React Training

Delivered by finnovate.io

Prerequisite

Prior software development experience

JavaScript development experience is a plus

Access to workstation with admin privileges and internet access

Agenda

Day 1

- Single Page Application
- Introduction to React
- Redux and Uni-directional data flow
- JavaScript & ECMAScript 6
- Development Tools
- React Component
- React Router
- React Component State & Events
- Component styling

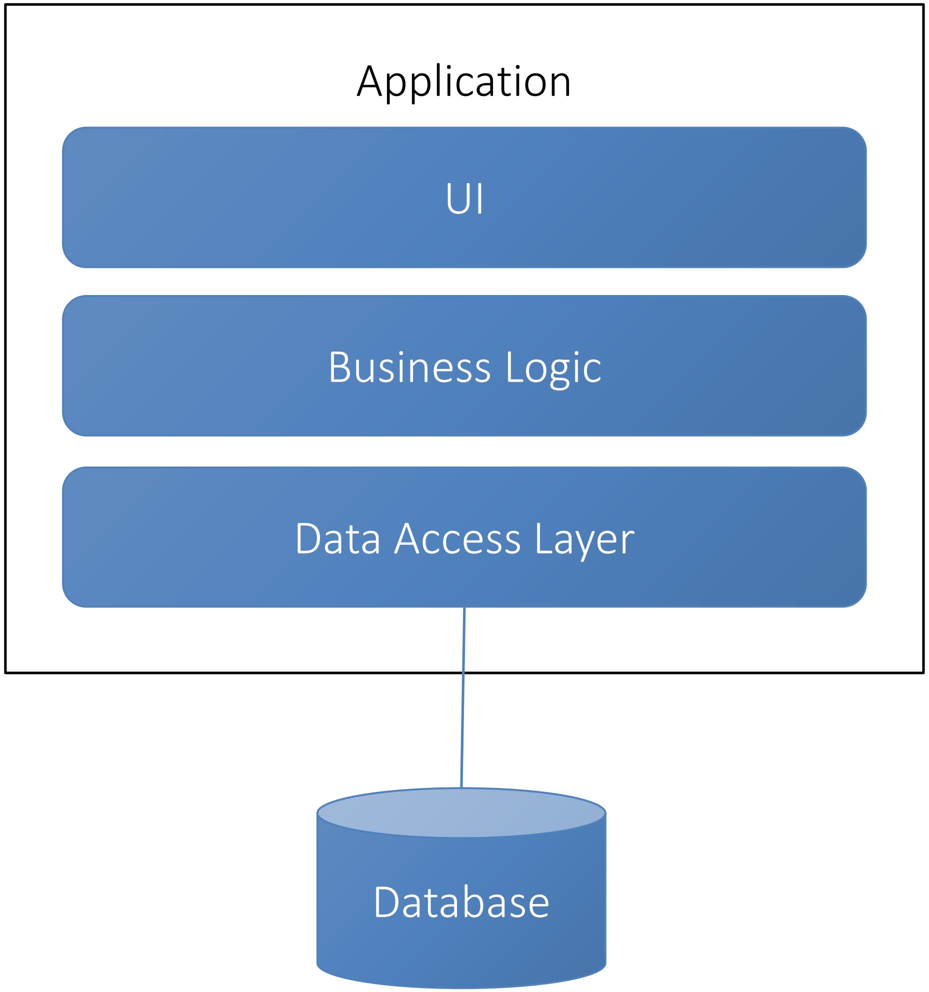
Day 2

- Redux Data Store and Reducer
- Redux Action
- Managing Business Logic
- RESTful API
- Promises
- Hydrating Data Store with Middleware
- API Security
- Unit Testing

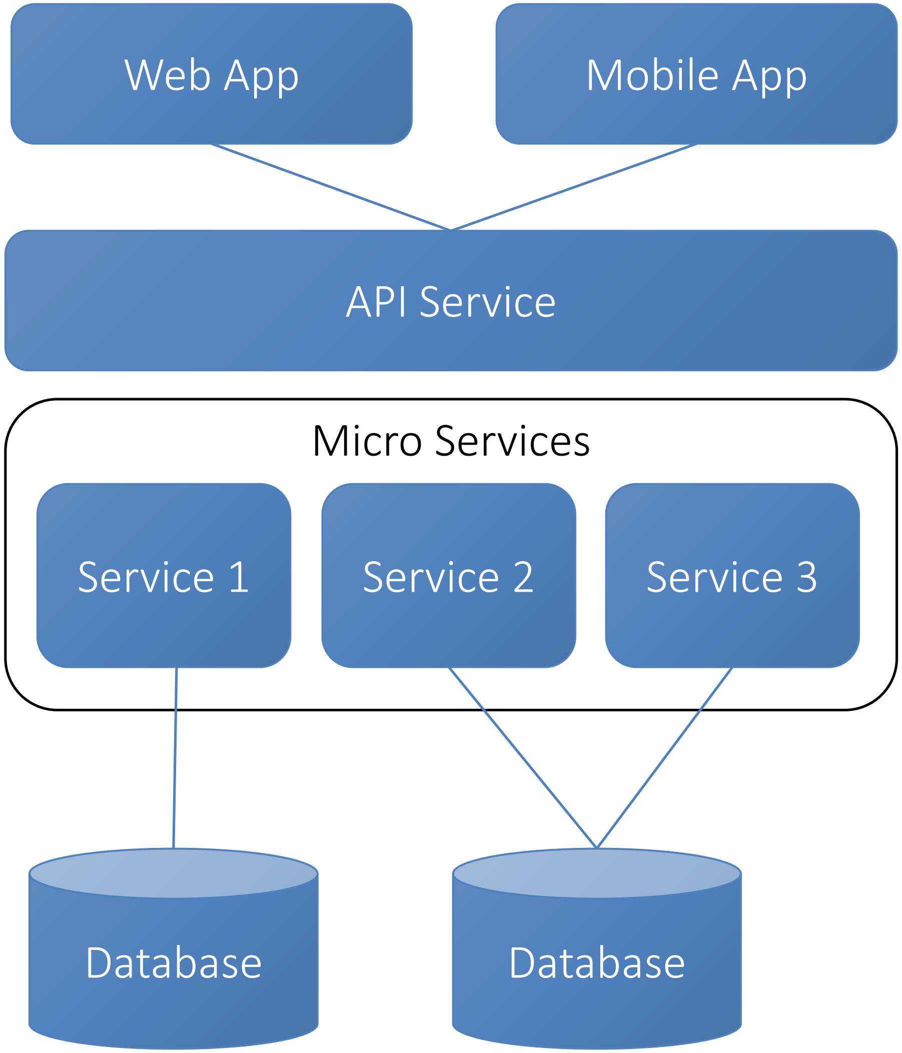
Single Page Application Architecture

Delivered by finnovate.io

Traditional Monolithic Application



SPA Architecture



SPA Overview

As the name implies, the entire application rests on a single page, more specifically index.html

Content is rendered by JavaScript based on application state

SPA Benefits

Decoupling of front end back server side development

Allows for “snappier” apps and rich user interactions

Delegate business logic to browser and mobile devices to reduce the need for heavy servers

Allow front end developers to focus on user experience and back end developers to focus on data and security

Architecture is web and mobile friendly

Introduction to React

Delivered by finnovate.io

React Overview

React is a JavaScript library created by Facebook for rendering views in web applications

It is fast, simple and scalable

It is **not** a MVC framework

At its core, it uses Virtual DOM to process updates in the view

It endorses a component based architecture, where an application is a tree of components

React is used extensively by Facebook, Instagram, Netflix, Airbnb, Walmart and many more organizations

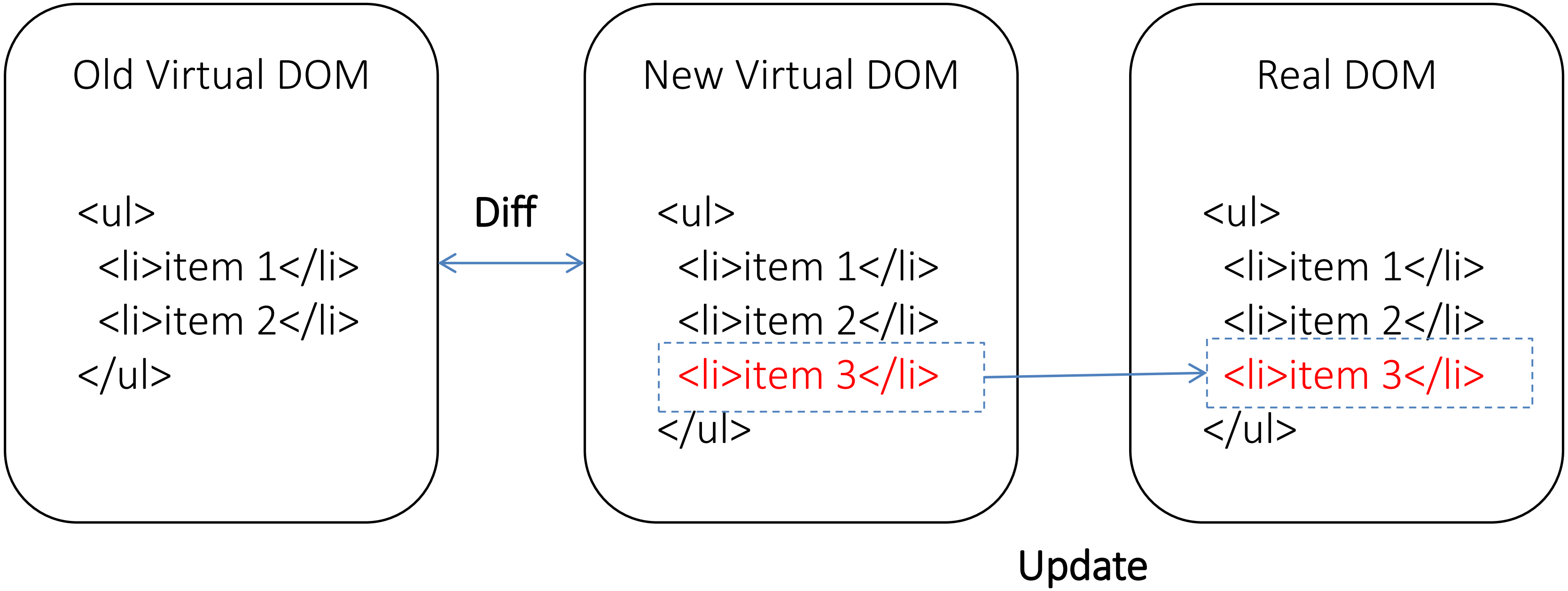
Virtual DOM

React uses virtual DOM to compare changes between render function calls in memory and makes minimal changes to the real DOM

Eliminates the need the JQuery – avoid JQuery whenever possible

Virtual DOM

Virtual DOM exists in memory



Old Virtual DOM

```
<ul>
  <li>item 1</li>
  <li>item 2</li>
</ul>
```

Diff

New Virtual DOM

```
<ul>
  <li>item 1</li>
  <li>item 2</li>
  <li>item 3</li>
</ul>
```

Real DOM

```
<ul>
  <li>item 1</li>
  <li>item 2</li>
  <li>item 3</li>
</ul>
```

Update

One-way Data Flow

Immutable data is passed to React components to be rendered

No more two-way data binding

Components cannot modify data being passed to it, but instead, they can trigger actions to change application state

Overall application data flow is done with Redux

JSX

React doesn’t use HTML templates like other SPA frameworks

Instead, it uses JSX, an HTML like extension syntax that gets processed into JavaScript

JSX creates a development experience that is familiar without the performance limitations of HTML templates

Redux and uni-directional data flow

Delivered by finnovate.io

Uni-directional data flow

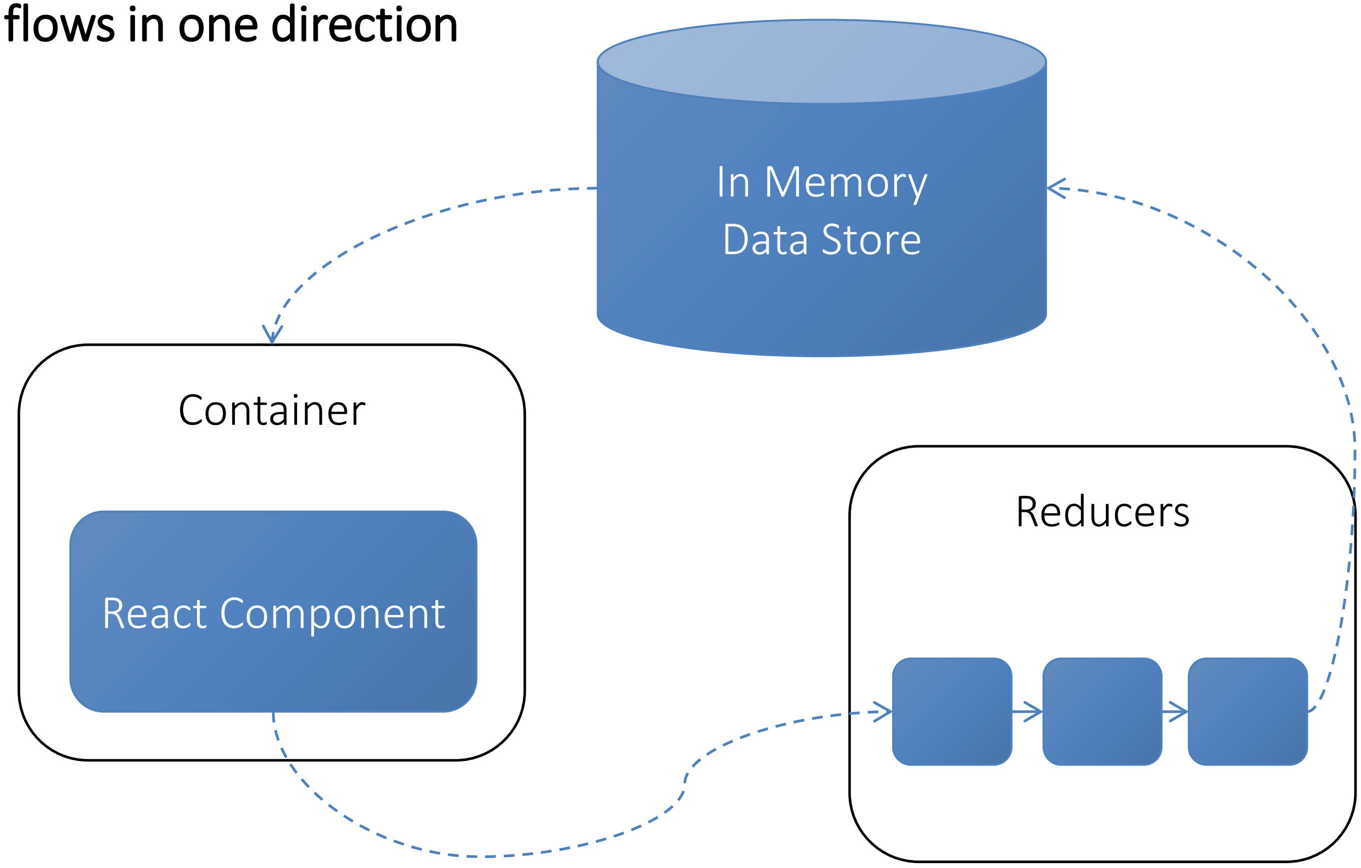
Data flows in a circle, in one direction between the data store and the view

The view is React

Redux is a popular implementation of the data store and various application elements

Redux Architecture

Data flows in one direction



Action & Payload

Redux Data Store

Redux features a single data store to capture the current state of the application

The data store is a single tree of data

The idea is to have the data store dictate what the view (React) renders

Redux Action

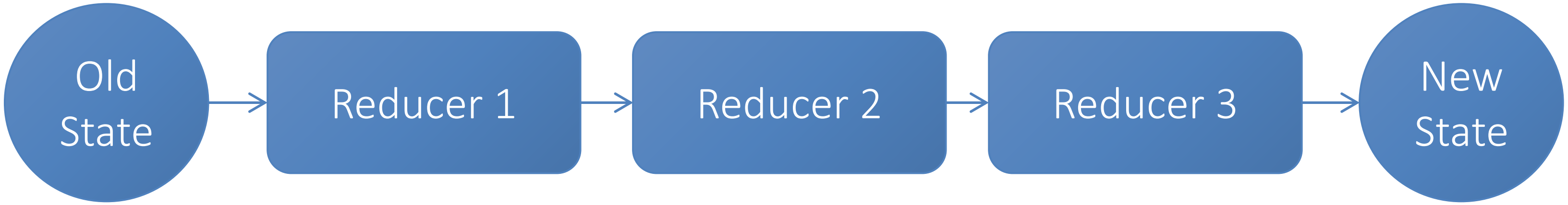
The view cannot make direct changes to the data store

Instead, the view can trigger actions (usually from user interactions) to modify application state in the data store

Redux Reducer

A reducer is a “pure function” that looks for specific actions and introduces changes to the application state accordingly

In Redux, reducers are “chained” together with function composition



Container

Container glues the specific parts of the data store to individual React components

It is good practice to use containers because it decouples the view from the store, allowing for multiple implementations of the view (i.e. desktop & mobile)

Introduction to JavaScript

Delivered by finnovate.io

JavaScript Overview

JavaScript is an interpreted programming language, no compilation is required

JavaScript runs within the browser, or within a runtime environment like Node.js

JavaScript originated in the mid-90’s to make HTML more dynamic

Now JavaScript can be used to write complex and scalable applications

Functional Programming

JavaScript supports various programming styles, but favors functional programming

JavaScript supports first-class functions

- Functions can be assigned to variables
- Functions can be passed as arguments into a function
- A function can be returned by a function

Pure function

A pure function is deterministic with no side effects

Use pure functions whenever possible

Easier to test and debug

No side effects

```
function add(x, y) {
  return x + y;
}
```

Side effect

```
var z = 0;

function add(x, y) {
  z = x + y + z;
  return z;
}
```

Function composition

Combining two or more functions to form a new function

```
var numbers = [1, 2, 3];

var doubles = numbers.map(function(num) {
  return num * 2;
});
```

ECMAScript 6

Delivered by finnovate.io

ES6 Overview

ES6 is an extension of ES5, more commonly known as JavaScript

ES6 is not currently supported by browsers

In order to develop with ES6, transpilation is required to convert code to ES5 (Babel)

ES6 offers features that React takes advantage of

Classes

Special functions used to create “objects”

It simplifies existing JavaScript prototype inheritance, it’s really syntactical sugar and it doesn’t introduce new concepts

React components are classes

```
Class Customer {
  constructor(name, company) {
    this.name = name;
    this.company = company;
  }
}

Class Shopper extends Customer {
  buyItem(itemId) {
    console.log(this.name + ‘ buys an item’);
  }
}
```

Importing Functions

Used to import functions, objects and primitives from another module, script or file

Use {} to import functions that are not default members

```
import { getTodaysDate } from ‘./date-module.js’
```

Export in date-module.js:

```
export function getTodaysDate() {
  return new Date()
}
```

Importing Classes

Classes can also be imported

Use default to specify default function or class

```
import Pokemon from ‘../components/pokemon/pokemon.js’
```

Export in date-module.js:

```
import React from ‘react’

export default class Pokemon extends React.Component {

}
```

Arrow functions

Short hand way to declare functions

Unlike function(), arrow function does not bind its own “this”. “this” in an arrow function refers to its enclosing context. This concept becomes important when working with React components.

```
(a, b) => a + b
```

Is the same as:

```
function(a, b) {
  return a + b;
}
```

Functions in JavaScript can be anonymous!

Other ES6 Features

Const

Let

Template Strings

```
console.log(`strings can be declared
  on multiple lines`);
```

Spread

```
const array = [1, 2, 3];
 function add(x, y, z) {
   return x + y + z;
 }
 add(…array);
```

Others…

Further content removed from Sample

Visit us at

https://finnovate.io