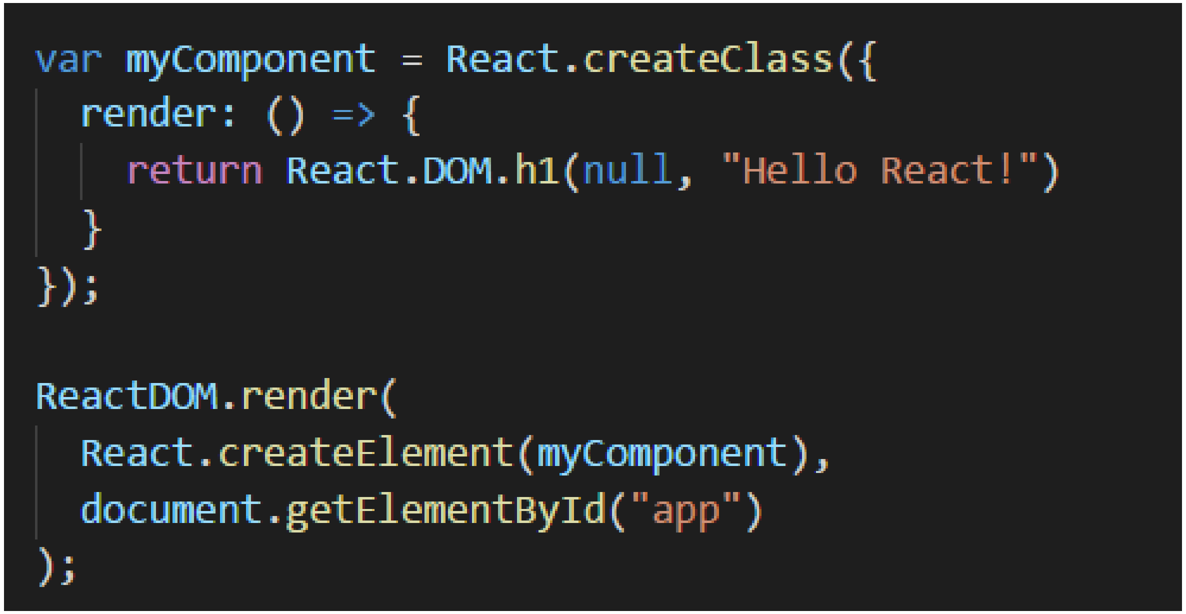
- **React** and **ReactDOM** are separate libraries
  - **React (react.js)** is the primary library used to build out user experiences
  - **ReactDOM** (**react-dom.js**) is used to render **React** user experience in the browser
- **React** library exposes global **React** object
  - **React** object is the main entry point into React API
  - **React.DOM** wraps standard HTML elements
- **ReactDOM** library exposes global **ReactDOM** object
  - **ReactDOM** object used to render React components on web page



```
var reactComponenent = React.DOM.h1(null, "Hello, React!");

var target = document.getElementById("app");

ReactDOM.render(reactComponenent, target);
```

React Component Created Using ES5

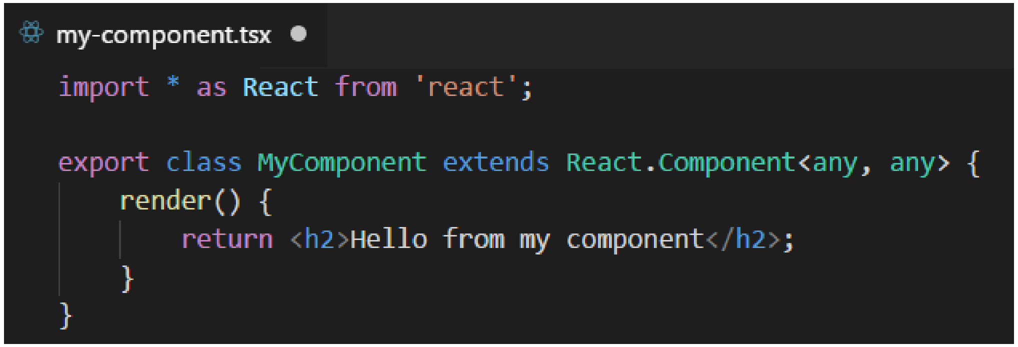
- React component can be created using EcmaScript 5
  - React component definition created using **React.createClass**
  - React component must be defined with **render** method
  - React component can be instantiated with **React.createElement**



```
var myComponent = React.createClass({
  render: () => {
    return React.DOM.h1(null, "Hello React!")
  }
});

ReactDOM.render(
  React.createElement(myComponent),
  document.getElementById("app")
);
```

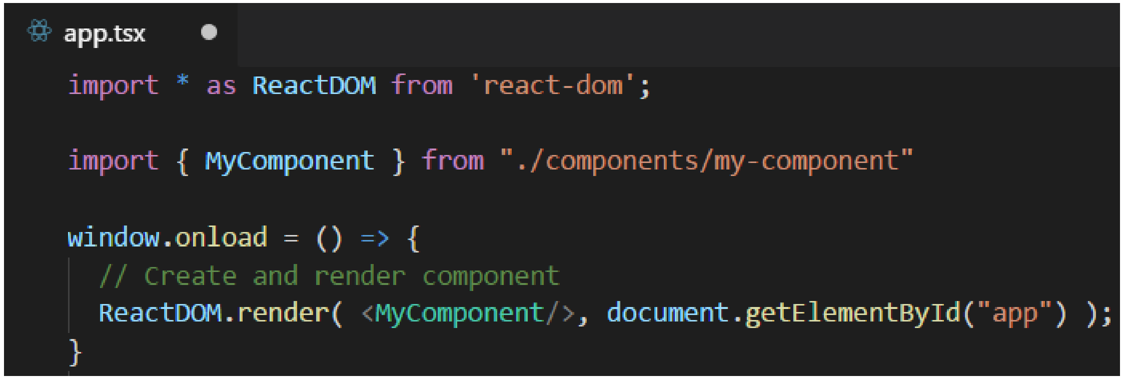
Defining React Components using TypeScript

- Component is class extending `React.Component`
  - Component usually defined in its own **tsx** file
  - Component class must define **render** method



```
import * as React from 'react';

export class MyComponent extends React.Component<any, any> {
    render() {
        return <h2>Hello from my component</h2>;
    }
}
```

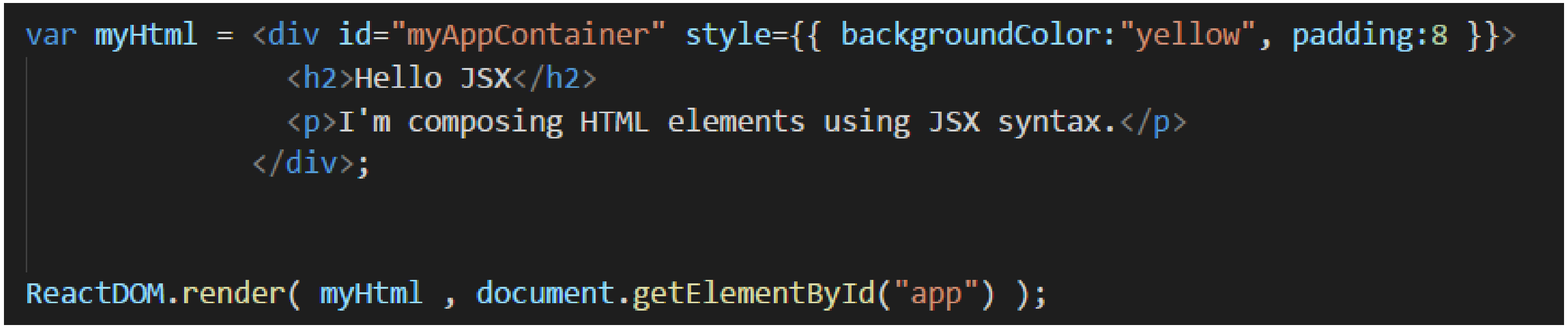
  - Component can be instantiated with JSX/TSX syntax



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

import { MyComponent } from "./components/my-component"

window.onload = () => {
  // Create and render component
  ReactDOM.render( <MyComponent/>, document.getElementById("app") );
}
```

Understanding JSX (and TSX)

- JSX provides better syntax for HTML composition
  - JSX allows extends JavaScript with XML-like syntax
  - JSX syntax must be transpiled into JavaScript code



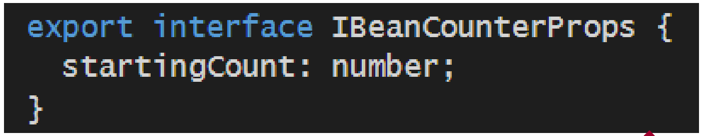
```
var myHtml = <div id="myAppContainer" style={{ backgroundColor:"yellow", padding:8 }}>
               <h2>Hello JSX</h2>
               <p>I'm composing HTML elements using JSX syntax.</p>
             </div>;


ReactDOM.render( myHtml , document.getElementById("app") );
```

- JSX/TSX is separate from React library
  - JSX/TSX commonly used in React development
  - Babel compiler used to transpile JSX to JavaScript
  - TypeScript compiler used to transpile TSX to JavaScript

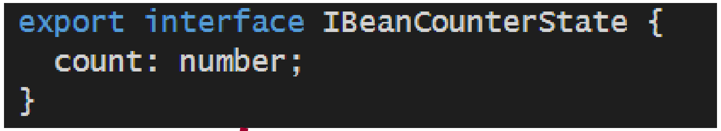
Component Properties and State

- Component can contain properties and state
  - Properties are initialized by external components
  - Properties are read-only to hosting component
  - State is set internally by hosting component
  - Changing state triggers UI refresh by calling render
  - UI experience created by ***reacting*** to changes in state

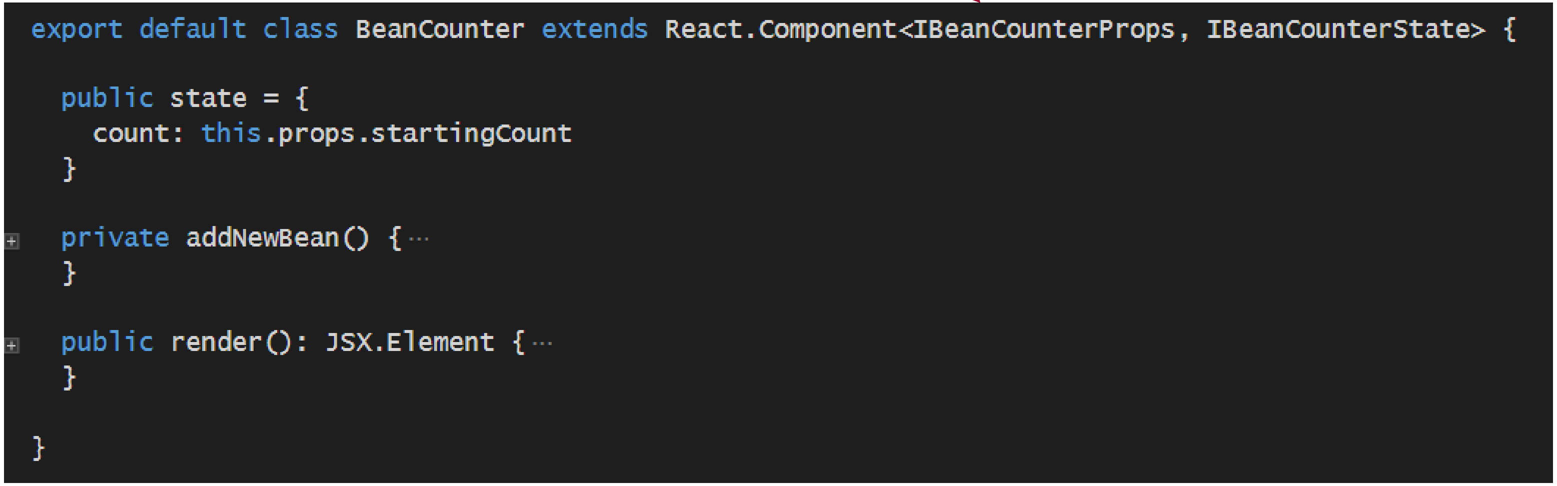
Designing with Properties and State



```
export interface IBeanCounterProps {
  startingCount: number;
}
```



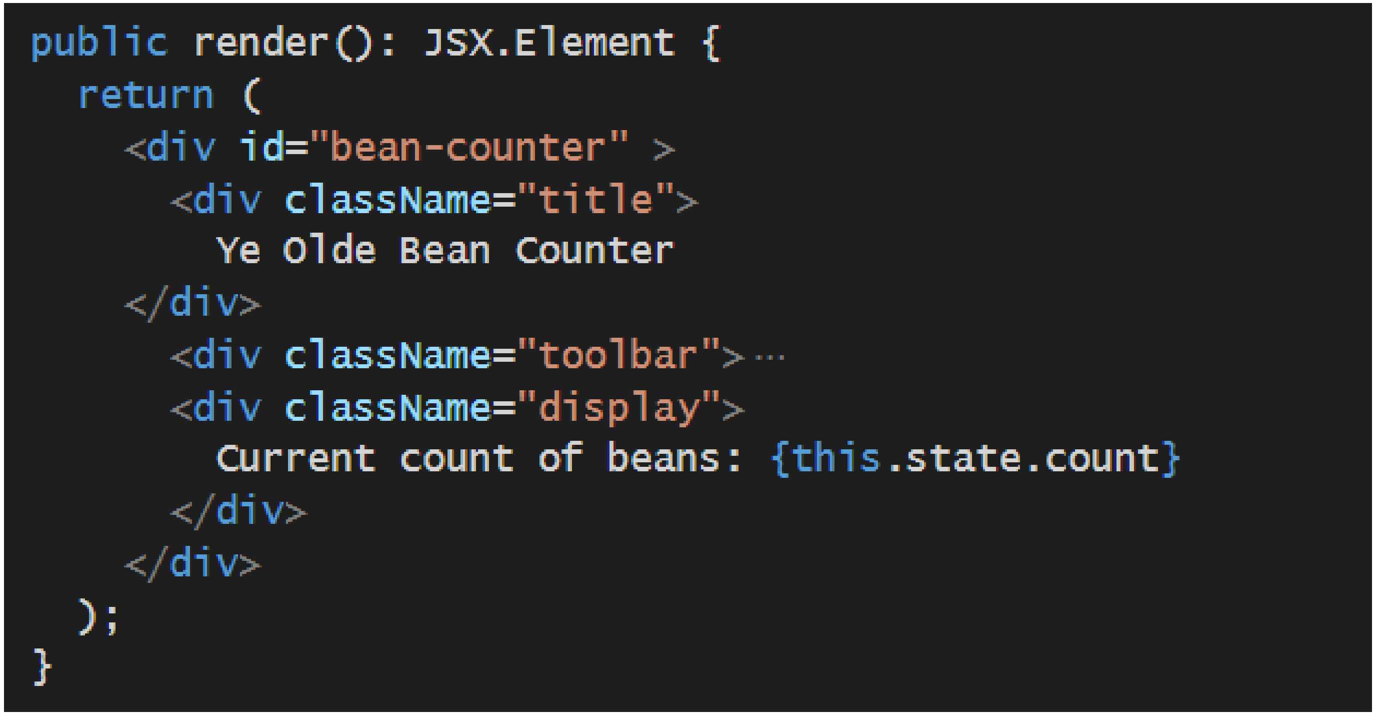
```
export interface IBeanCounterState {
  count: number;
}
```



```
export default class BeanCounter extends React.Component<IBeanCounterProps, IBeanCounterState> {

  public state = {
    count: this.props.startingCount
  }

  private addNewBean() {···
  }

  public render(): JSX.Element {···
  }

}
```

Component Rendering using State

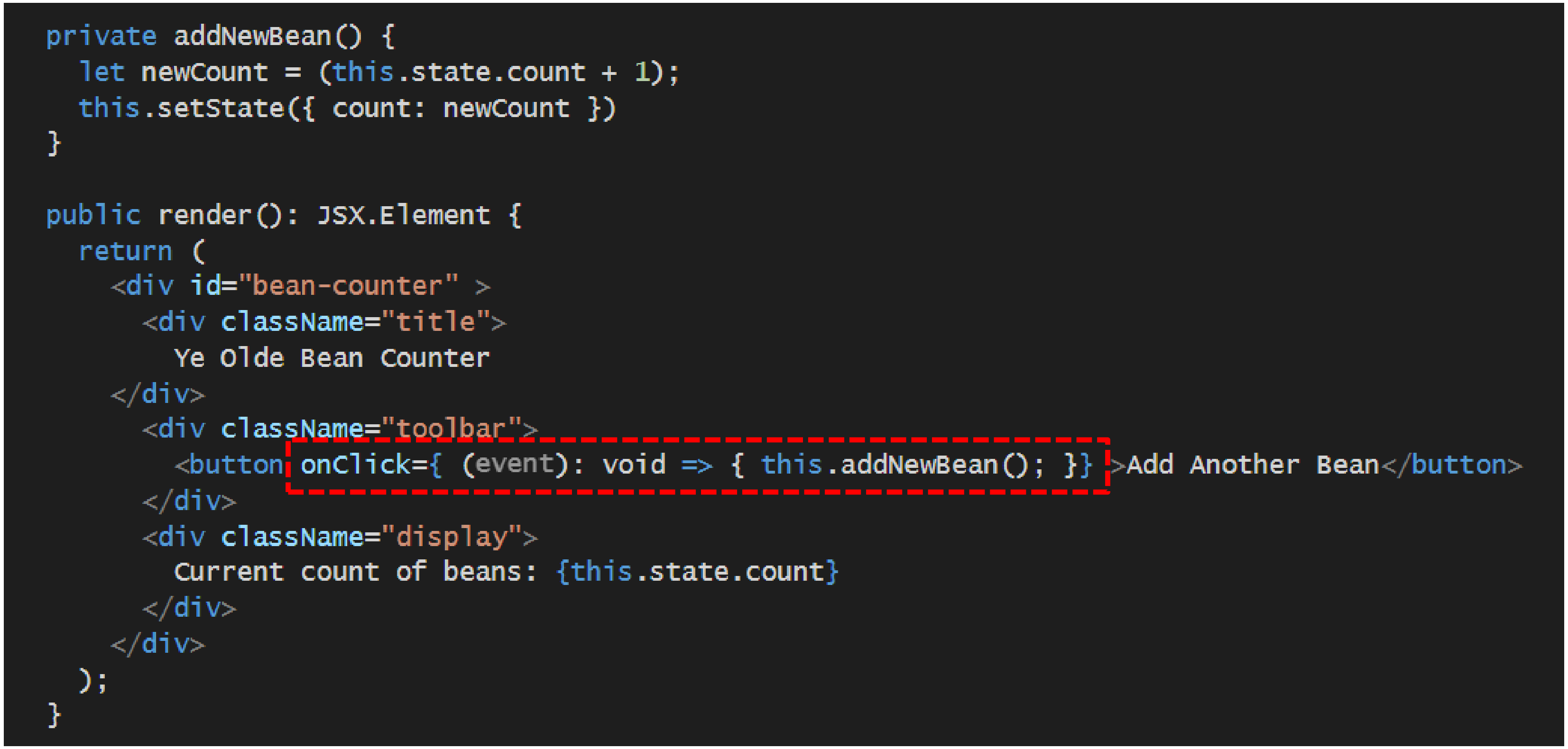
- **render** method implemented using state
  - **render** method executes whenever state is updated



```
public render(): JSX.Element {
  return (
    <div id="bean-counter" >
      <div className="title">
        Ye Olde Bean Counter
    </div>
      <div className="toolbar">⋯
      <div className="display">
        Current count of beans: {this.state.count}
      </div>
    </div>
  );
}
```

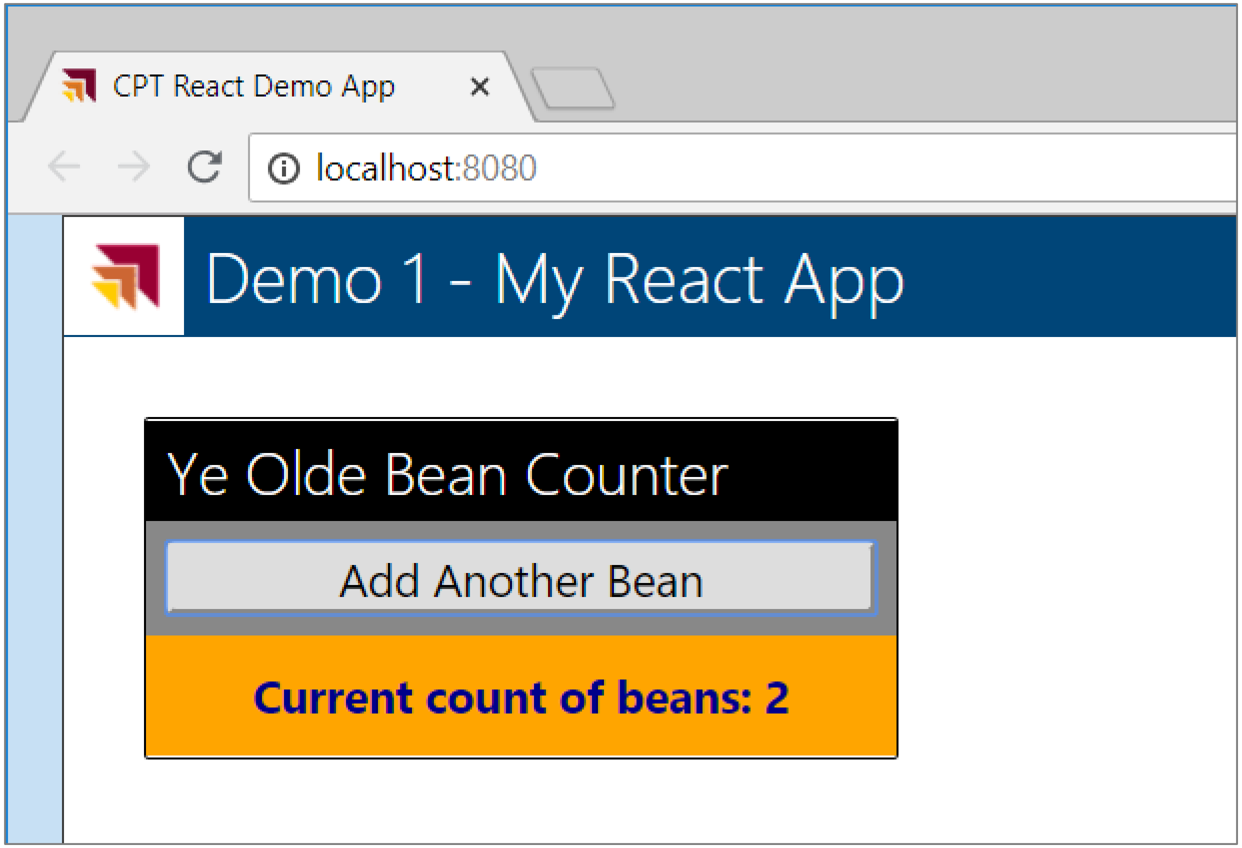
React Provides Synthetic Events

- Replaces standard DOM-based event handling
  - React creates virtual DOM for elements created by component
  - Provides high-performant code in busy web pages

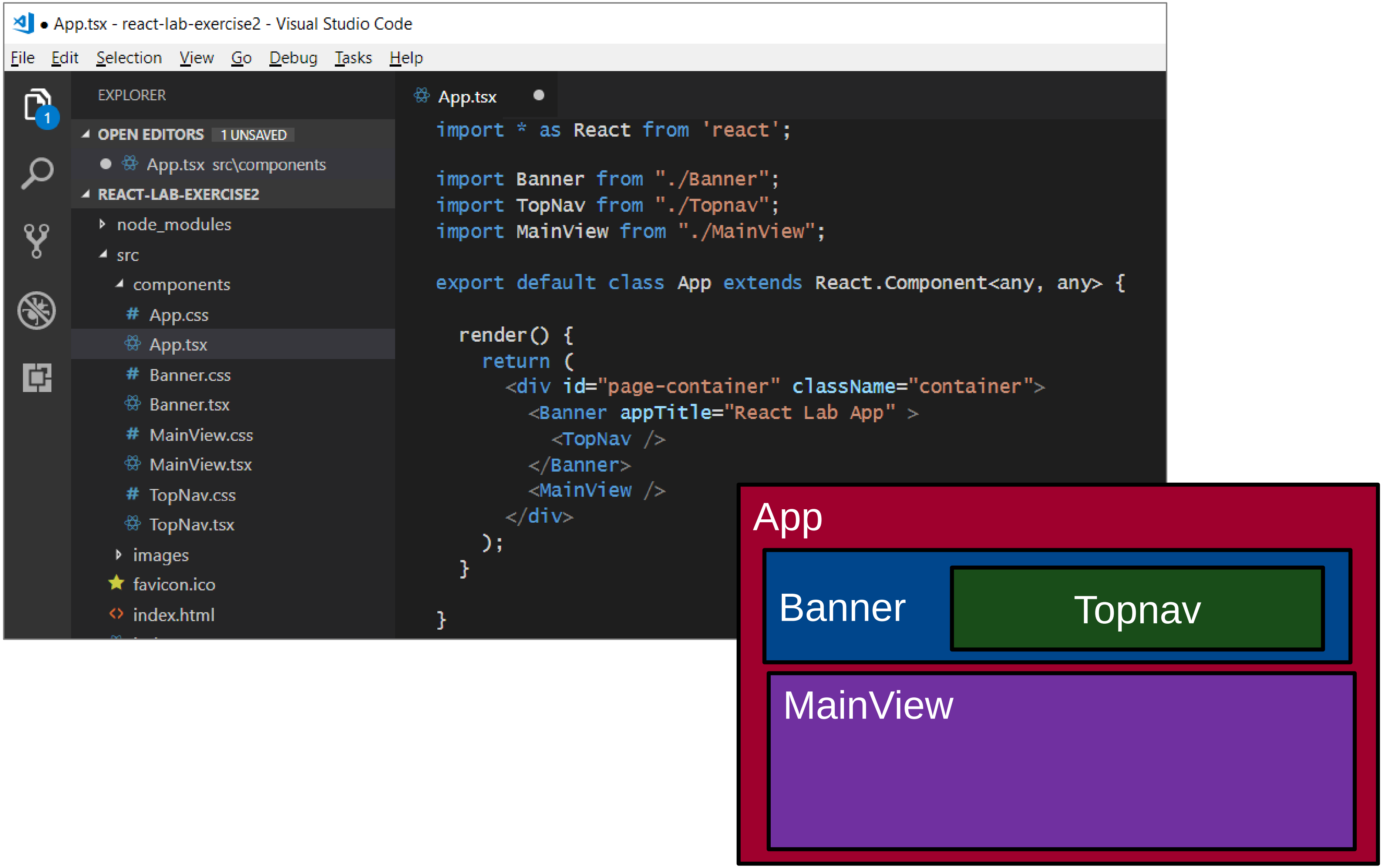


```
private addNewBean() {
  let newCount = (this.state.count + 1);
  this.setState({ count: newCount })
}

public render(): JSX.Element {
  return (
    <div id="bean-counter" >
      <div className="title">
        Ye Olde Bean Counter
    </div>
      <div className="toolbar">
        <button onClick={ (event): void => { this.addNewBean(); }} >Add Another Bean</button>
      </div>
      <div className="display">
        Current count of beans: {this.state.count}
      </div>
    </div>
  );
}
```

Demo 1 - Creating a Simple React Component

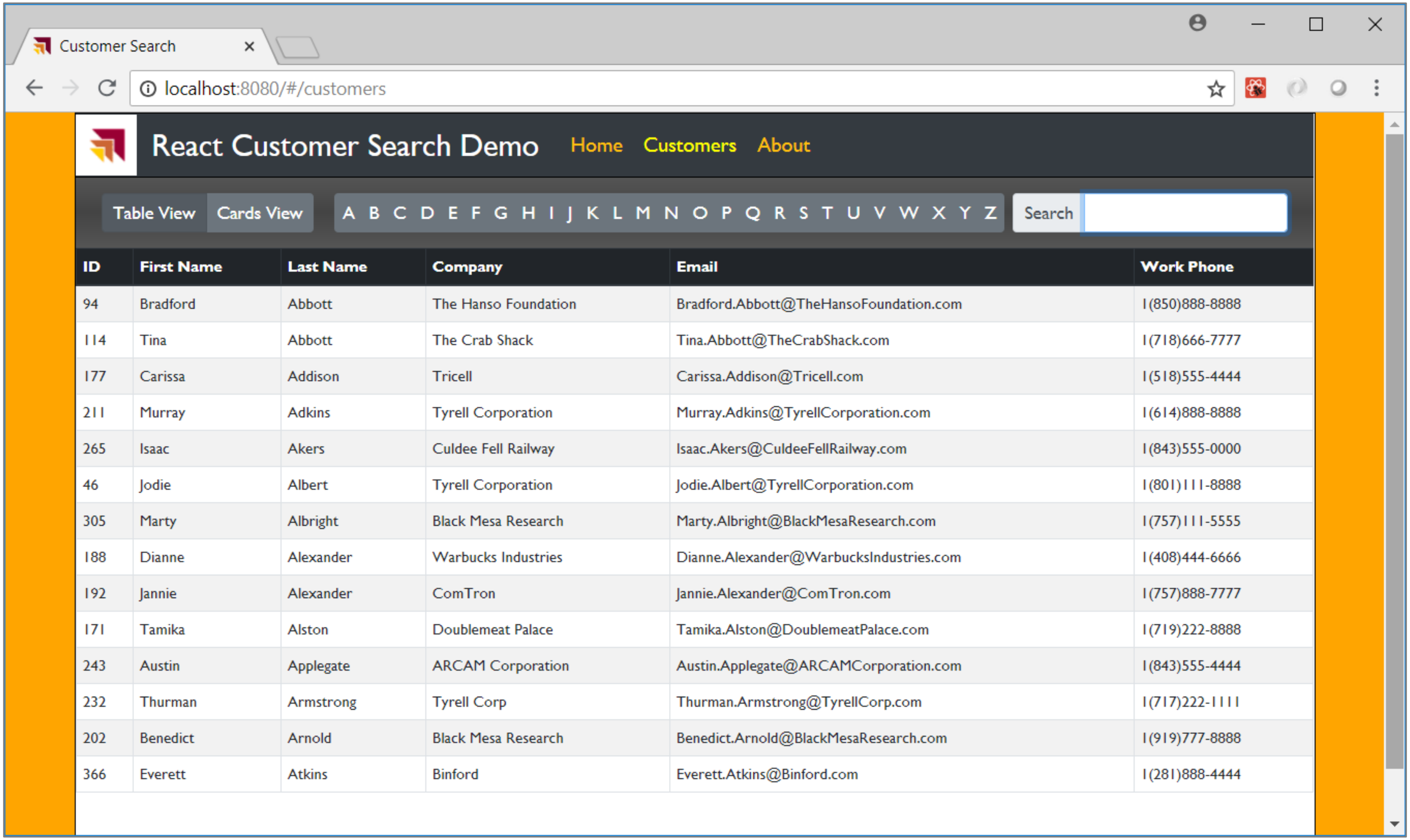


React Component Hierarchies



```
import * as React from 'react';

import Banner from "./Banner";
import TopNav from "./Topnav";
import MainView from "./MainView";

export default class App extends React.Component<any, any> {

  render() {
    return (
      <div id="page-container" className="container">
        <Banner appTitle="React Lab App" >
          <TopNav />
        </Banner>
        <MainView />
      </div>
    );
  }

}
```

Demo 2 - Customer Search


Customer Search
localhost:8080/#/customers
React Customer Search Demo
Home
Customers
About
Table View
Cards View
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Search

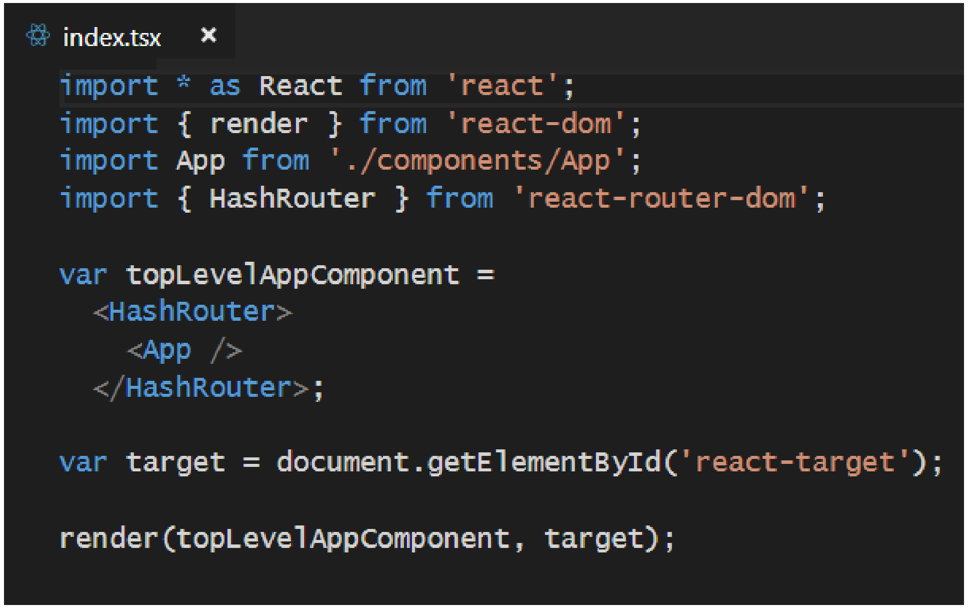
ID	First Name	Last Name	Company	Email	Work Phone
94	Bradford	Abbott	The Hanso Foundation	Bradford.Abbott@TheHansoFoundation.com	1(850)888-8888
114	Tina	Abbott	The Crab Shack	Tina.Abbott@TheCrabShack.com	1(718)666-7777
177	Carissa	Addison	Tricell	Carissa.Addison@Tricell.com	1(518)555-4444
211	Murray	Adkins	Tyrell Corporation	Murray.Adkins@TyrellCorporation.com	1(614)888-8888
265	Isaac	Akers	Culdee Fell Railway	Isaac.Akers@CuldeeFellRailway.com	1(843)555-0000
46	Jodie	Albert	Tyrell Corporation	Jodie.Albert@TyrellCorporation.com	1(801)111-8888
305	Marty	Albright	Black Mesa Research	Marty.Albright@BlackMesaResearch.com	1(757)111-5555
188	Dianne	Alexander	Warbucks Industries	Dianne.Alexander@WarbucksIndustries.com	1(408)444-6666
192	Jannie	Alexander	ComTron	Jannie.Alexander@ComTron.com	1(757)888-7777
171	Tamika	Alston	Doublemeat Palace	Tamika.Alston@DoublemeatPalace.com	1(719)222-8888
243	Austin	Applegate	ARCAM Corporation	Austin.Applegate@ARCAMCorporation.com	1(843)555-4444
232	Thurman	Armstrong	Tyrell Corp	Thurman.Armstrong@TyrellCorp.com	1(717)222-1111
202	Benedict	Arnold	Black Mesa Research	Benedict.Arnold@BlackMesaResearch.com	1(919)777-8888
366	Everett	Atkins	Binford	Everett.Atkins@Binford.com	1(281)888-4444

React Router

- Used to create route map in single page application (SPA)
  - Installed as a pair of npm packages

```
npm install react-router @types/react-router --save-dev
npm install react-router-dom @types/react-router-dom --save-dev
```

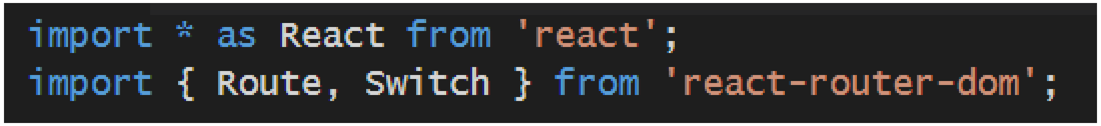
- Router must be added in as top-level component above App



```
import * as React from 'react';
import { render } from 'react-dom';
import App from './components/App';
import { HashRouter } from 'react-router-dom';

var topLevelAppComponent =
  <HashRouter>
    <App />
  </HashRouter>;

var target = document.getElementById('react-target');

render(topLevelAppComponent, target);
```

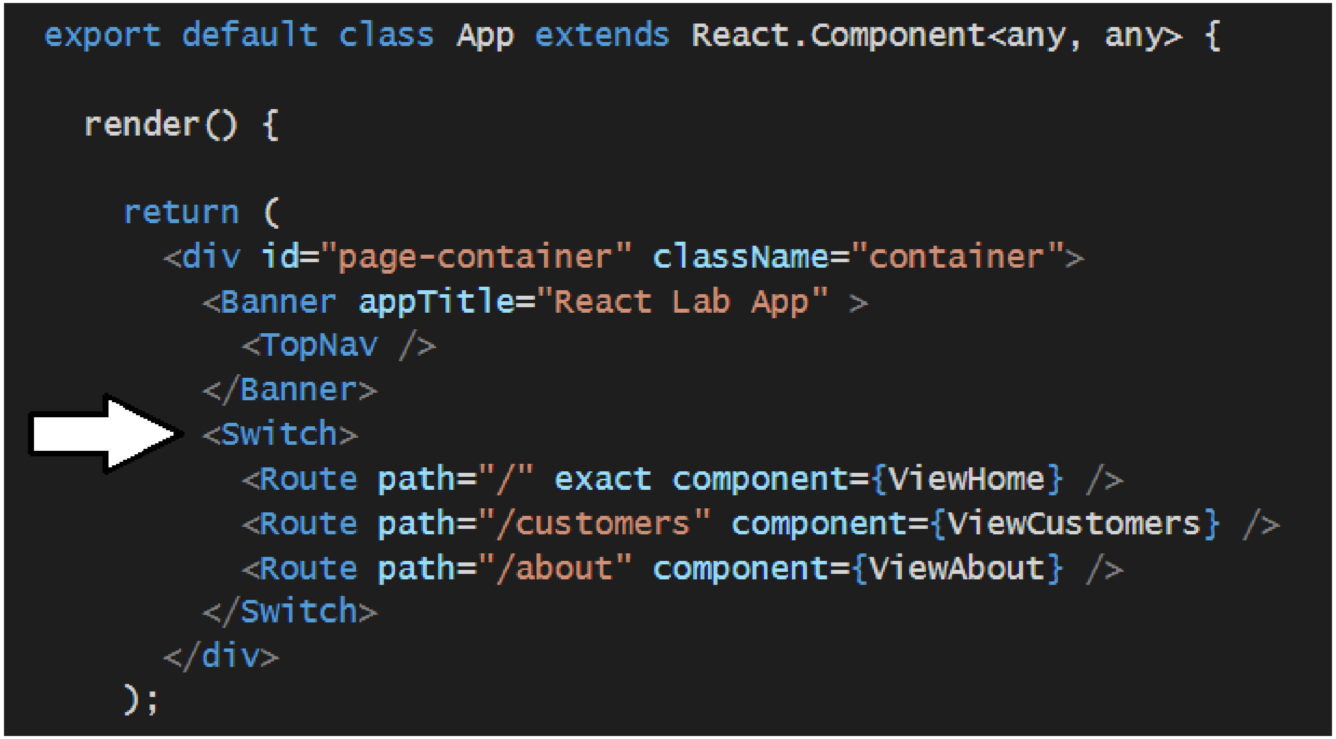
Using React Router

- Import Route and Switch components



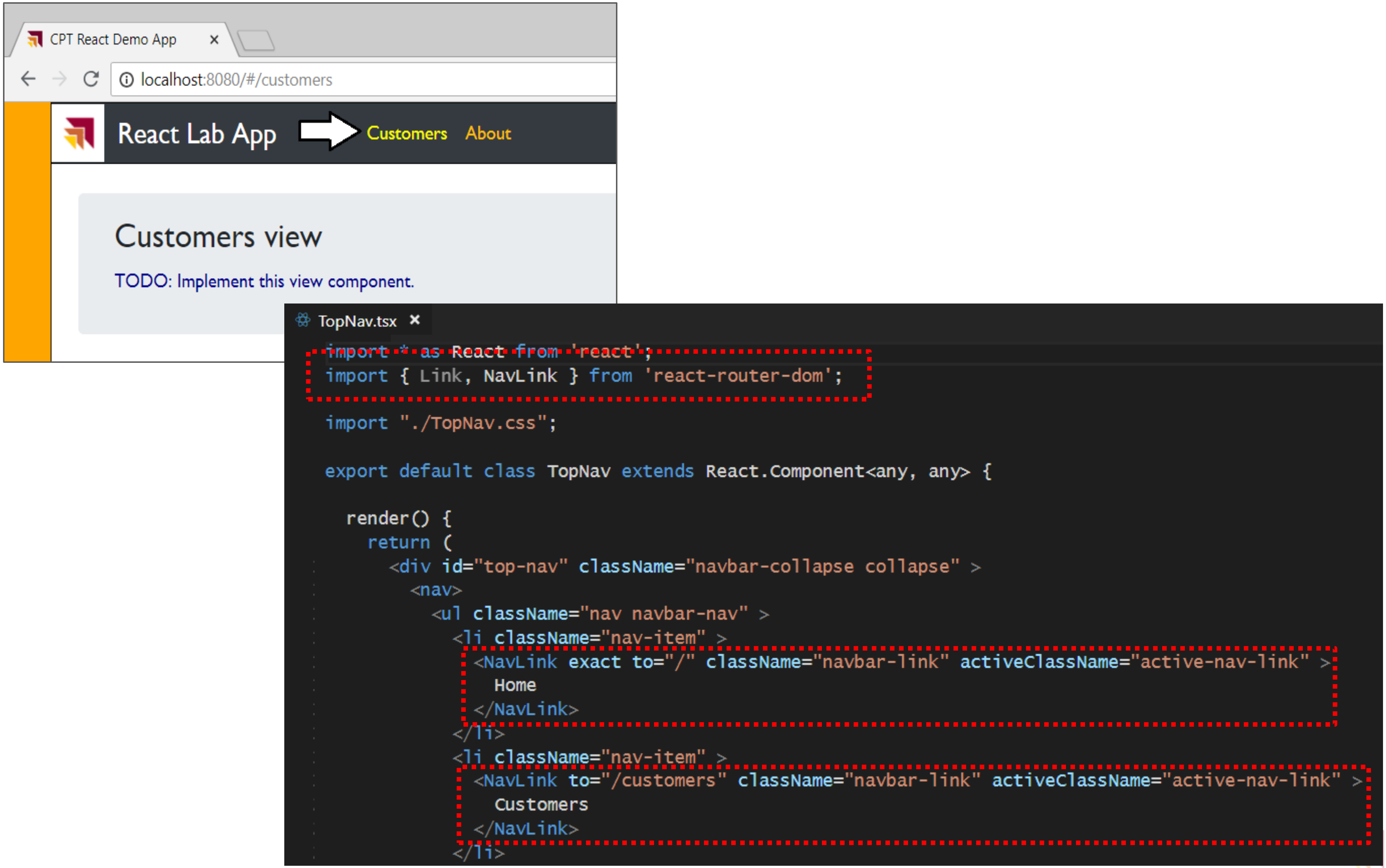
```
import * as React from 'react';
import { Route, Switch } from 'react-router-dom';
```

- Create route map in HTML output



```
export default class App extends React.Component<any, any> {

  render() {

    return (
      <div id="page-container" className="container">
        <Banner appTitle="React Lab App" >
          <TopNav />
        </Banner>
        <Switch>
          <Route path="/" exact component={ViewHome} />
          <Route path="/customers" component={ViewCustomers} />
          <Route path="/about" component={ViewAbout} />
        </Switch>
      </div>
    );
```

Creating Route Links



Customers view

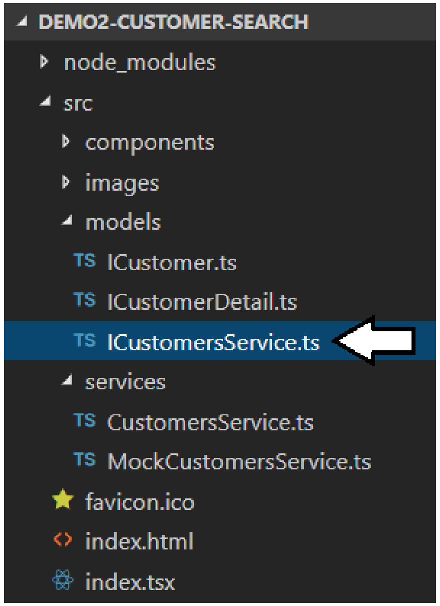
TODO: Implement this view component.

```
import * as React from 'react';
import { Link, NavLink } from 'react-router-dom';

import "./TopNav.css";

export default class TopNav extends React.Component<any, any> {

  render() {
    return (
      <div id="top-nav" className="navbar-collapse collapse" >
        <nav>
          <ul className="nav navbar-nav" >
            <li className="nav-item" >
              <NavLink exact to="/" className="navbar-link" activeClassName="active-nav-link" >
                Home
              </NavLink>
            </li>
            <li className="nav-item" >
              <NavLink to="/customers" className="navbar-link" activeClassName="active-nav-link" >
                Customers
              </NavLink>
            </li>
```

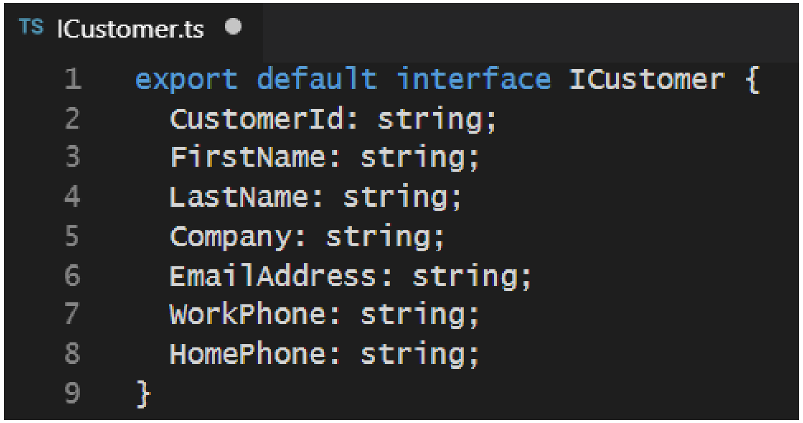
Component Lifecycle

- componentWillUpdate
  - executed before component is rendered
- componentDidUpdate
  - executed after component is rendered
- componentWillMount
  - executed before node is added to the DOM
- componentDidMount
  - executed after node is added to the DOM
- componentWillUnmount
  - executed before node is removed from the DOM
- shouldComponentUpdate(newProps, newState)
  - executed before component is updated

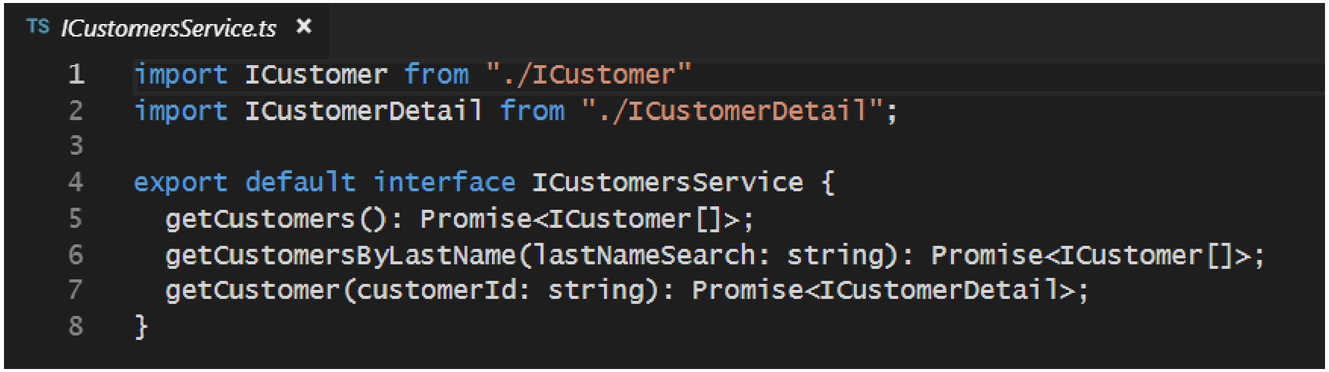
Defining Interfaces for Data Access Code



```
DEMO2-CUSTOMER-SEARCH
  node_modules
  src
    components
    images
    models
      ICustomer.ts
      ICustomerDetail.ts
      ICustomersService.ts
    services
      CustomersService.ts
      MockCustomersService.ts
    favicon.ico
    index.html
    index.tsx
```

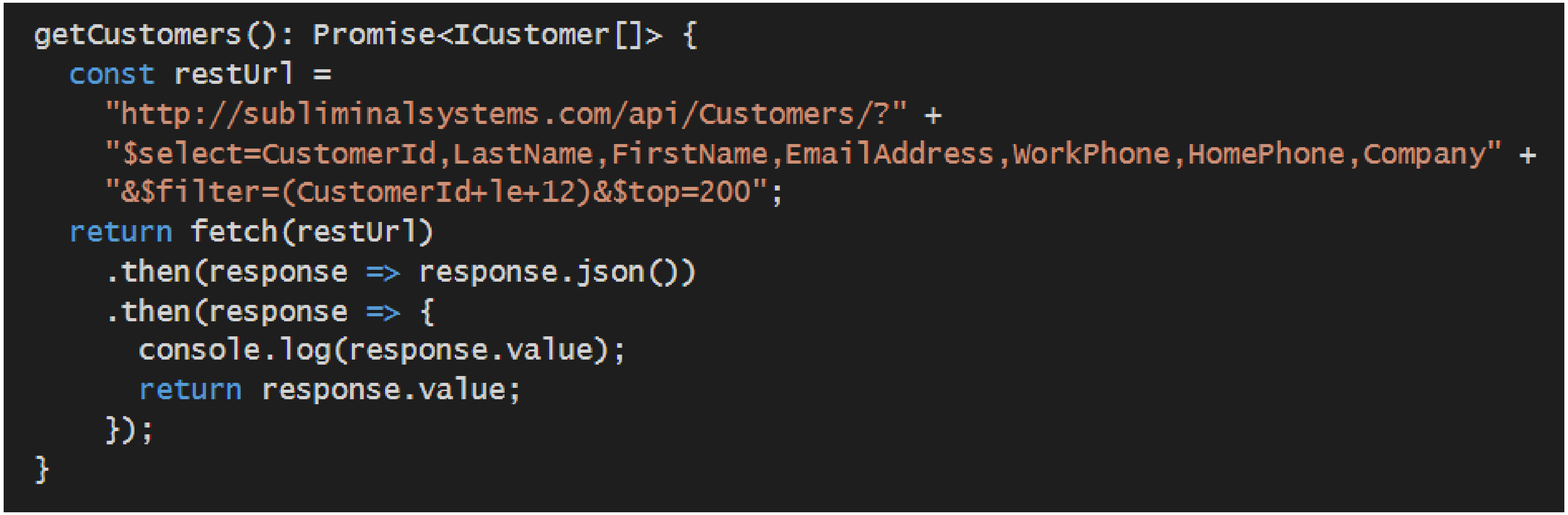


```
export default interface ICustomer {
  CustomerId: string;
  FirstName: string;
  LastName: string;
  Company: string;
  EmailAddress: string;
  WorkPhone: string;
  HomePhone: string;
}
```

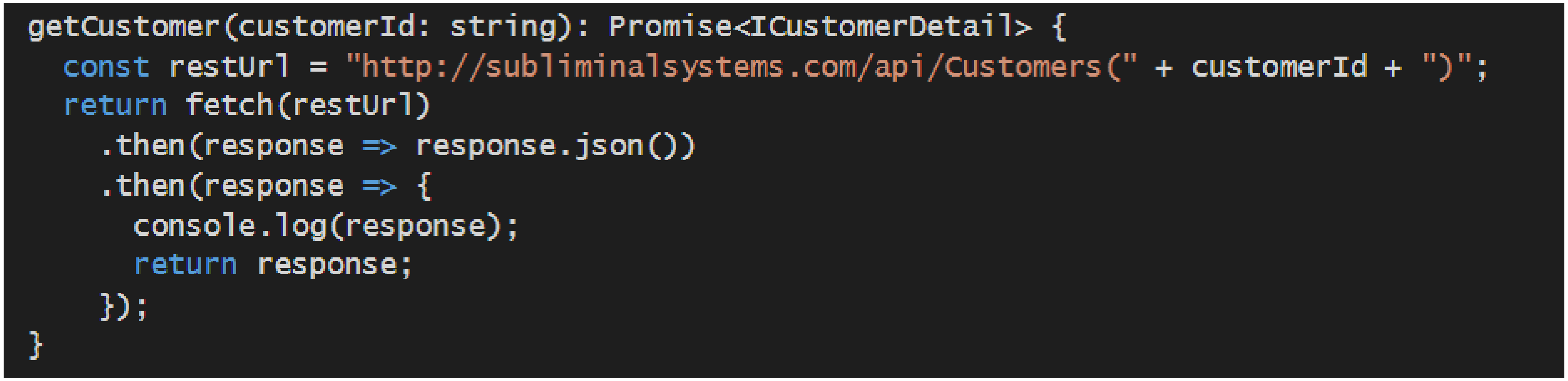


```
import ICustomer from "./ICustomer"
import ICustomerDetail from "./ICustomerDetail";

export default interface ICustomersService {
  getCustomers(): Promise<ICustomer[]>;
  getCustomersByLastName(lastNameSearch: string): Promise<ICustomer[]>;
  getCustomer(customerId: string): Promise<ICustomerDetail>;
}
```

Calling a Web Service using the Fetch API



```
getCustomers(): Promise<ICustomer[]> {
  const restUrl =
    "http://subliminalsystems.com/api/Customers/?" +
    "$select=CustomerId,LastName,FirstName,EmailAddress,WorkPhone,HomePhone,Company" +
    "&$filter=(CustomerId+le+12)&$top=200";
  return fetch(restUrl)
    .then(response => response.json())
    .then(response => {
      console.log(response.value);
      return response.value;
    });
}
```



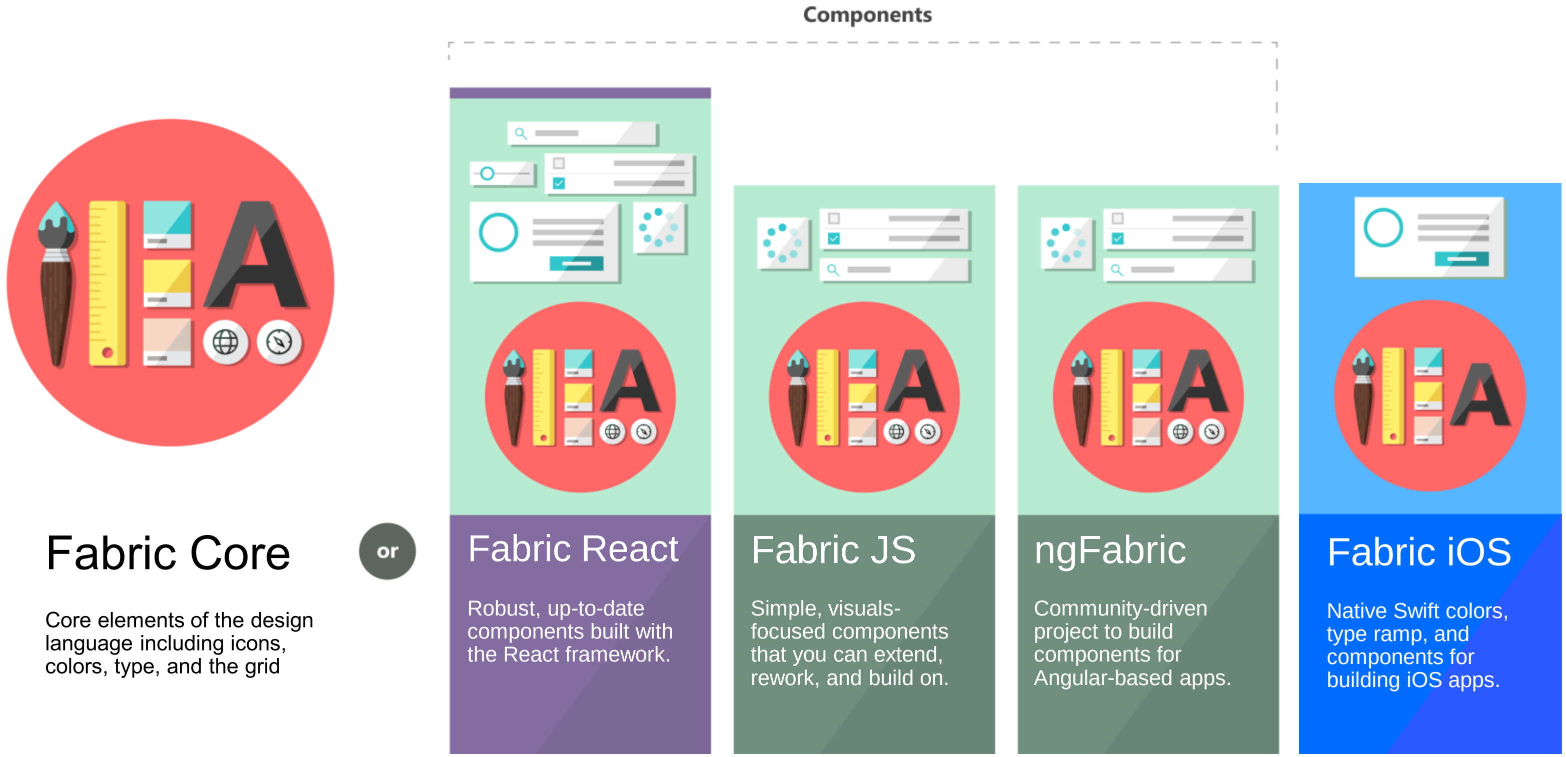
```
getCustomer(customerId: string): Promise<ICustomerDetail> {
  const restUrl = "http://subliminalsystems.com/api/Customers(" + customerId + ")";
  return fetch(restUrl)
    .then(response => response.json())
    .then(response => {
      console.log(response);
      return response;
    });
}
```

Agenda

- ✓ Developing with Node.js, TypeScript and Webpack
- ✓ Learning React.js Fundamentals
- ➢ Using the Office UI Fabric React Component Library
- Developing React Webparts with SharePoint Framework
- Calling the Microsoft Graph API from React Webparts

What is the Office UI Fabric?

- Office UI Fabric is responsive, mobile-first, front-end style framework
  - Built by Microsoft to style Office 365, OneDrive and SharePoint sites
  - All about styling instead of JavaScript
  - Can be used by 3rd party developers



Fabric Core

Core elements of the design language including icons, colors, type, and the grid

Fabric React

Robust, up-to-date components built with the React framework.

Fabric JS

Simple, visuals-focused components that you can extend, rework, and build on.

ngFabric

Community-driven project to build components for Angular-based apps.

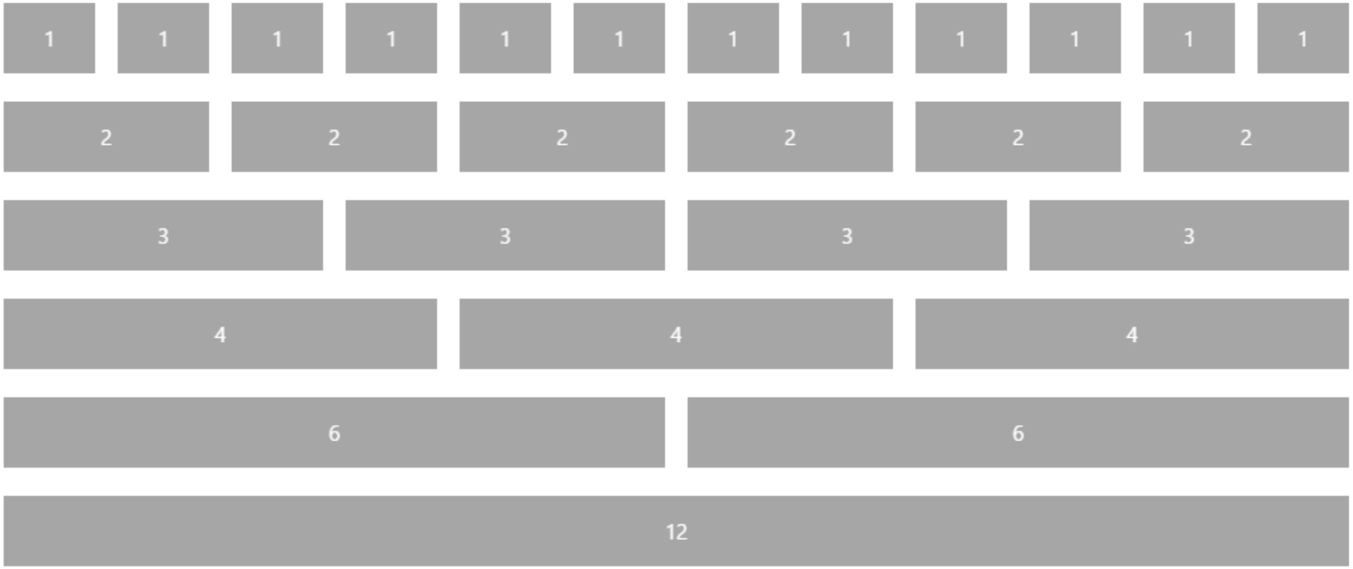
Fabric iOS

Native Swift colors, type ramp, and components for building iOS apps.

Responsive Grid

- Fabric comes with a mobile-first, responsive grid
  - Based on 12 column grid
  - Used to create flexible layouts

```
<div class="ms-Grid">
    <div class="ms-Grid-row">
        <div class="ms-Grid-col ms-u-sm6 ms-u-md4 ms-u-lg2">First</div>
        <div class="ms-Grid-col ms-u-sm6 ms-u-md8 ms-u-lg10">Second</div>
    </div>
</div>
```



1 1 1 1 1 1 1 1 1 1 1 1

2 2 2 2 2 2

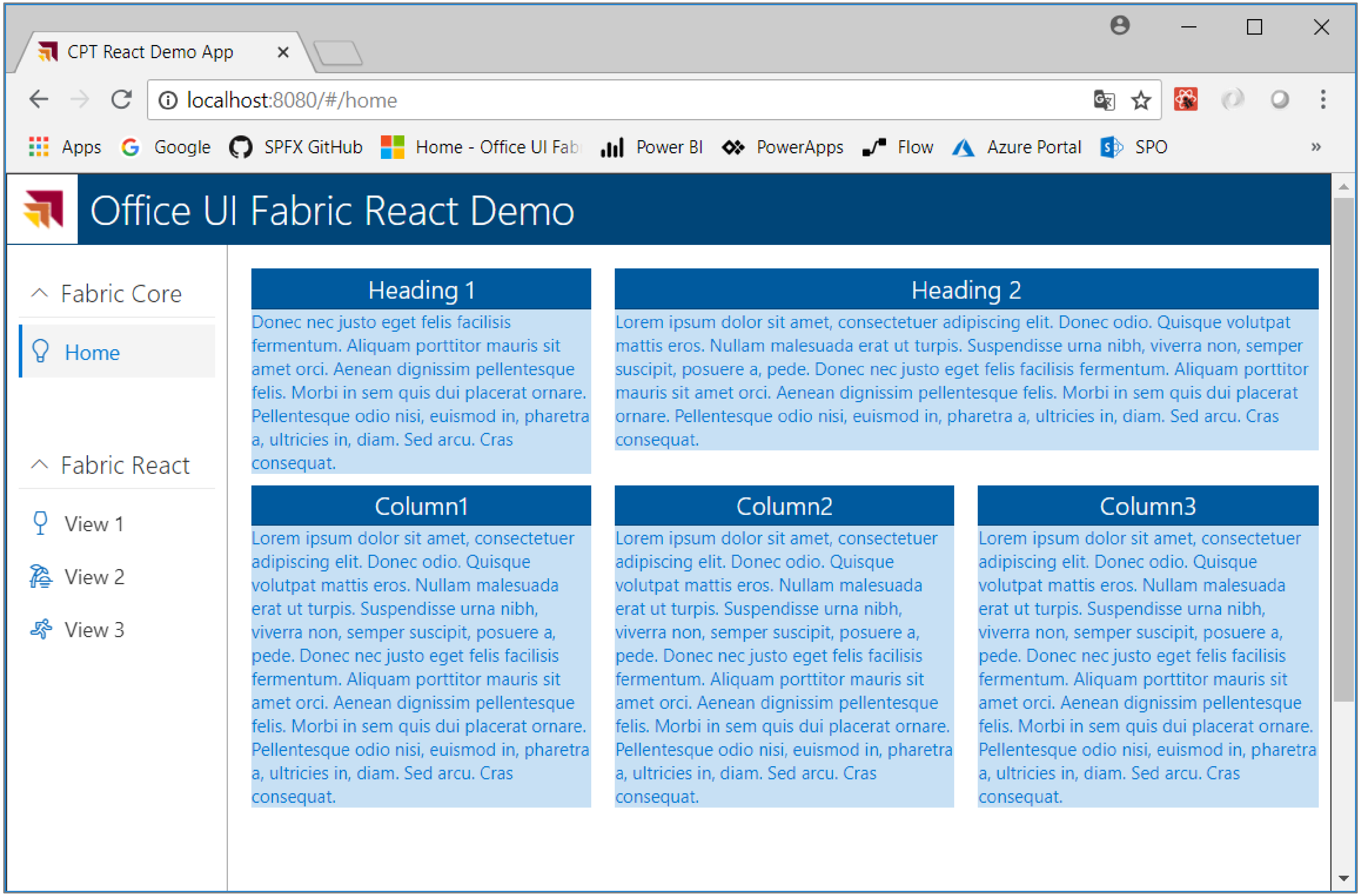
3 3 3 3

4 4 4

6 6

12

Demo 3 - Office UI Fabric



Office UI Fabric React Demo

Heading 1

Donec nec justo eget felis facilisis fermentum. Aliquam porttitor mauris sit amet orci. Aenean dignissim pellentesque felis. Morbi in sem quis dui placerat ornare. Pellentesque odio nisi, euismod in, pharetra a, ultricies in, diam. Sed arcu. Cras consequat.

Heading 2

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec odio. Quisque volutpat mattis eros. Nullam malesuada erat ut turpis. Suspendisse urna nibh, viverra non, semper suscipit, posuere a, pede. Donec nec justo eget felis facilisis fermentum. Aliquam porttitor mauris sit amet orci. Aenean dignissim pellentesque felis. Morbi in sem quis dui placerat ornare. Pellentesque odio nisi, euismod in, pharetra a, ultricies in, diam. Sed arcu. Cras consequat.

Column1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec odio. Quisque volutpat mattis eros. Nullam malesuada erat ut turpis. Suspendisse urna nibh, viverra non, semper suscipit, posuere a, pede. Donec nec justo eget felis facilisis fermentum. Aliquam porttitor mauris sit amet orci. Aenean dignissim pellentesque felis. Morbi in sem quis dui placerat ornare. Pellentesque odio nisi, euismod in, pharetra a, ultricies in, diam. Sed arcu. Cras consequat.

Column2

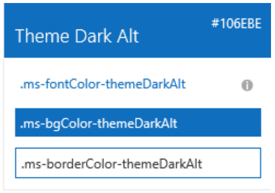
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec odio. Quisque volutpat mattis eros. Nullam malesuada erat ut turpis. Suspendisse urna nibh, viverra non, semper suscipit, posuere a, pede. Donec nec justo eget felis facilisis fermentum. Aliquam porttitor mauris sit amet orci. Aenean dignissim pellentesque felis. Morbi in sem quis dui placerat ornare. Pellentesque odio nisi, euismod in, pharetra a, ultricies in, diam. Sed arcu. Cras consequat.

Column3

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec odio. Quisque volutpat mattis eros. Nullam malesuada erat ut turpis. Suspendisse urna nibh, viverra non, semper suscipit, posuere a, pede. Donec nec justo eget felis facilisis fermentum. Aliquam porttitor mauris sit amet orci. Aenean dignissim pellentesque felis. Morbi in sem quis dui placerat ornare. Pellentesque odio nisi, euismod in, pharetra a, ultricies in, diam. Sed arcu. Cras consequat.

Fabric Core styling

- Fonts and typography
  - Segoe font family + type ramp
  - Official Office 365 iconography
- Color
  - Official Office 365 color palette
- Branded assets
  - Product symbols + product filetype symbols
- Animations
  - Official Office 365 selection of easings and animations
- Responsive grid
  - Tailored to Office 365 silhouettes


Theme Dark Alt #106EBE

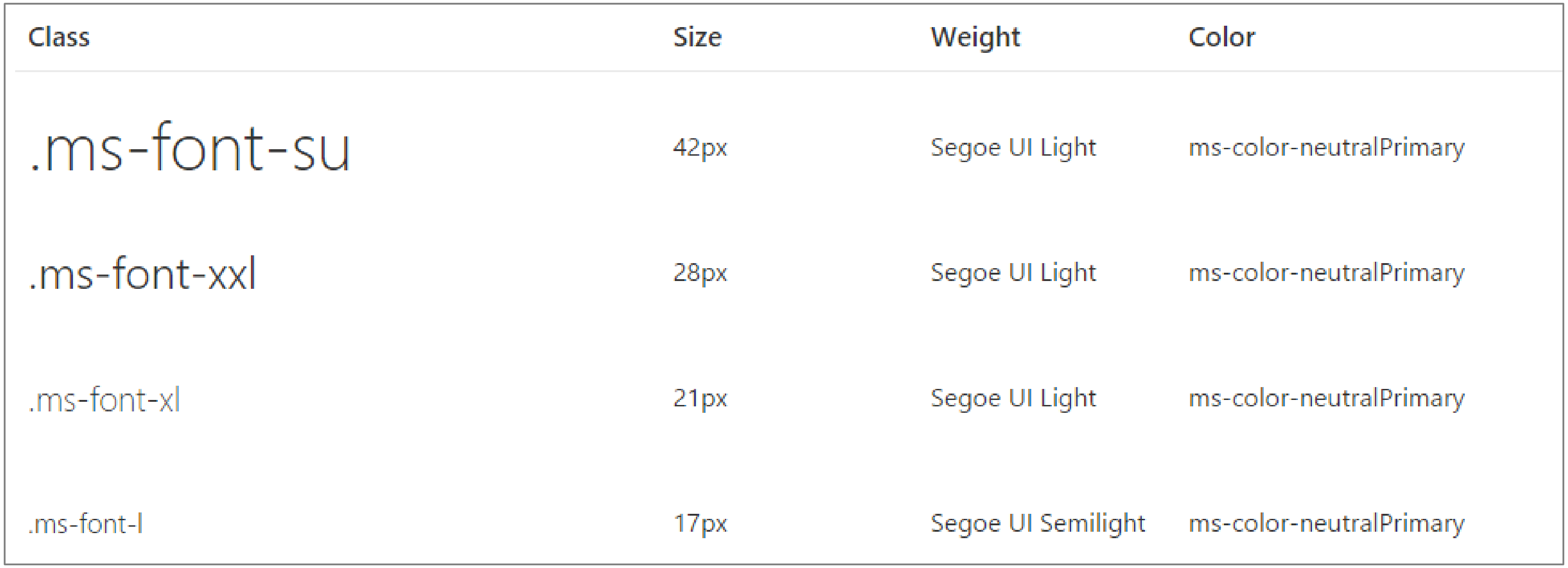
.ms-fontColor-themeDarkAlt

.ms-bgColor-themeDarkAlt

.ms-borderColor-themeDarkAlt

Typography

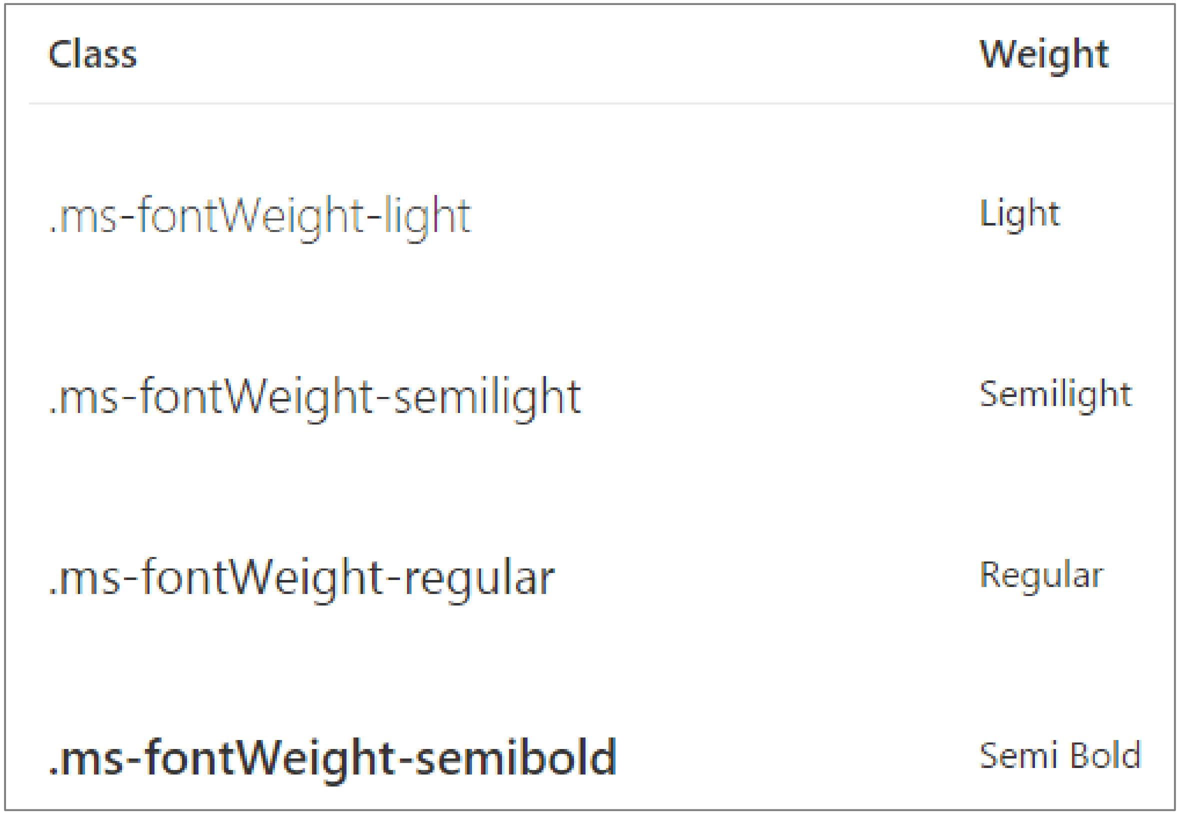
- Base font classes
  - Fabric includes 10 base font classes
  - Each base class sets a default size, weight, and color.



Class	Size	Weight	Color
.ms-font-su	42px	Segoe UI Light	ms-color-neutralPrimary
.ms-font-xxl	28px	Segoe UI Light	ms-color-neutralPrimary
.ms-font-xl	21px	Segoe UI Light	ms-color-neutralPrimary
.ms-font-l	17px	Segoe UI Semilight	ms-color-neutralPrimary

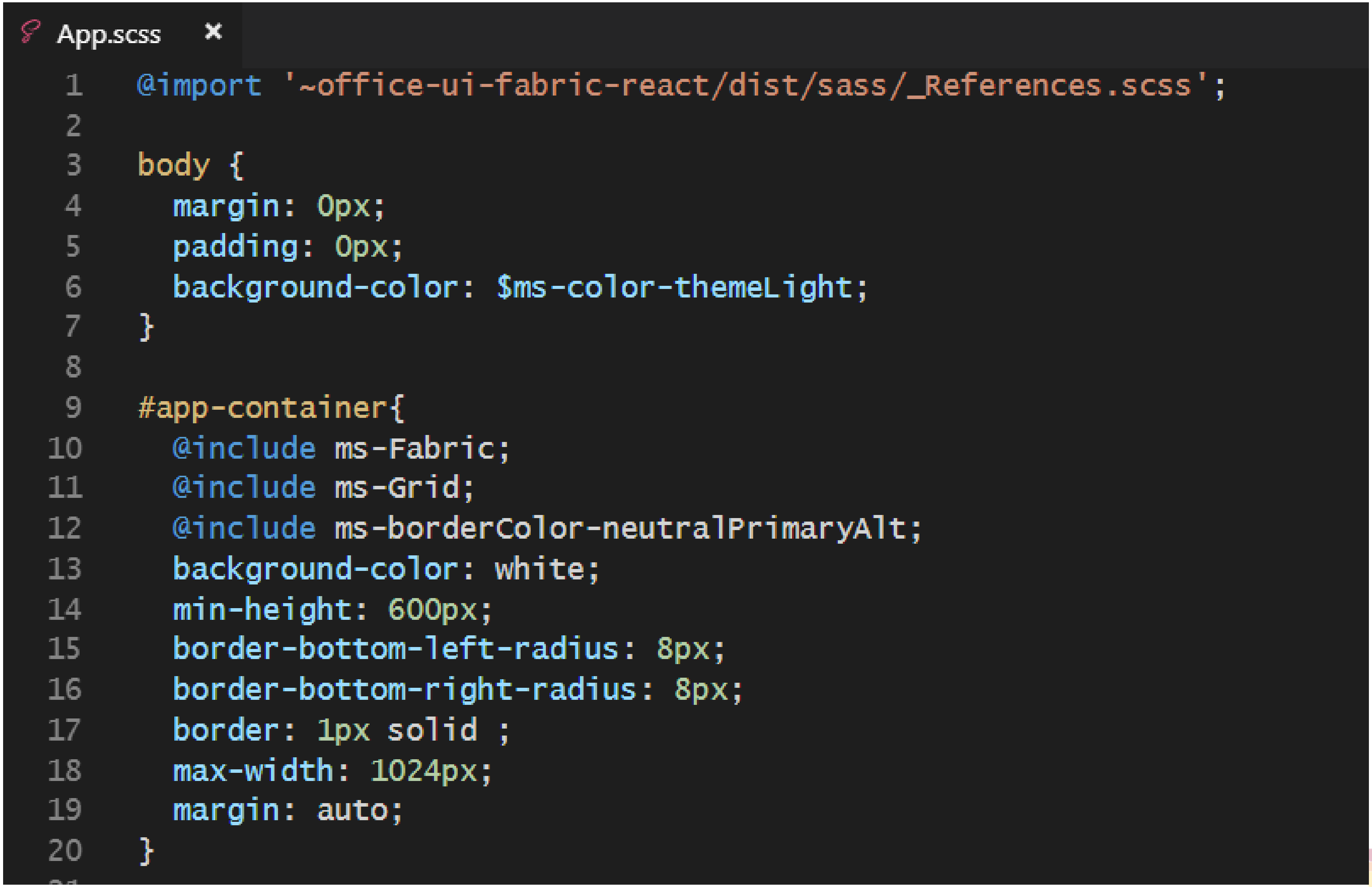
Typography

- Helper font classes
  - There are helper font classes to change the text weight.



Class	Weight
.ms-fontWeight-light	Light
.ms-fontWeight-semilight	Semilight
.ms-fontWeight-regular	Regular
.ms-fontWeight-semibold	Semi Bold

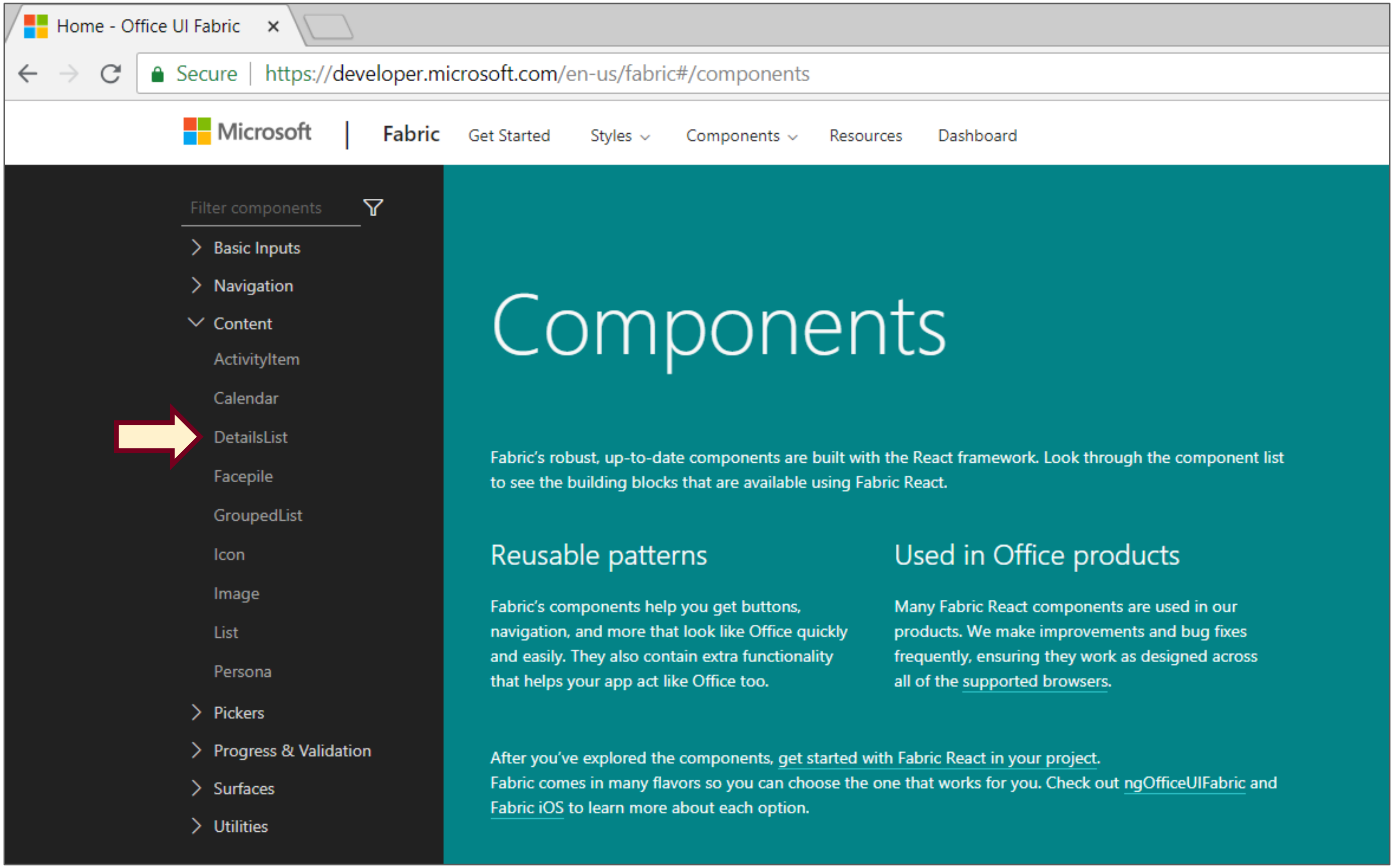
Putting Office UI Fabric Styles to Use



```scss
@import '~office-ui-fabric-react/dist/sass/_References.scss';

body {
  margin: 0px;
  padding: 0px;
  background-color: $ms-color-themeLight;
}

#app-container{
  @include ms-Fabric;
  @include ms-Grid;
  @include ms-borderColor-neutralPrimaryAlt;
  background-color: white;
  min-height: 600px;
  border-bottom-left-radius: 8px;
  border-bottom-right-radius: 8px;
  border: 1px solid ;
  max-width: 1024px;
  margin: auto;
}
```

Office UI Fabric React Component Library

- https://developer.microsoft.com/en-us/fabric#/components



Components

Fabric's robust, up-to-date components are built with the React framework. Look through the component list to see the building blocks that are available using Fabric React.

Reusable patterns

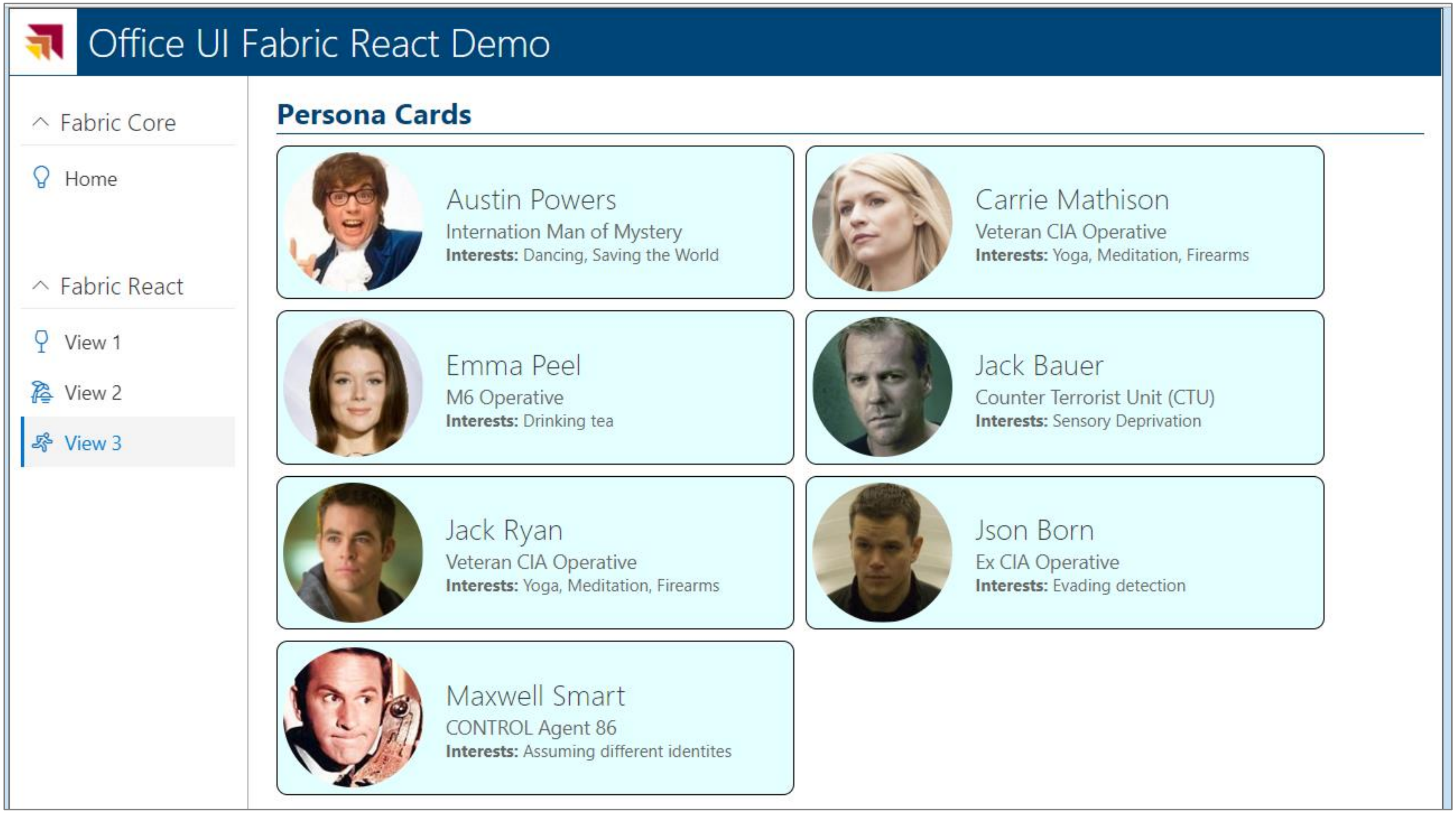
Fabric's components help you get buttons, navigation, and more that look like Office quickly and easily. They also contain extra functionality that helps your app act like Office too.

Used in Office products

Many Fabric React components are used in our products. We make improvements and bug fixes frequently, ensuring they work as designed across all of the supported browsers.

After you've explored the components, get started with Fabric React in your project.
Fabric comes in many flavors so you can choose the one that works for you. Check out ngOfficeUIFabric and Fabric iOS to learn more about each option.

Using Persona Cards



Office UI Fabric React Demo

Persona Cards

Austin Powers
Internation Man of Mystery
Interests: Dancing, Saving the World

Carrie Mathison
Veteran CIA Operative
Interests: Yoga, Meditation, Firearms

Emma Peel
M6 Operative
Interests: Drinking tea

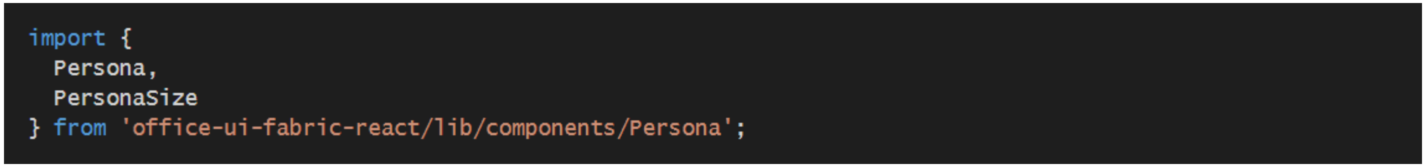
Jack Bauer
Counter Terrorist Unit (CTU)
Interests: Sensory Deprivation

Jack Ryan
Veteran CIA Operative
Interests: Yoga, Meditation, Firearms

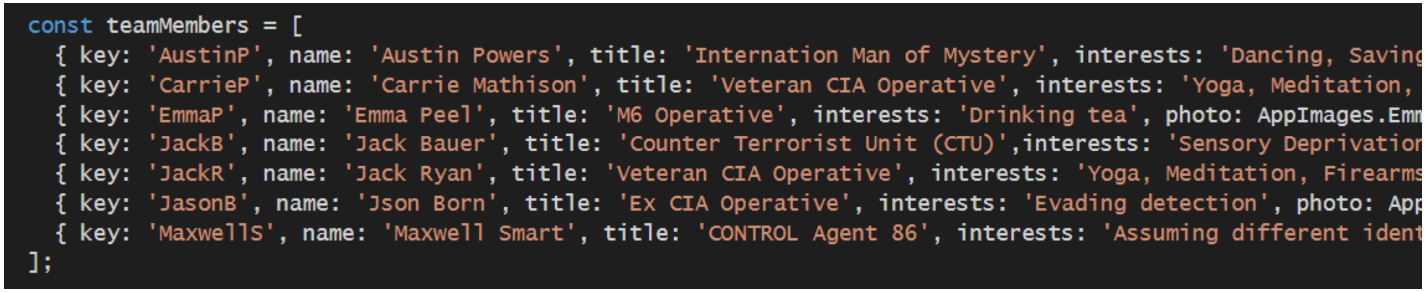
Json Born
Ex CIA Operative
Interests: Evading detection

Maxwell Smart
CONTROL Agent 86
Interests: Assuming different identites

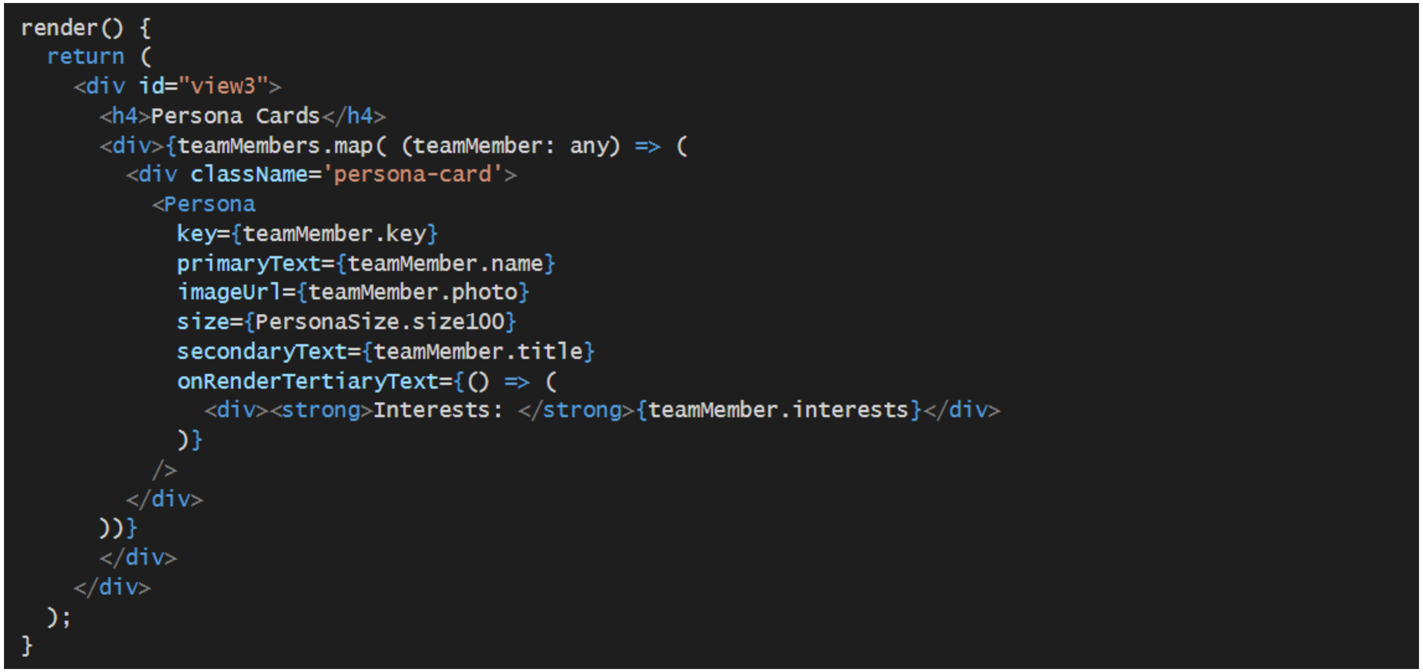
Adding a Persona Component



```
import {
  Persona,
  PersonaSize
} from 'office-ui-fabric-react/lib/components/Persona';
```

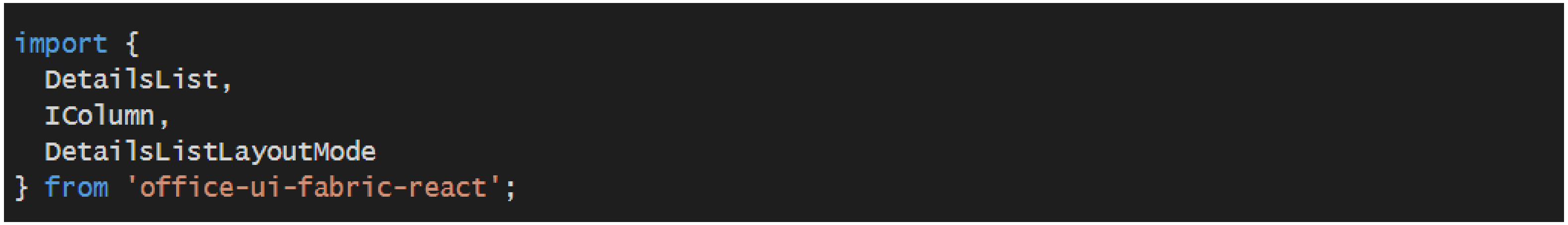


```
const teamMembers = [
  { key: 'AustinP', name: 'Austin Powers', title: 'Internation Man of Mystery', interests: 'Dancing, Saving
  { key: 'CarrieP', name: 'Carrie Mathison', title: 'Veteran CIA Operative', interests: 'Yoga, Meditation,
  { key: 'EmmaP', name: 'Emma Peel', title: 'M6 Operative', interests: 'Drinking tea', photo: AppImages.Emm
  { key: 'JackB', name: 'Jack Bauer', title: 'Counter Terrorist Unit (CTU)',interests: 'Sensory Deprivatio
  { key: 'JackR', name: 'Jack Ryan', title: 'Veteran CIA Operative', interests: 'Yoga, Meditation, Firearms
  { key: 'JasonB', name: 'Json Born', title: 'Ex CIA Operative', interests: 'Evading detection', photo: App
  { key: 'MaxwellS', name: 'Maxwell Smart', title: 'CONTROL Agent 86', interests: 'Assuming different ident
];
```

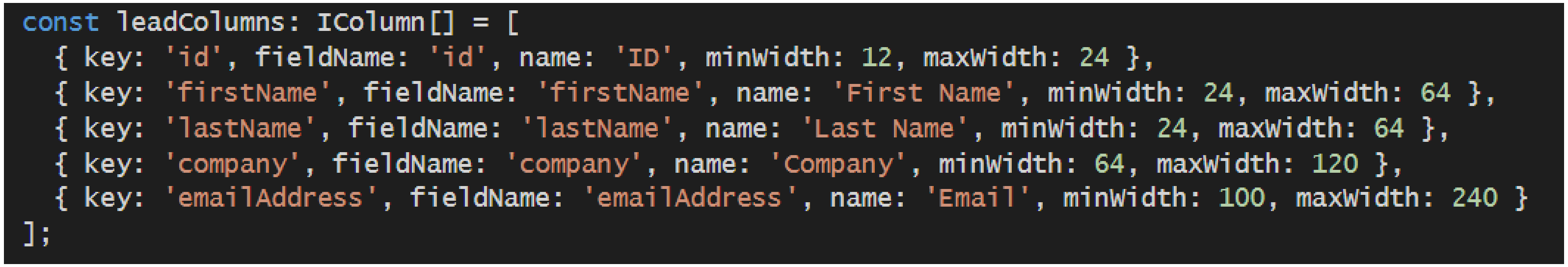


```
render() {
  return (
    <div id="view3">
      <h4>Persona Cards</h4>
      <div>{teamMembers.map( (teamMember: any) => (
        <div className='persona-card'>
          <Persona
            key={teamMember.key}
            primaryText={teamMember.name}
            imageUrl={teamMember.photo}
            size={PersonaSize.size100}
            secondaryText={teamMember.title}
            onRenderTertiaryText={() => (
              <div><strong>Interests: </strong>{teamMember.interests}</div>
            )}
          />
        </div>
      ))}
      </div>
    </div>
  );
}
```

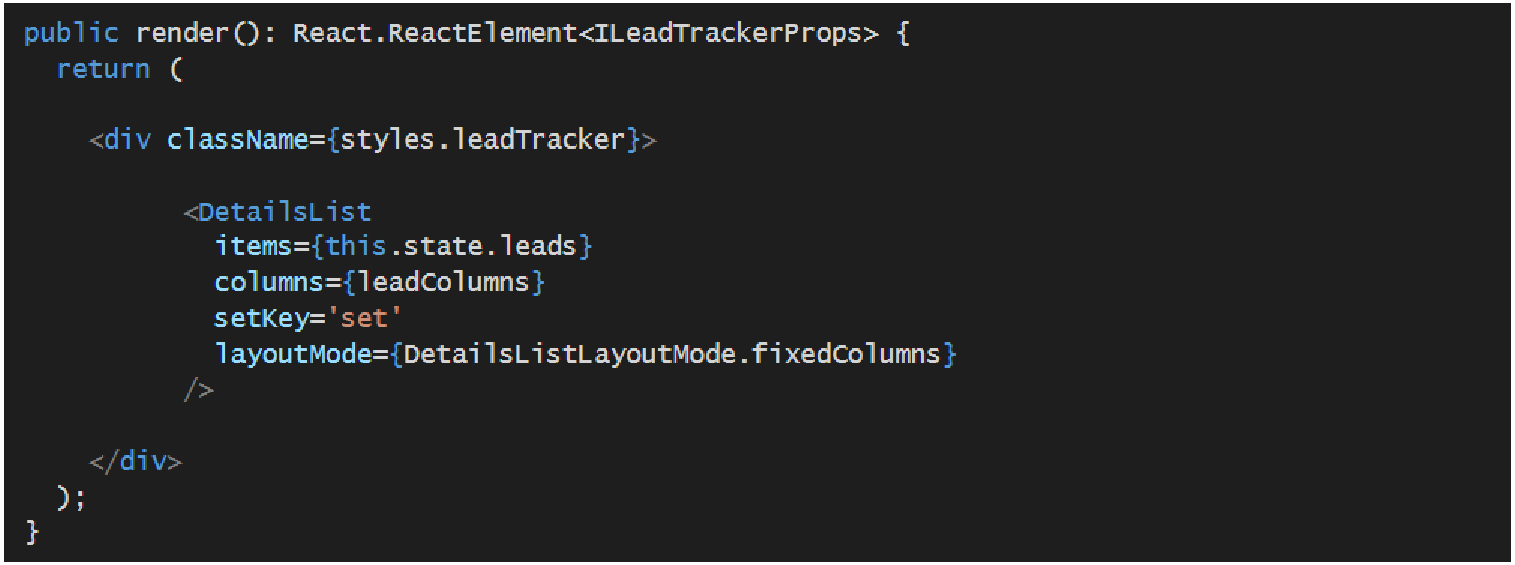
Using the DetailsList Component



```
import {
  DetailsList,
  IColumn,
  DetailsListLayoutMode
} from 'office-ui-fabric-react';
```

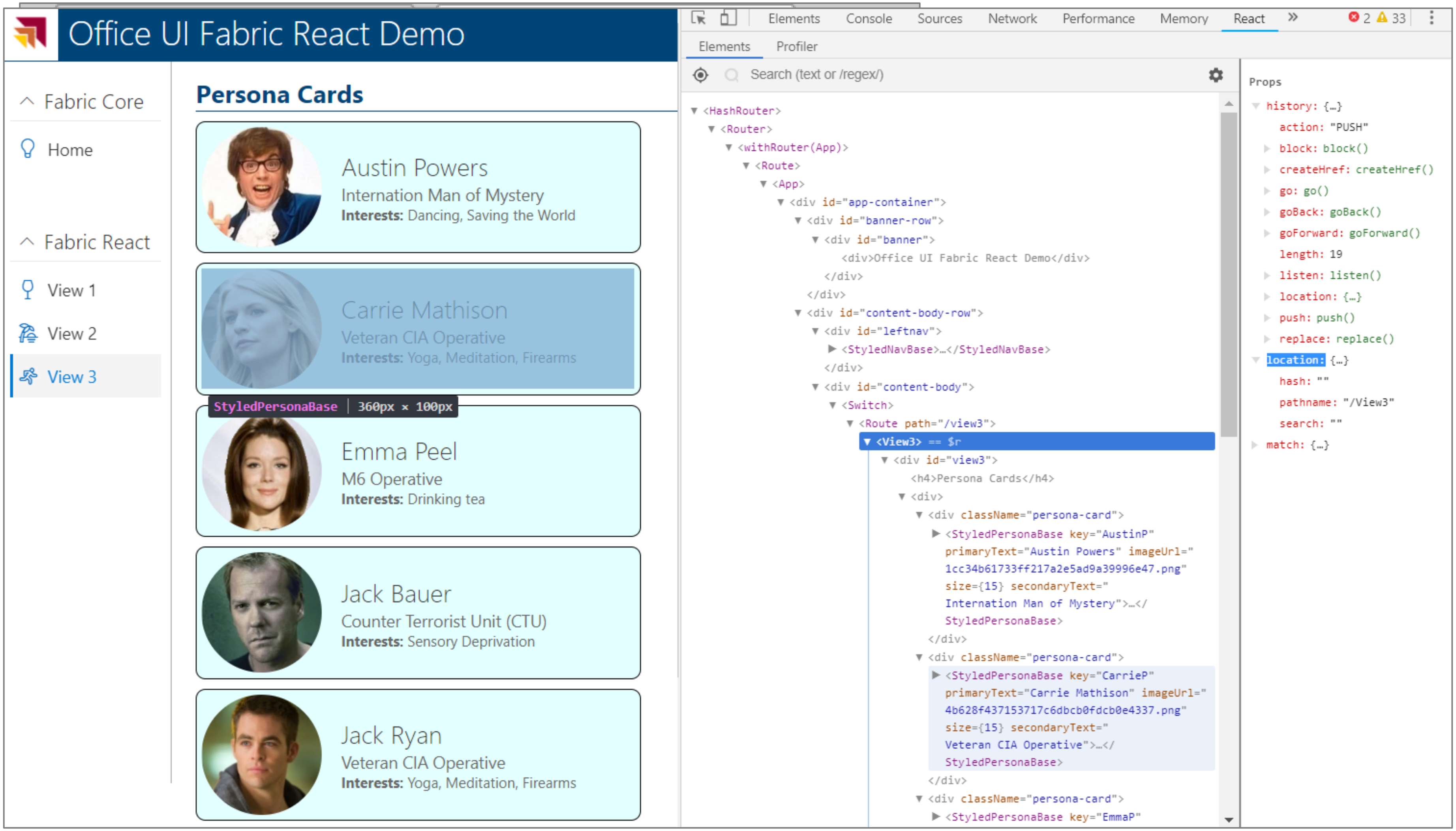


```
const leadColumns: IColumn[] = [
  { key: 'id', fieldName: 'id', name: 'ID', minWidth: 12, maxWidth: 24 },
  { key: 'firstName', fieldName: 'firstName', name: 'First Name', minWidth: 24, maxWidth: 64 },
  { key: 'lastName', fieldName: 'lastName', name: 'Last Name', minWidth: 24, maxWidth: 64 },
  { key: 'company', fieldName: 'company', name: 'Company', minWidth: 64, maxWidth: 120 },
  { key: 'emailAddress', fieldName: 'emailAddress', name: 'Email', minWidth: 100, maxWidth: 240 }
];
```



```
public render(): React.ReactElement<ILeadTrackerProps> {
  return (

    <div className={styles.leadTracker}>

          <DetailsList
            items={this.state.leads}
            columns={leadColumns}
            setKey='set'
            layoutMode={DetailsListLayoutMode.fixedColumns}
          />

    </div>
  );
}
```

React Developer Tools - Chrome Extension



Office UI Fabric React Demo

Fabric Core

Home

Fabric React

View 1

View 2

View 3

Persona Cards

Austin Powers
Internation Man of Mystery
Interests: Dancing, Saving the World

Carrie Mathison
Veteran CIA Operative
Interests: Yoga, Meditation, Firearms

Emma Peel
M6 Operative
Interests: Drinking tea

Jack Bauer
Counter Terrorist Unit (CTU)
Interests: Sensory Deprivation

Jack Ryan
Veteran CIA Operative
Interests: Yoga, Meditation, Firearms

```
▼ <HashRouter>
  ▼ <Router>
    ▼ <withRouter(App)>
      ▼ <Route>
        ▼ <App>
          ▼ <div id="app-container">
            ▼ <div id="banner-row">
              ▼ <div id="banner">
                  <div>Office UI Fabric React Demo</div>
                </div>
              </div>
            ▼ <div id="content-body-row">
              ▼ <div id="leftnav">
                ▶ <StyledNavBase>…</StyledNavBase>
                </div>
              ▼ <div id="content-body">
                ▼ <Switch>
                  ▼ <Route path="/view3">
                    ▼ <View3> == $r
                      ▼ <div id="view3">
                          <h4>Persona Cards</h4>
                        ▼ <div>
                          ▼ <div className="persona-card">
                            ▶ <StyledPersonaBase key="AustinP"
                              primaryText="Austin Powers" imageUrl="
                              1cc34b61733ff217a2e5ad9a39996e47.png"
                              size={15} secondaryText="
                              Internation Man of Mystery">…</
                              StyledPersonaBase>
                            </div>
                          ▼ <div className="persona-card">
                            ▶ <StyledPersonaBase key="CarrieP"
                              primaryText="Carrie Mathison" imageUrl="
                              4b628f437153717c6dbcb0fdcb0e4337.png"
                              size={15} secondaryText="
                              Veteran CIA Operative">…</
                              StyledPersonaBase>
                            </div>
                          ▼ <div className="persona-card">
                            ▶ <StyledPersonaBase key="EmmaP"
```

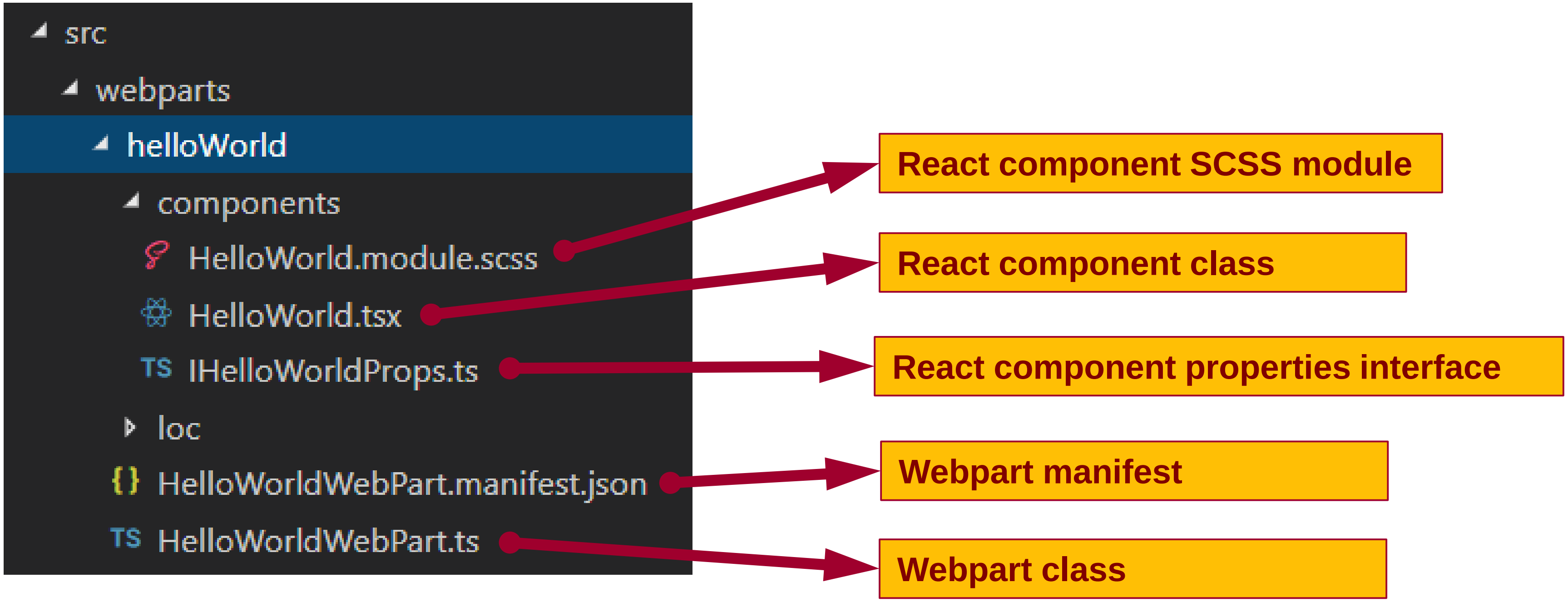
```
Props
▼ history: {…}
    action: "PUSH"
  ▶ block: block()
  ▶ createHref: createHref()
  ▶ go: go()
  ▶ goBack: goBack()
  ▶ goForward: goForward()
    length: 19
  ▶ listen: listen()
  ▶ location: {…}
  ▶ push: push()
  ▶ replace: replace()
▼ location: {…}
    hash: ""
    pathname: "/View3"
    search: ""
▶ match: {…}
```

Agenda

- ✓ Developing with Node.js, TypeScript and Webpack
- ✓ Learning React.js Fundamentals
- ✓ Using the Office UI Fabric React Component Library
- ➢ Developing React Webparts with SharePoint Framework
- • Calling the Microsoft Graph API from React Webparts

Creating a React Webpart

- You can select React as framework for your webpart
  - You can create a React webpart when creating new project
  - You can add React webpart to existing project
  - React webpart made up of several different source files



```
◢ src
  ◢ webparts
    ◢ helloWorld
      ◢ components
          HelloWorld.module.scss
          HelloWorld.tsx
        TS IHelloWorldProps.ts
      ▷ loc
      {} HelloWorldWebPart.manifest.json
      TS HelloWorldWebPart.ts
```

React component SCSS module

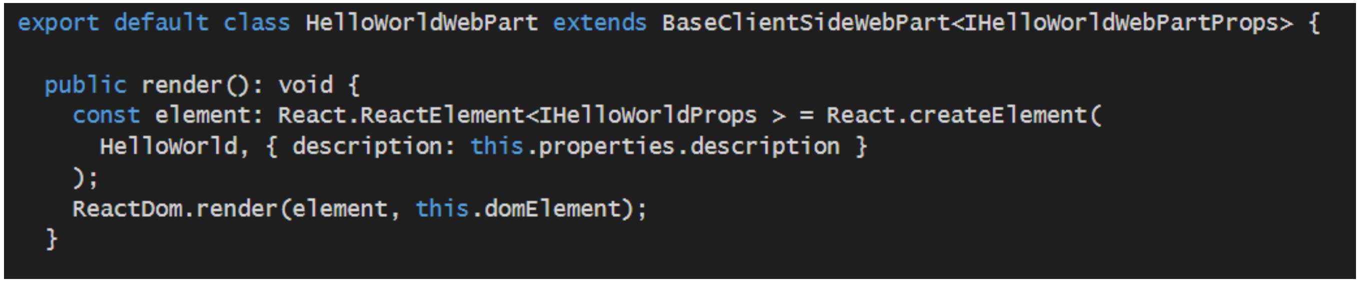
React component class

React component properties interface

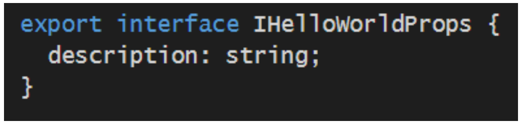
Webpart manifest

Webpart class

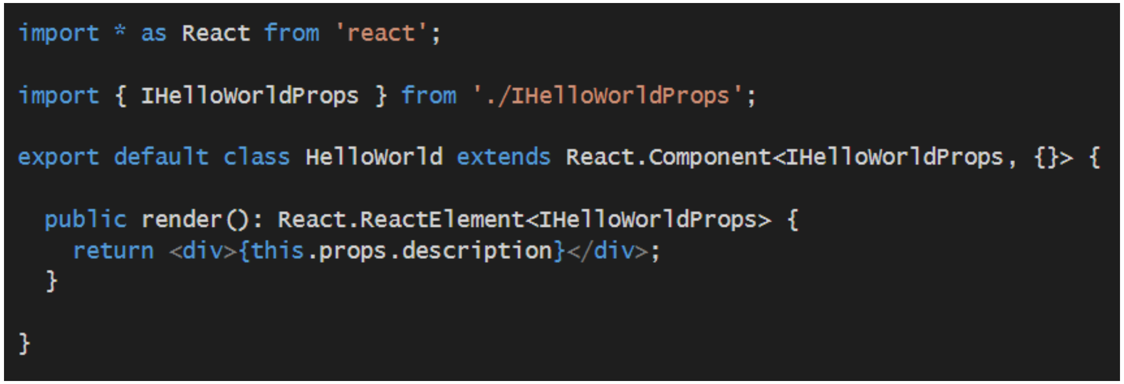
React Webpart Architecture



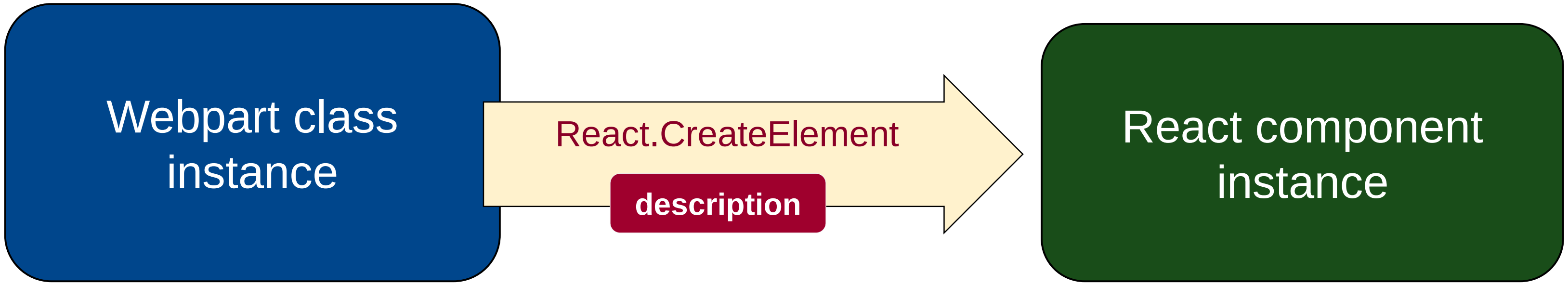
```
export default class HelloWorldWebPart extends BaseClientSideWebPart<IHelloWorldWebPartProps> {

  public render(): void {
    const element: React.ReactElement<IHelloWorldProps > = React.createElement(
      HelloWorld, { description: this.properties.description }
    );
    ReactDom.render(element, this.domElement);
  }
```



```
export interface IHelloWorldProps {
  description: string;
}
```



```
import * as React from 'react';

import { IHelloWorldProps } from './IHelloWorldProps';

export default class HelloWorld extends React.Component<IHelloWorldProps, {}> {

  public render(): React.ReactElement<IHelloWorldProps> {
    return <div>{this.props.description}</div>;
  }

}
```



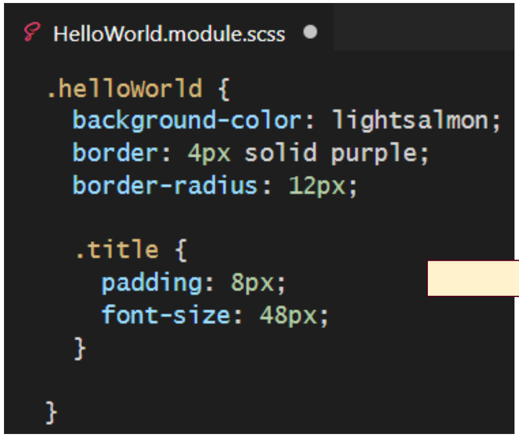
Webpart class instance

React.CreateElement

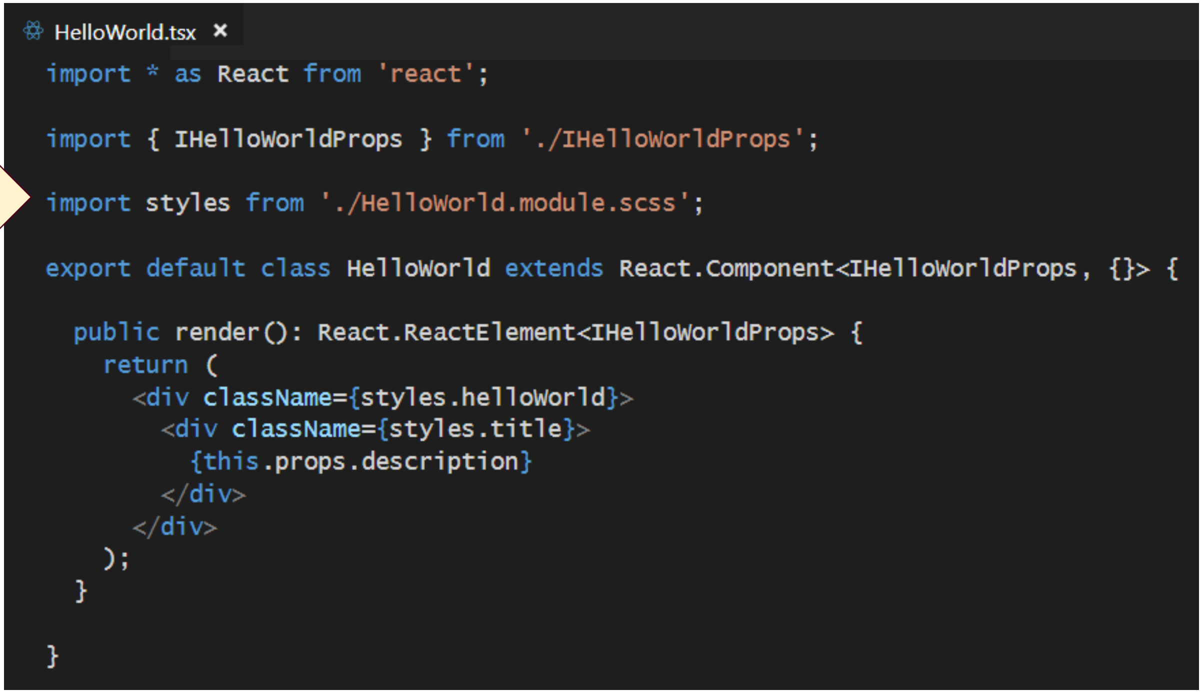
description

React component instance

React Webpart Styling



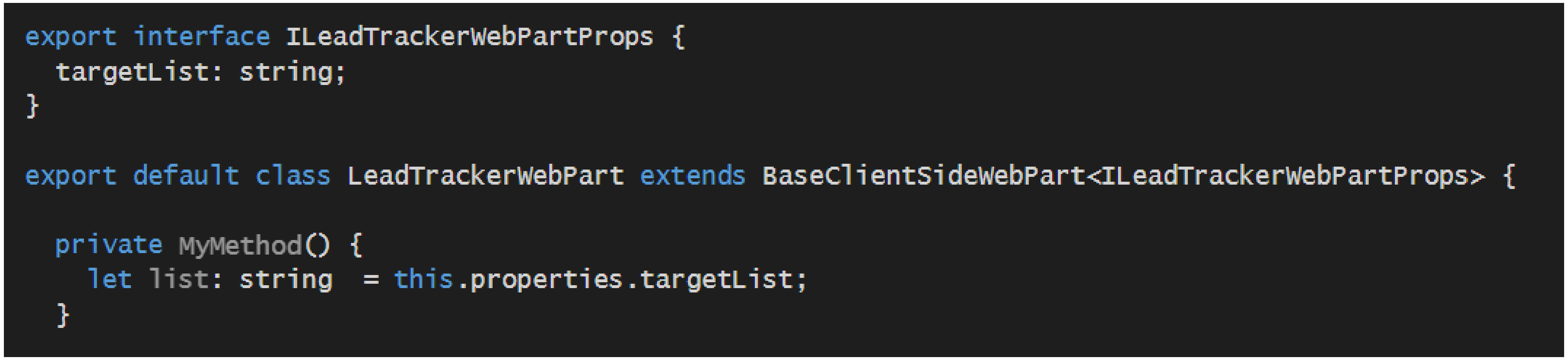
```scss
.helloWorld {
  background-color: lightsalmon;
  border: 4px solid purple;
  border-radius: 12px;

  .title {
    padding: 8px;
    font-size: 48px;
  }

}
```



```tsx
import * as React from 'react';

import { IHelloWorldProps } from './IHelloWorldProps';

import styles from './HelloWorld.module.scss';

export default class HelloWorld extends React.Component<IHelloWorldProps, {}> {

  public render(): React.ReactElement<IHelloWorldProps> {
    return (
      <div className={styles.helloWorld}>
        <div className={styles.title}>
          {this.props.description}
        </div>
      </div>
    );
  }

}
```

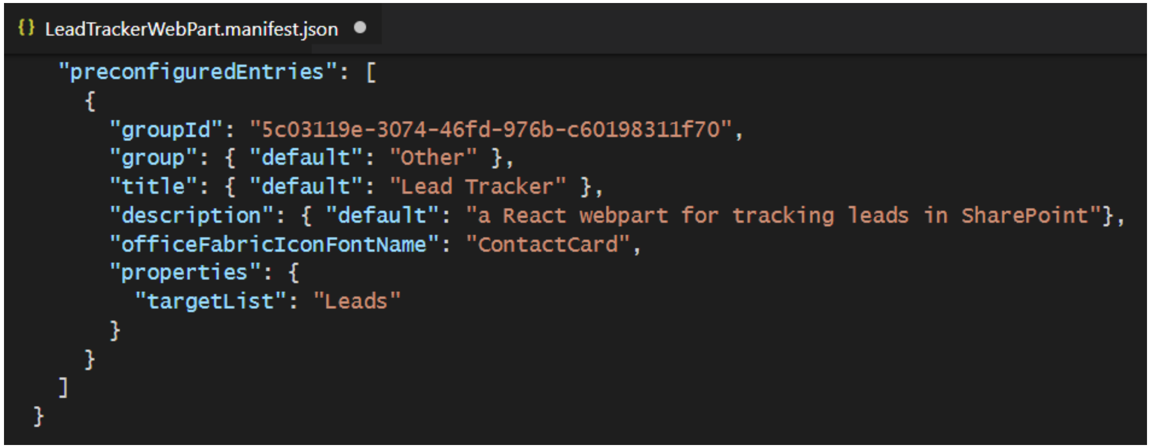
Webpart Persistent Properties

- Persistent properties defined in webpart using interface

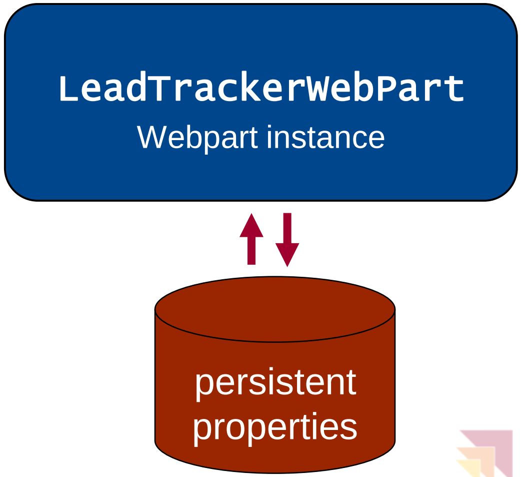


```
export interface ILeadTrackerWebPartProps {
  targetList: string;
}

export default class LeadTrackerWebPart extends BaseClientSideWebPart<ILeadTrackerWebPartProps> {

  private MyMethod() {
    let list: string  = this.properties.targetList;
  }
```

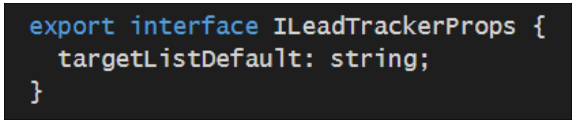
- Property default values add to webpart manifest



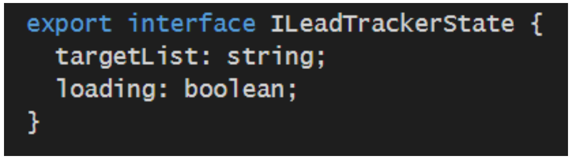
```
  "preconfiguredEntries": [
    {
      "groupId": "5c03119e-3074-46fd-976b-c60198311f70",
      "group": { "default": "Other" },
      "title": { "default": "Lead Tracker" },
      "description": { "default": "a React webpart for tracking leads in SharePoint"},
      "officeFabricIconFontName": "ContactCard",
      "properties": {
        "targetList": "Leads"
      }
    }
  ]
}
```



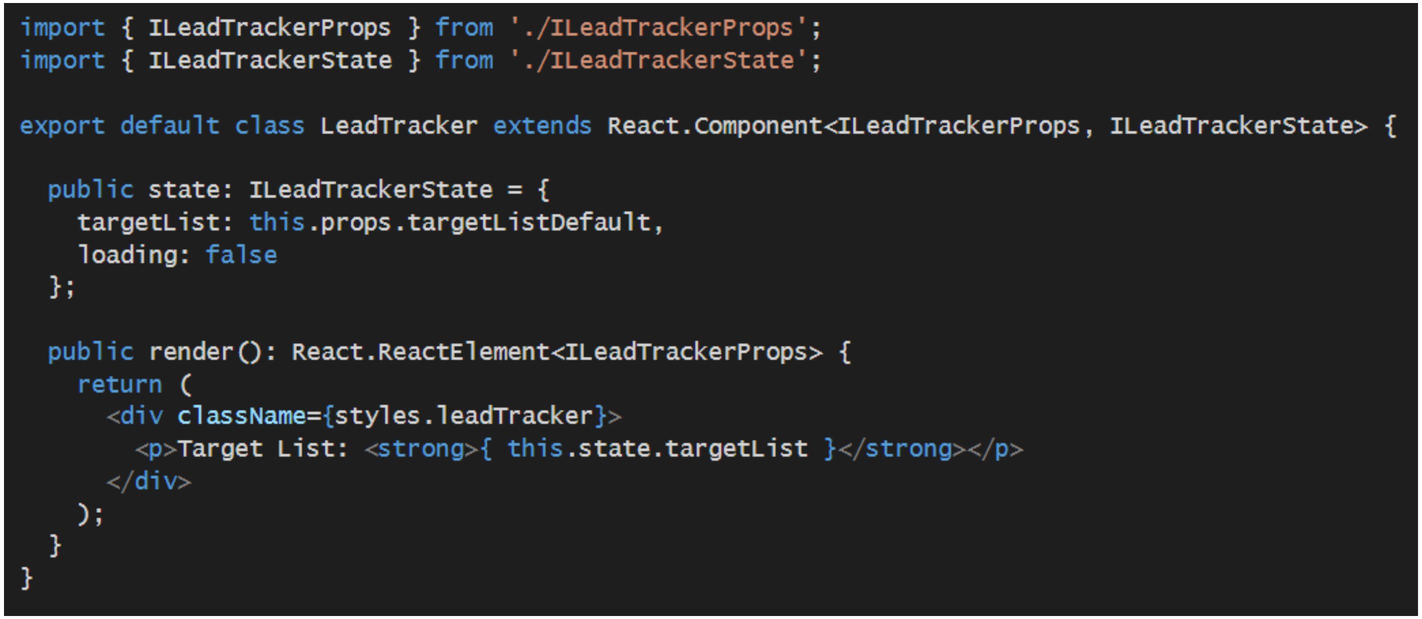
Designing the React Component



```
export interface ILeadTrackerProps {
  targetListDefault: string;
}
```



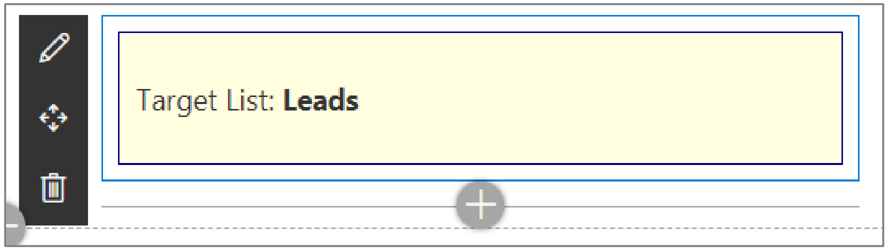
```
export interface ILeadTrackerState {
  targetList: string;
  loading: boolean;
}
```



```
import { ILeadTrackerProps } from './ILeadTrackerProps';
import { ILeadTrackerState } from './ILeadTrackerState';

export default class LeadTracker extends React.Component<ILeadTrackerProps, ILeadTrackerState> {

  public state: ILeadTrackerState = {
    targetList: this.props.targetListDefault,
    loading: false
  };

  public render(): React.ReactElement<ILeadTrackerProps> {
    return (
      <div className={styles.leadTracker}>
        <p>Target List: <strong>{ this.state.targetList }</strong></p>
      </div>
    );
  }
}
```

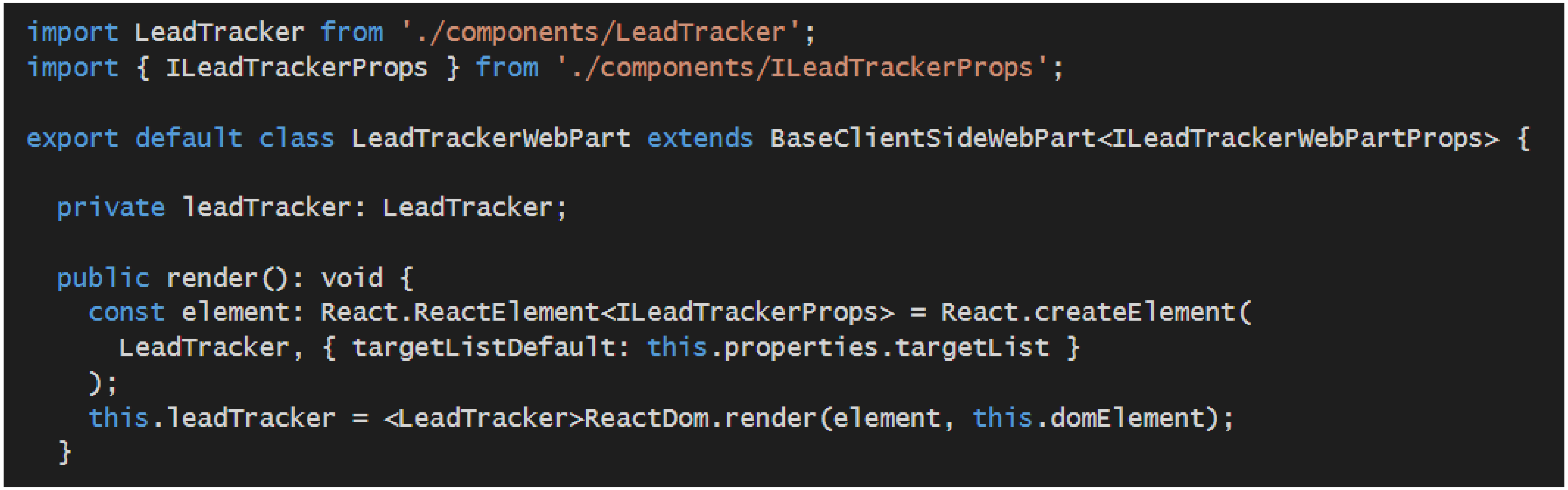
LeadTracker

React component

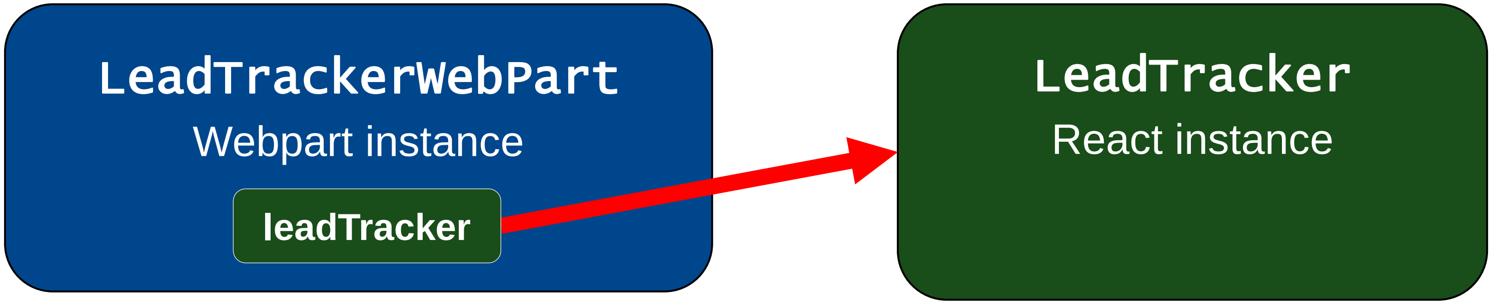


Target List: **Leads**

Referencing the React Component Instance



```
import LeadTracker from './components/LeadTracker';
import { ILeadTrackerProps } from './components/ILeadTrackerProps';

export default class LeadTrackerWebPart extends BaseClientSideWebPart<ILeadTrackerWebPartProps> {

  private leadTracker: LeadTracker;

  public render(): void {
    const element: React.ReactElement<ILeadTrackerProps> = React.createElement(
      LeadTracker, { targetListDefault: this.properties.targetList }
    );
    this.leadTracker = <LeadTracker>ReactDom.render(element, this.domElement);
  }
```



LeadTrackerWebPart

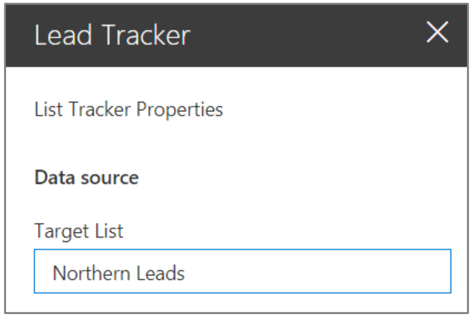
Webpart instance

leadTracker

LeadTracker

React instance

Synchronizing React State with Webpart Properties



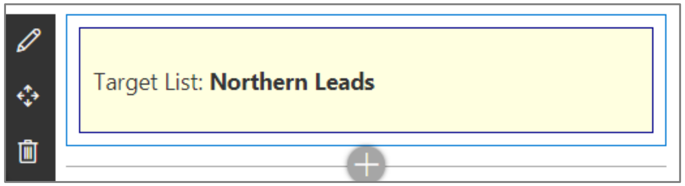
Lead Tracker

List Tracker Properties

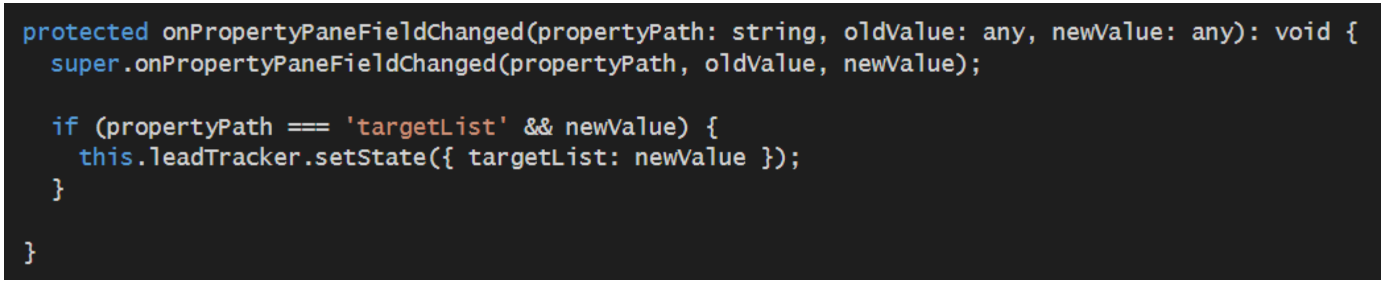
Data source

Target List: Northern Leads

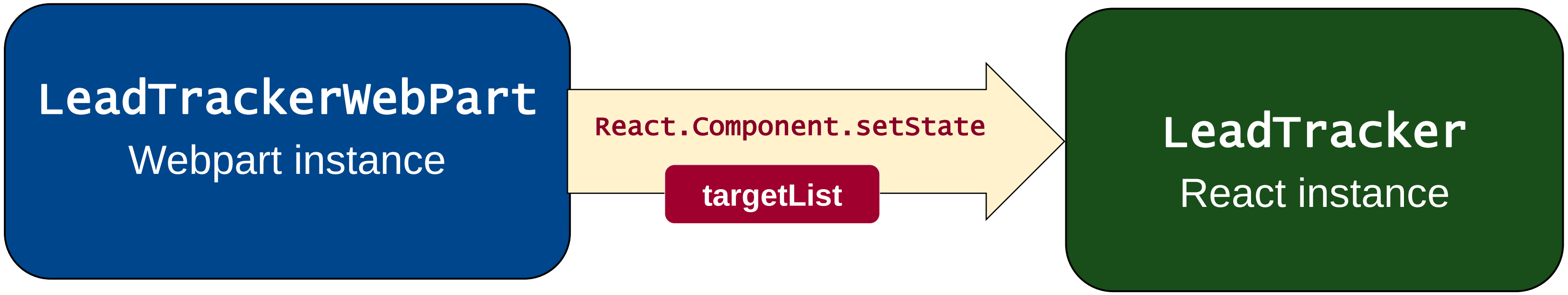
Automatic updates



Target List: **Northern Leads**



```
protected onPropertyPaneFieldChanged(propertyPath: string, oldValue: any, newValue: any): void {
  super.onPropertyPaneFieldChanged(propertyPath, oldValue, newValue);

  if (propertyPath === 'targetList' && newValue) {
    this.leadTracker.setState({ targetList: newValue });
  }

}
```



`LeadTrackerWebPart`

Webpart instance

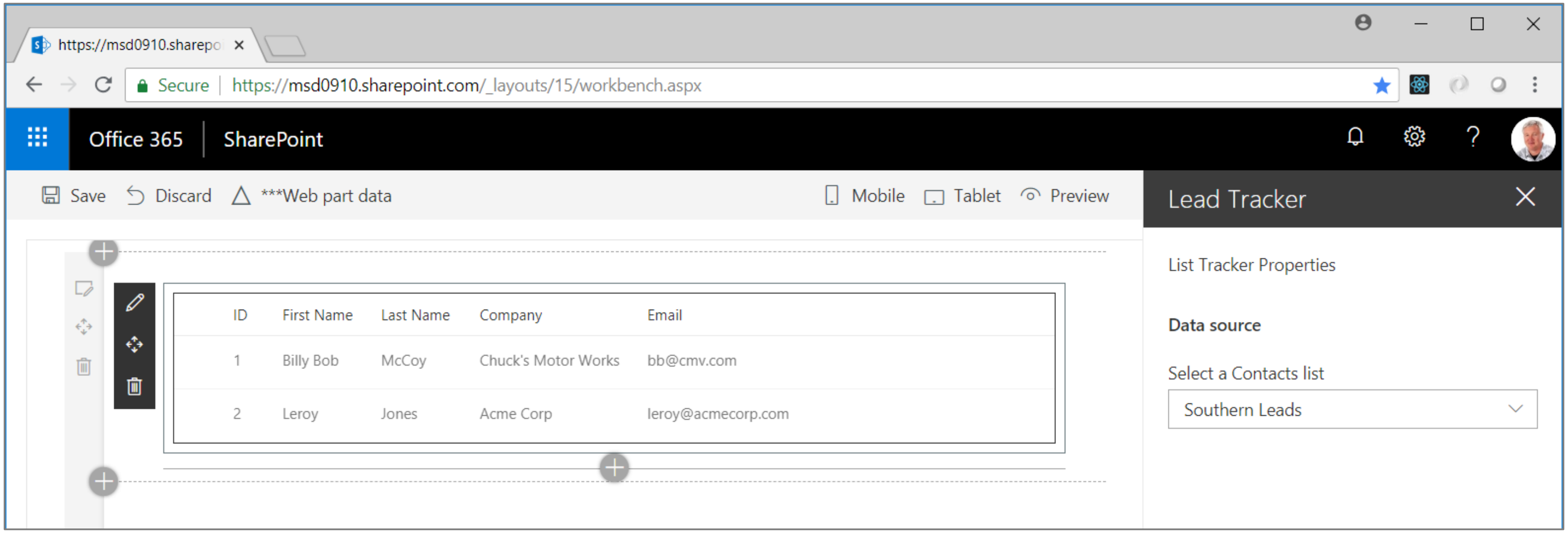
`React.Component.setState`

targetList

`LeadTracker`

React instance

Demo 4 - The Lead Tracker React Webpart



https://msd0910.sharepoint.com/_layouts/15/workbench.aspx

ID	First Name	Last Name	Company	Email
1	Billy Bob	McCoy	Chuck's Motor Works	bb@cmv.com
2	Leroy	Jones	Acme Corp	leroy@acmecorp.com

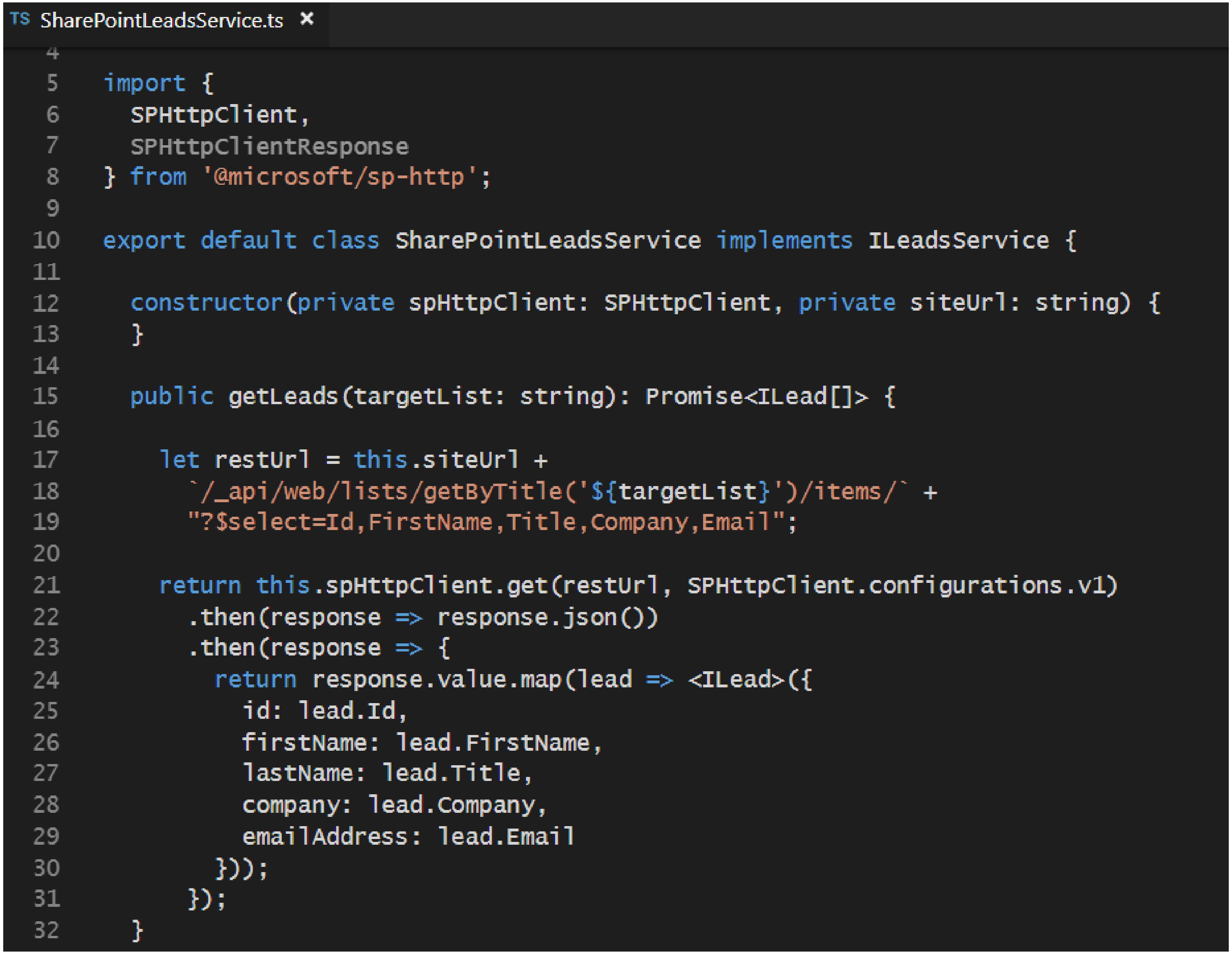
Lead Tracker

List Tracker Properties

Data source

Select a Contacts list: Southern Leads

Calling to the SharePoint Rest API



```typescript
import {
  SPHttpClient,
  SPHttpClientResponse
} from '@microsoft/sp-http';

export default class SharePointLeadsService implements ILeadsService {

  constructor(private spHttpClient: SPHttpClient, private siteUrl: string) {
  }

  public getLeads(targetList: string): Promise<ILead[]> {

    let restUrl = this.siteUrl +
      `/_api/web/lists/getByTitle('${targetList}')/items/` +
      "?$select=Id,FirstName,Title,Company,Email";

    return this.spHttpClient.get(restUrl, SPHttpClient.configurations.v1)
      .then(response => response.json())
      .then(response => {
        return response.value.map(lead => <ILead>({
          id: lead.Id,
          firstName: lead.FirstName,
          lastName: lead.Title,
          company: lead.Company,
          emailAddress: lead.Email
        }));
      });
  }
```

Agenda

- ✓ Developing with Node.js, TypeScript and Webpack
- ✓ Learning React.js Fundamentals
- ✓ Using the Office UI Fabric React Component Library
- ✓ Developing React Webparts with SharePoint Framework
- ➢ Calling the Microsoft Graph API from React Webparts

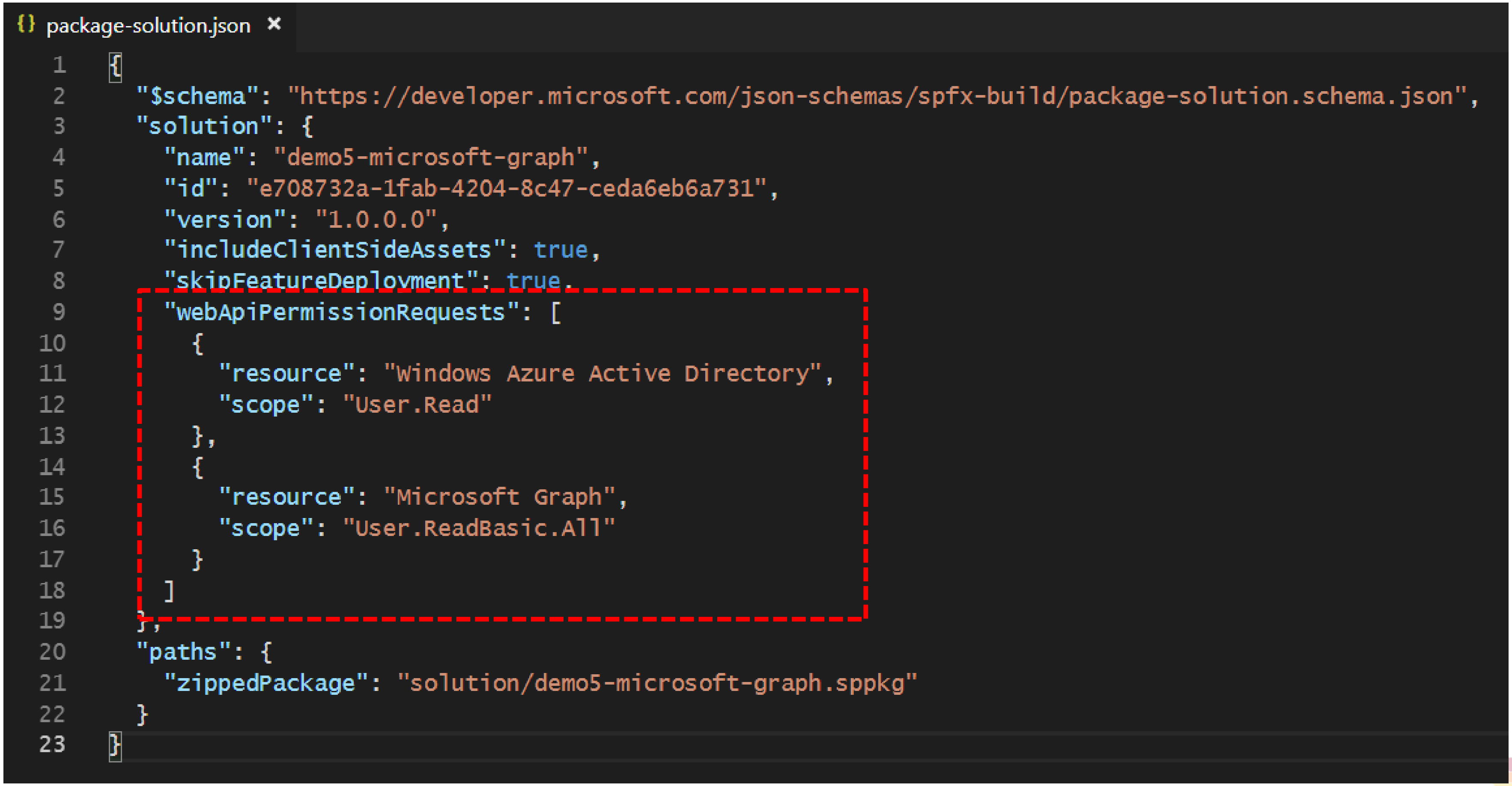
SPFx Includes Microsoft Graph Client

- SharePoint Online already has an Azure AD application
  - SPFX solutions can call SharePoint REST API in the same domain
  - No extra authentication required

- What about calling Microsoft Graph API across domains?
  - SPFX provides proxy to call Microsoft Graph API
  - MSGraphClient is the new Microsoft Graph Client for SPFx
  - Moved out of preview to release with API v1.6
  - Abstracts the token acquisition from the SPFx development

SPFx Solutions Declare Permission Requests

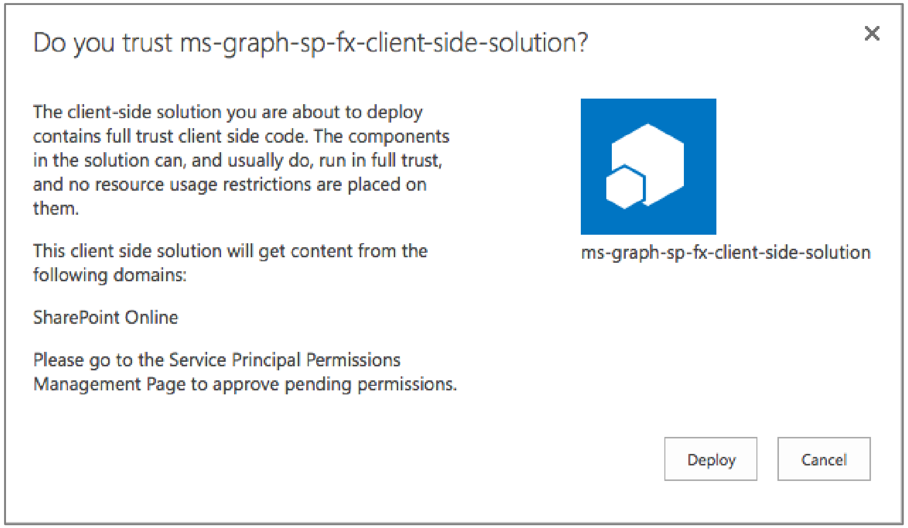
- Used to request tenant-wide permissions
- Permissions must be granted by tenant adminsitrator



```json
{
  "$schema": "https://developer.microsoft.com/json-schemas/spfx-build/package-solution.schema.json",
  "solution": {
    "name": "demo5-microsoft-graph",
    "id": "e708732a-1fab-4204-8c47-ceda6eb6a731",
    "version": "1.0.0.0",
    "includeClientSideAssets": true,
    "skipFeatureDeployment": true,
    "webApiPermissionRequests": [
      {
        "resource": "Windows Azure Active Directory",
        "scope": "User.Read"
      },
      {
        "resource": "Microsoft Graph",
        "scope": "User.ReadBasic.All"
      }
    ]
  },
  "paths": {
    "zippedPackage": "solution/demo5-microsoft-graph.sppkg"
  }
}
```

Add Package to SharePoint App Catalog

- Extra note in dialog notifies of additional step required
- While application can be installed in SharePoint sites, it does not have the permissions granted that it needs to access Azure AD protected resources



Do you trust ms-graph-sp-fx-client-side-solution?

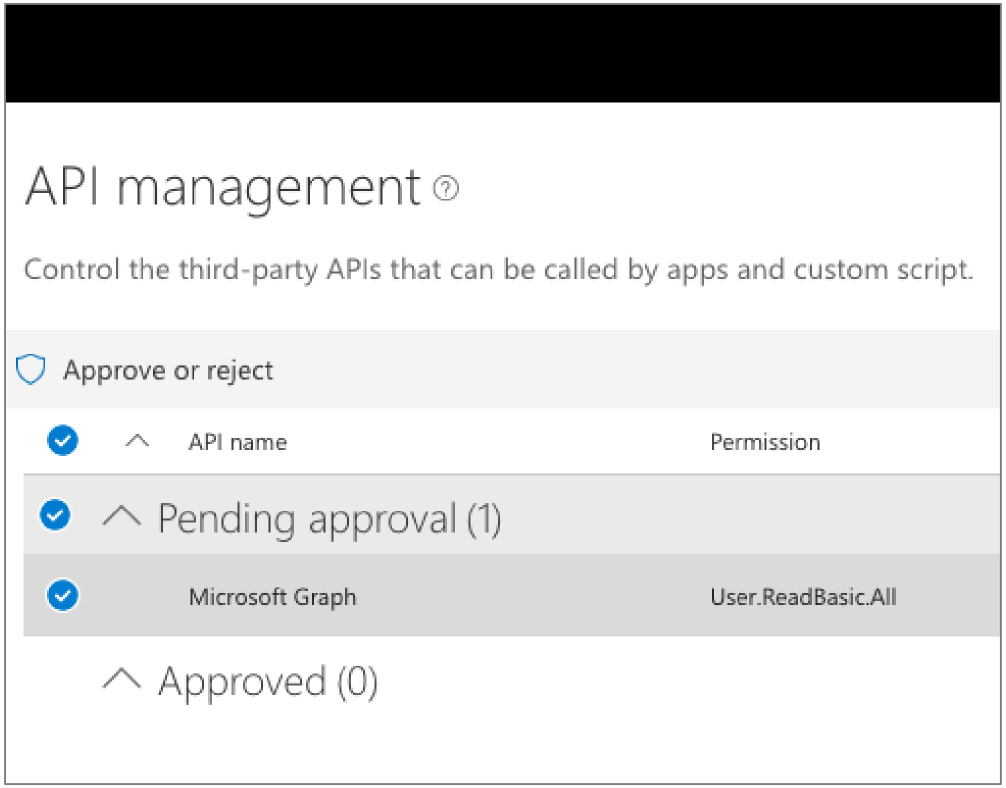
The client-side solution you are about to deploy contains full trust client side code. The components in the solution can, and usually do, run in full trust, and no resource usage restrictions are placed on them.

This client side solution will get content from the following domains:

SharePoint Online

Please go to the Service Principal Permissions Management Page to approve pending permissions.

Approve / Reject with SharePoint Online API Management Page

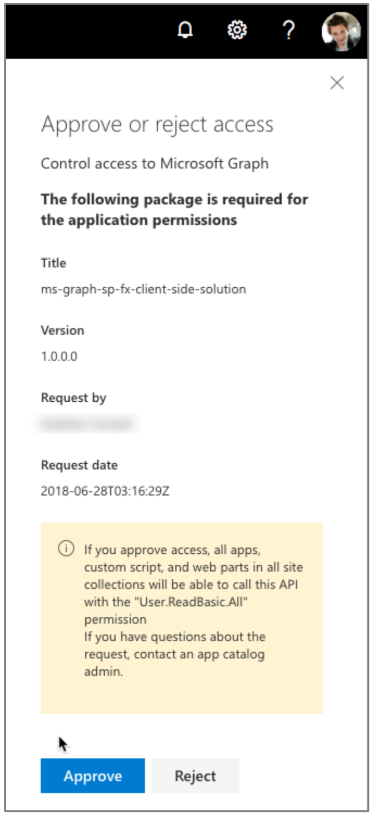


API management

Control the third-party APIs that can be called by apps and custom script.

Approve or reject

✓	API name	Permission
✓	**Pending approval (1)**	
✓	Microsoft Graph	User.ReadBasic.All
	Approved (0)	



Approve or reject access

Control access to Microsoft Graph

The following package is required for the application permissions

Title

ms-graph-sp-fx-client-side-solution

Version

1.0.0.0

Request by

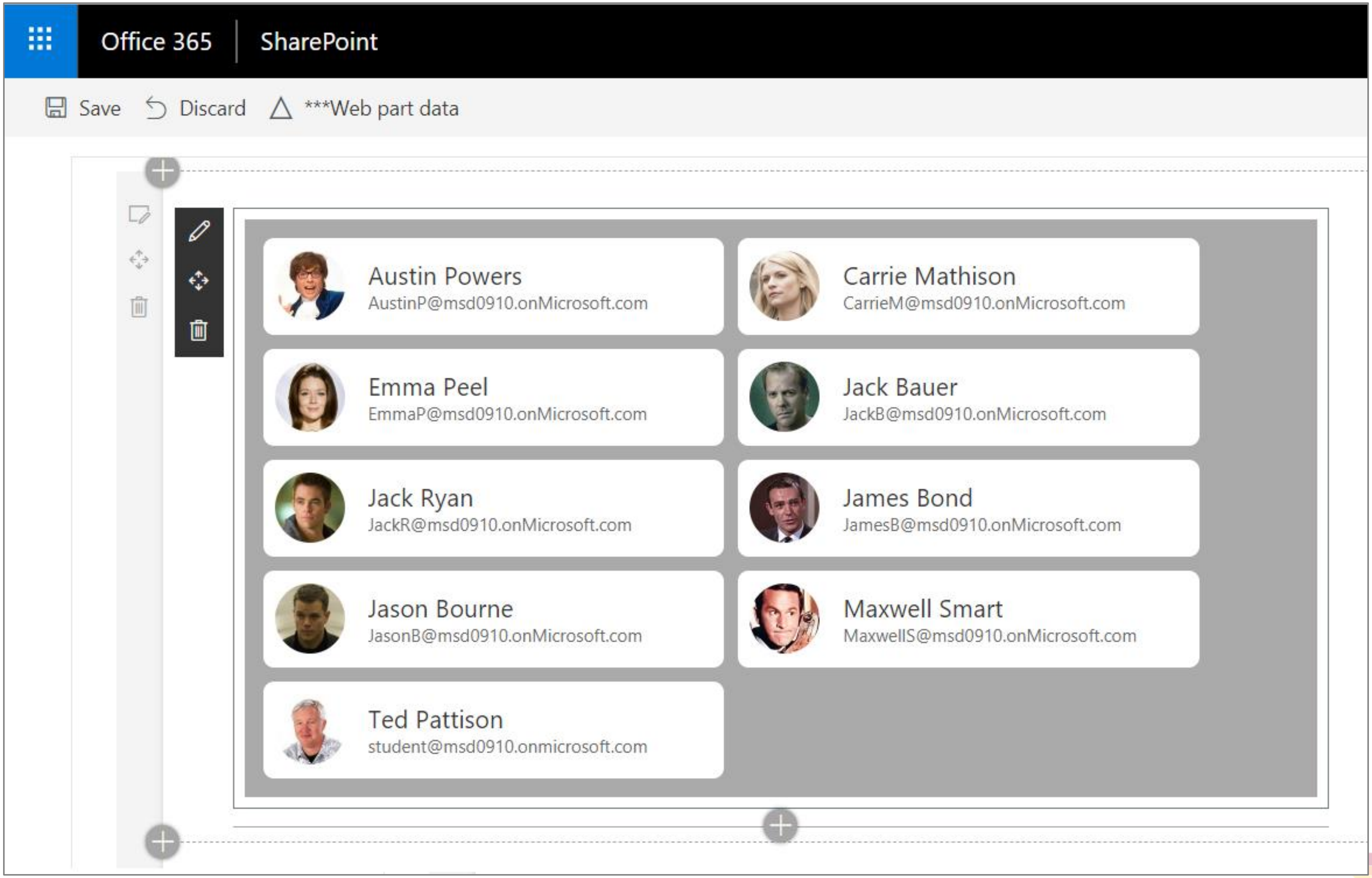
[redacted]

Request date

2018-06-28T03:16:29Z

ⓘ If you approve access, all apps, custom script, and web parts in all site collections will be able to call this API with the "User.ReadBasic.All" permission
If you have questions about the request, contact an app catalog admin.

Demo 5 - Calling Microsoft Graph API



Austin Powers
AustinP@msd0910.onMicrosoft.com

Carrie Mathison
CarrieM@msd0910.onMicrosoft.com

Emma Peel
EmmaP@msd0910.onMicrosoft.com

Jack Bauer
JackB@msd0910.onMicrosoft.com

Jack Ryan
JackR@msd0910.onMicrosoft.com

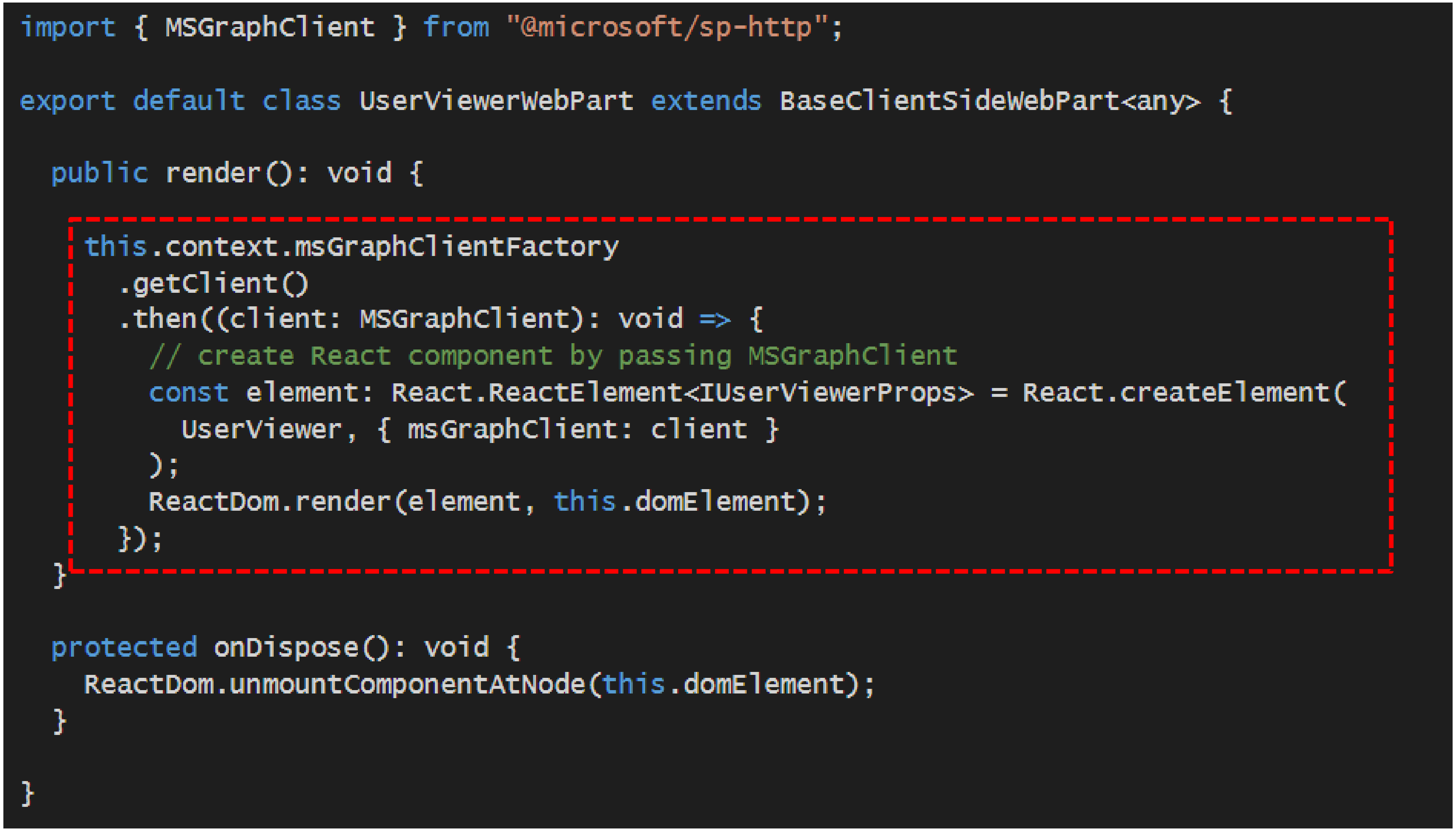
James Bond
JamesB@msd0910.onMicrosoft.com

Jason Bourne
JasonB@msd0910.onMicrosoft.com

Maxwell Smart
MaxwellS@msd0910.onMicrosoft.com

Ted Pattison
student@msd0910.onmicrosoft.com

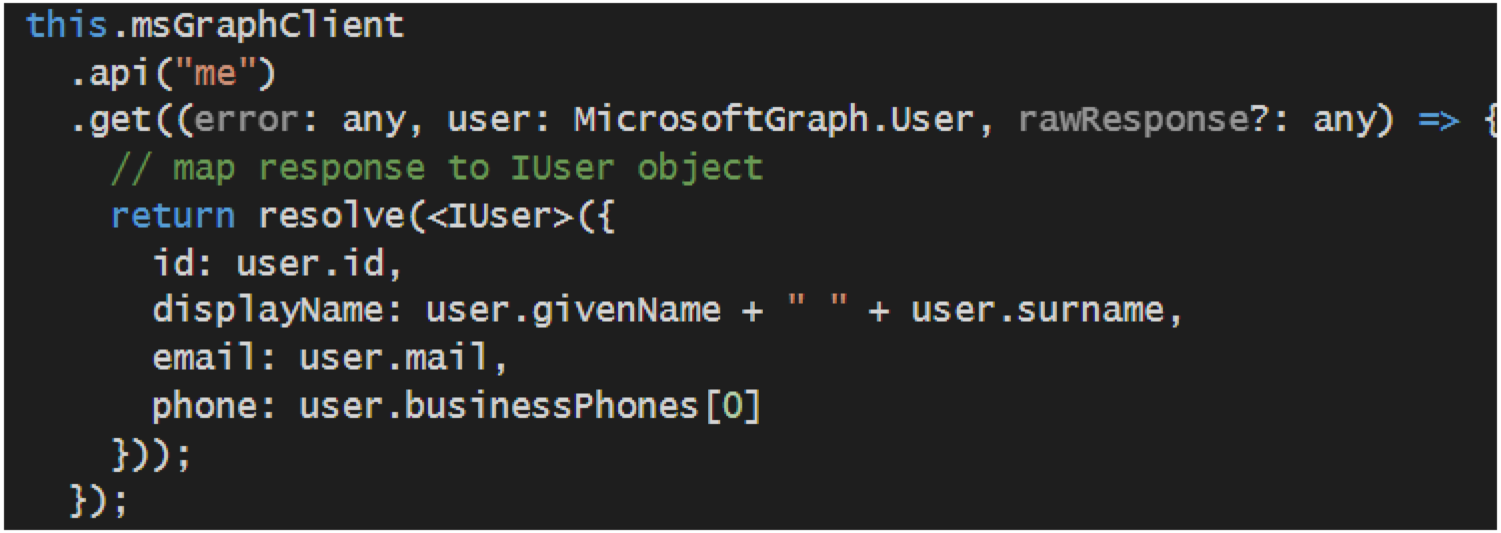
Acquiring the MSGraphClient from WebPart



```
import { MSGraphClient } from "@microsoft/sp-http";

export default class UserViewerWebPart extends BaseClientSideWebPart<any> {

  public render(): void {

    this.context.msGraphClientFactory
      .getClient()
      .then((client: MSGraphClient): void => {
        // create React component by passing MSGraphClient
        const element: React.ReactElement<IUserViewerProps> = React.createElement(
          UserViewer, { msGraphClient: client }
        );
        ReactDom.render(element, this.domElement);
      });
  }

  protected onDispose(): void {
    ReactDom.unmountComponentAtNode(this.domElement);
  }

}
```

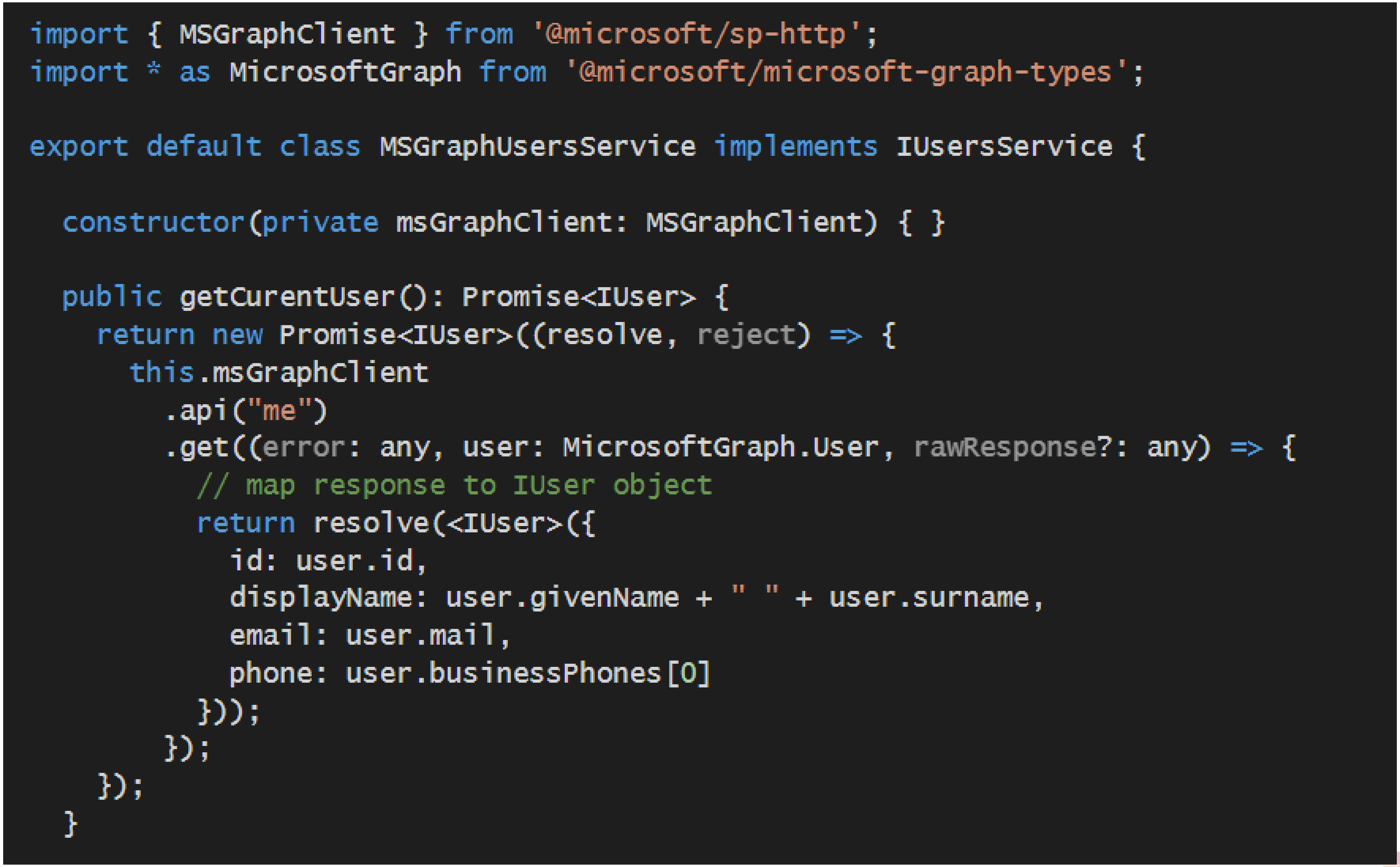
Microsoft Graph TypeScript Type Declarations

- Use the Microsoft Graph JavaScript SDK in TypeScript applications
- TypeScript type declarations introduce strong types
  - https://github.com/microsoftgraph/msgraph-typescript-typings



```
this.msGraphClient
  .api("me")
  .get((error: any, user: MicrosoftGraph.User, rawResponse?: any) => {
    // map response to IUser object
    return resolve(<IUser>({
      id: user.id,
      displayName: user.givenName + " " + user.surname,
      email: user.mail,
      phone: user.businessPhones[0]
    }));
  });
```

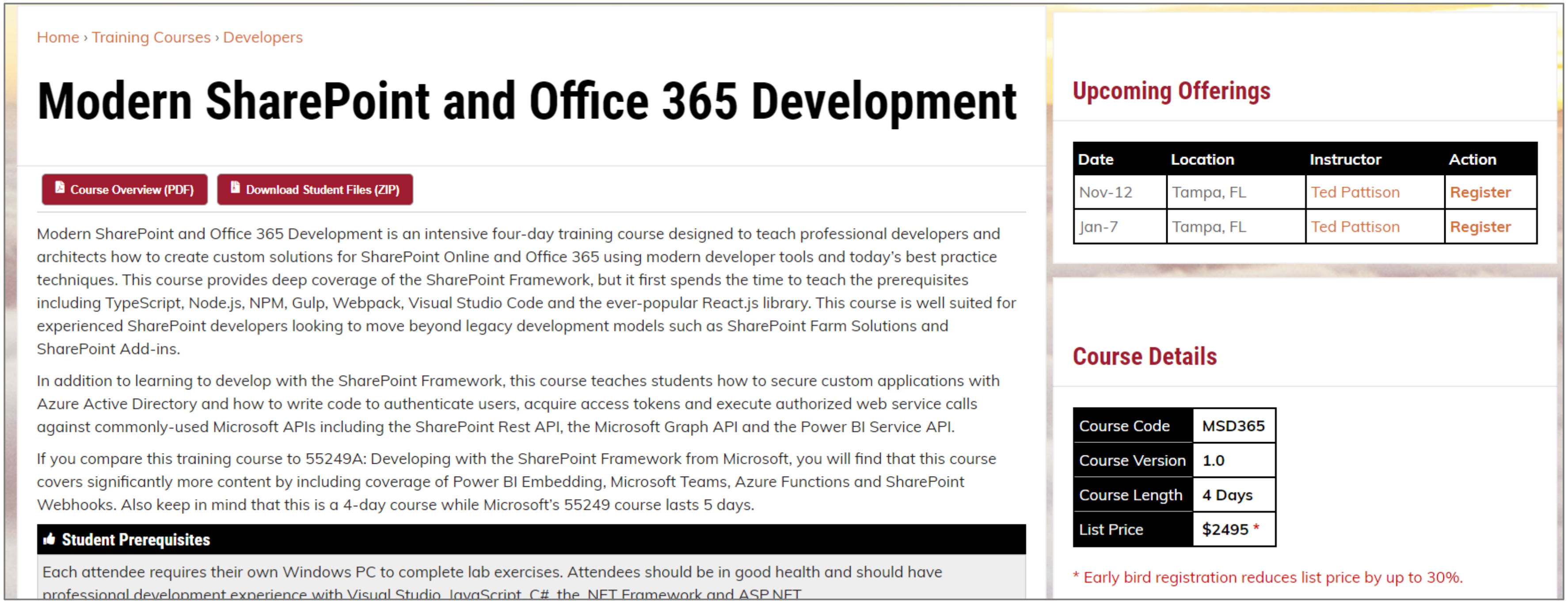
Service Class using Microsoft Graph API



```
import { MSGraphClient } from '@microsoft/sp-http';
import * as MicrosoftGraph from '@microsoft/microsoft-graph-types';

export default class MSGraphUsersService implements IUsersService {

  constructor(private msGraphClient: MSGraphClient) { }

  public getCurentUser(): Promise<IUser> {
    return new Promise<IUser>((resolve, reject) => {
      this.msGraphClient
        .api("me")
        .get((error: any, user: MicrosoftGraph.User, rawResponse?: any) => {
          // map response to IUser object
          return resolve(<IUser>({
            id: user.id,
            displayName: user.givenName + " " + user.surname,
            email: user.mail,
            phone: user.businessPhones[0]
          }));
        });
    });
  }
```

Effective SharePoint Framework Training

- MSD365: Modern SharePoint and Office 365 Development
  - 4-day of training with lots of hands-on labs
  - Learn how to develop with SPFx the right way right from the start
  - https://www.criticalpathtraining.com/courses/sharepoint/modern-sharepoint-office-365-development/



Home › Training Courses › Developers

Modern SharePoint and Office 365 Development

Modern SharePoint and Office 365 Development is an intensive four-day training course designed to teach professional developers and architects how to create custom solutions for SharePoint Online and Office 365 using modern developer tools and today's best practice techniques. This course provides deep coverage of the SharePoint Framework, but it first spends the time to teach the prerequisites including TypeScript, Node.js, NPM, Gulp, Webpack, Visual Studio Code and the ever-popular React.js library. This course is well suited for experienced SharePoint developers looking to move beyond legacy development models such as SharePoint Farm Solutions and SharePoint Add-ins.

In addition to learning to develop with the SharePoint Framework, this course teaches students how to secure custom applications with Azure Active Directory and how to write code to authenticate users, acquire access tokens and execute authorized web service calls against commonly-used Microsoft APIs including the SharePoint Rest API, the Microsoft Graph API and the Power BI Service API.

If you compare this training course to 55249A: Developing with the SharePoint Framework from Microsoft, you will find that this course covers significantly more content by including coverage of Power BI Embedding, Microsoft Teams, Azure Functions and SharePoint Webhooks. Also keep in mind that this is a 4-day course while Microsoft's 55249 course lasts 5 days.

Student Prerequisites

Each attendee requires their own Windows PC to complete lab exercises. Attendees should be in good health and should have professional development experience with Visual Studio, JavaScript, C#, the .NET Framework and ASP.NET

Upcoming Offerings

Date	Location	Instructor	Action
Nov-12	Tampa, FL	Ted Pattison	Register
Jan-7	Tampa, FL	Ted Pattison	Register

Course Details

Course Code	MSD365
Course Version	1.0
Course Length	4 Days
List Price	$2495 *

* Early bird registration reduces list price by up to 30%.

Summary

- ✓ Developing with Node.js, TypeScript and Webpack
- ✓ Learning React.js Fundamentals
- ✓ Using the Office UI Fabric React Component Library
- ✓ Developing React Webparts with SharePoint Framework
- ✓ Calling the Microsoft Graph API from React Webparts