

/#ILoveReact

Jon D Jones

Contents

I LOVE React	2
Testing	3
How To Write A Test	3
Code Samples	4
Hello World	4
Higher Order Components	4
Filter import React from 'react'	5
JSON Form Example	6
Redux 101	7
Resources	10
Interesting Reads	10
Interesting Videos	10
React Documentation	10
Why React?	11
Stuff I Use	12
Equipment I Use	14
My Code	16
React Projects	16
Node Projects	16
My NPM Packages	16
Misc Projects	16
Episerver Projects	16
Umbraco Projects	17

I LOVE React

Welcome to I LOVE React, in this book you will learn everything React.

Testing

How To Write A Test

```
describe('My Feature', () => {  
  it('WHEN triggered with this data THEN this should happen', () => {  
    expect(true).ToBeTruthy();  
  });  
});
```

Code Samples

Hello World

Below shows the smallest react app you can make

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Hello World</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
</head>

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

<script src="https://fb.me/react-0.14.6.js"></script>
<script src="https://fb.me/react-dom-0.14.6.js"></script>

<script>
  var displayHtml = React.createElement('h1', {}, 'Hello World');
  ReactDOM.render(displayHtml, document.getElementById('react-target'))
</script>
</html>
```

Higher Order Components

HOC names usually start with the term with as it adds on extra information

```

import React from 'react';

const withValidation = (WrappedComponent, validations) => {
  class HOC extends React.Component {
    render() {
      return (
        <WrappedComponent
          {...this.props}
          validations={validations}
        />
      );
    }
  }

  return HOC;
};

export default withValidation;

import React from 'react';
import withValidation from './with-validation';

const DisplayTheSecret = props => (
  <div>
    The secret to life is {props.secret} {props.secret}.
  </div>
);

const WrappedComponent = withValidation(DisplayTheSecret, 'this');
export default WrappedComponent;

```

Filter import React from 'react'

This component will take in an array of data and filter it. This component shows the benefit of using `componentWillMount()`

```

import React from 'react';

export default class MyFilteringComponent extends React.Component {
  state = {
    initialItems: [],
    items: []
  }

  filterList = (event) => {
    let items = this.state.initialItems;

```

```

        items = items.filter((item) => {
            return item.toLowerCase().search(event.target.value.toLowerCase()) !== -1;
        });
        this.setState({items: items});
    }

    componentWillMount = () => {
        this.setState({
            initialItems: this.props.content,
            items: this.props.content
        })
    }

    render() {
        return (
            <div>
                <h3>
                    Filtering Example
                </h3>
                <form>
                    <input type="text" placeholder="Search" onChange={this.filterList}/>
                </form>
                <div>
                    {
                        this.state.items.map(function(item) {
                            return <div key={item}>{item}</div>
                        })
                    }
                </div>
            </div>
        );
    }
};

```

JSON Form Example

JSON form allows you to create a web form in a react app. When you use JSON form you do not need to write much code yourself, as it allows you to build your form through a JSON schema. Lets say you have the JSON schema defined as below

```

const mySchema = {
    "title": "My Form",
    "description": "My Form",
    "type": "object",

```

```

    "required": [
      "name"
    ],
    "properties": {
      "name": {
        "type": "string",
        "title": "Name"
      }
    }
  }
};

```

You could then use this code to render a form

```

import React, { Component } from 'react';
import Form from 'react-jsonschema-form'

export default class MyForm extends Component {
  constructor(props) {
    super(props);
    this.handleSubmit = this.handleSubmit.bind(this);
  }

  handleSubmit({formData}) {
    console.log(formData);
  }

  render() {
    return (
      <Form schema={mySchema} onSubmit={this.handleSubmit} />
    )
  }
}

```

Redux 101

Basic example of how to get data in and out of redux

```

import { createStore } from 'redux';

const incrementCount = ({increaseBy = 1} = {}) => ({
  type: 'ADD',
  increaseBy: increaseBy
});

const decrementCount = ({decrementBy = 1} = {}) => ({

```



```

    type: 'MINUS',
    decrementBy: decrementBy
  });

const multiplyCount = ({multiplyBy = 1} = {}) => ({
  type: 'MULTIPLY',
  multiplyBy: multiplyBy
});

const resetCount = ({Payload} = {}) => ({
  type: 'RESET'
});

const setCount = ({count} = {}) => ({
  type: 'SET',
  count: count
});

const countReducer = (state = {count: 0}, action) => {
  switch(action.type) {
    case 'ADD':
      return {
        count: state.count + action.increaseBy
      }
    case 'MINUS':
      return {
        count: state.count - action.decrementBy
      }
    case 'MULTIPLY':
      return {
        count: state.count + action.multiplyBy
      }
    case 'SET':
      return {
        count: action.count
      }
    case 'RESET':
      return {
        count: 0
      }
  }

  return state;
};

```

You can talk to the store, by:

```
const store = createStore(countReducer);
const unsubscribe = store.subscribe(() => {
  console.log(store.getState());
});

store.dispatch(incrementCount());
store.dispatch(mulitplyCount({multiplyBy: 5}));
store.dispatch(decrementCount({decrementBy: 2}));
store.dispatch(resetCount());
store.dispatch(setCount({count: 101}));
unsubscribe();
store.dispatch(mulitplyCount());
```

Resources

Interesting Reads

- Debugging Javascript in Chrome
- Enzyme Cheatsheet
- Higher Order Functions
- Intro To Hooks
- Jest Cheatsheet
- Hot Reloading with Time Travel by Dan Abramov
- How to Speed Up Learning as a Software Developer
- React Hooks by Dan Abramov
- Testing React Apps
- Why Do We Write super(props)?

Interesting Videos

- Fun with React Hooks - Michael Jackson and Ryan Florence
- :star: React Today and Tomorrow and 90% Cleaner React With Hooks
- React: CSS in your JS by Christopher Chedeau

React Documentation

- React Component
- Conditional Rendering
- Components and Props
- Composition vs Inheritance
- Error Handling
- Forms
- HTML Attributes
- Handling Events
- JSX
- Lifting State Up

- Lists and Keys
- PropTypes
- React Components, Elements, and Instances
- Refs
- Rendering Elements
- Render Props
- State and Lifecycle
- Synthetic Events

Why React?

- Getting Started With React
- 10 Reasons why I moved from Angular to React

Stuff I Use

Software

- Visual studio code My preferred text editor

My Top 10 List Of Books Every Developer Should Read

- Code Complete: A Practical Handbook of Software Construction, Second Edition
This is my bible of software development. It's a monster to read but it contains pretty much everything you need to know about crafting good code. After reading this book people commented about how much better my coding became.
- Clean Code: A Handbook of Agile Software Craftsmanship
Everyone of Uncle Bobs books are essential reading this one in my opinion is the best. Theres a big difference between writing code and writing good code. This book will help you to think about what good code is and how you can write it
- Refactoring: Improving the Design of Existing Code
I'm a strong believer in iterative design. Build something basic that works and fits the bill and then improve it constantly. This constant improvement is called refactoring. This book will teach you all the patterns and techniques you will need to accomplish that
- Dependency Injection in .NET
To check that the code you write works, you will need to test it. To write testable code, you need to understand dependency injection. This book will teach you how you should use dependency injectionS
- Head First Design Patterns: A Brain-Friendly Guide
Every software developer needs to know about design patterns. This book makes it super simple to learn the main patterns. If design patterns scare you, then this should be your next read.
- Code: The Hidden Language of Computer Hardware and Software
This book will help you understand how your computer works. It's easy to read, and it will help you understand concepts like binary addition, gates and circuit boards.

- **The Pragmatic Programmer**
This book is one of the first software development books I read. This book will provide you with a plan on how to improve your coding skills.
- **The DevOps Handbook**
Writing code isn't the only part of software development. You will also have to release it. This book will teach you everything you will need in order to be able to release your code correctly.
- **The Clean Coder**
This book will teach you how to be a professional. This book isn't about coding. It's how to behave like a great coder.
- **Design Patterns: Elements of Reusable Object-Oriented Software**
Every software developer should read this book. This isn't the easiest book to read, which is why I recommend reading head first patterns first. You should read this book for bragging rights!

List Of Great Books That Will Help You Get Promoted

- **Outliers: The Story of Success**
This book will teach you about practice. To be a good coder you will need to spend at least 10,000 hours coding. Being great isn't about talent, it's about practice. People who practice will become better. You will also learn about the right type of practice
- **Deep Work: Rules for Focused Success in a Distracted World**
This book changed my life. Just because your busy doesn't mean you're succeeding. Deep work will give you a system to prioritise your life so you can achieve more
- **Bounce: Mozart, Federer, Picasso, Beckham, and the Science of Success**
When I started programming it seemed like some people were naturally just gifted programmers. I was not one of them. I thought being good was down to your genes. This book proves differently. This book will teach you about practice and more importantly deliberate practice. Understanding this is essential in becoming a great programmer.
- **The Adweek Copywriting Handbook**
We all need to write emails, blog posts, memo's. Learning how to improve how you write is an important skill that you will use throughout your life. This book will help you do that.
- **The Power of Habit: Why We Do What We Do in Life and Business**
We all need to write emails, blog posts, memo's. Learning how to improve how you write is an important skill that you will use throughout your life. This book will help you do that.
- **The Intelligent Investor: The Definitive Book on Value Investing**
Software development is a well paying job, however, if you don't know how to manage your money you will be working all your life. This book was originally written in the 1930's. The advice in this book is still true.
- **Henry Ford - My Life and Work**
This book was written over 100 years ago. How Ford managed his planet..

was the original agile manifesto. The principles For applied to creating motorcars can be applied to modern software design.

- Henry Ford - My Life and Work
To be a good software craftsman you need to understand the beauty of design. Steve jobs was known for his eye for detail and his never ending pursuit for making things better. Reading this book will help you appreciate design more
- Eric Schmidt - How Google Works
Every developer has heard what a great place Google is to work. This book gives you an insight into how Google was started and the ethos that made them so great.

Equipment I Use

- Dell UltraSharp U3415W 34-Inch Curved LED-Lit Monitor
Dell is known for making great products. This monitor is no exception. It doesn't give me eye-strain. I can have a split screen on two HDMI's, so I view a MAC and a PC on the screen at the same time. Plugging the HDMI cables could be easier, aides from that its worth buying.
- ASUS ZenBook As a contractor I've had a lot of laptops (currently over 10 dell's)> I prefer the Zenbook over all of them. It's quick, light, keyboards good. The only downside are the speakers are too quiet.
- Apple Mac Mini
When I've doing Javascript coding or video editing I use a mac-mini. Its handy to have a desktop and the mac-mini is the best small looking computer.
- Bose QuietComfort 35 Wireless Noise Cancelling Headphones
I can't recommend these enough. When you put them on they block everything out. Sounds amazing. The charge lasts for ages. It recharges really quickly. I was skeptical about buying battery powered headphones, but these are much better than wired
- FSDUALWIN Aluminum Laptop Monitor Stand Space Bar, Monitor Riser MacBook Monitor Dock Desk Organizer with 4 USB Ports, Keyboard Storage
I spent ages looking for a good looking monitor stand and this one fits the bill. Looks good with brushed aluminum. HAS 4 USB ports so I can plug the Echo dot in. Makes my desk look cleaner as I can hide my mouse and keyboard under it when not in use
- WD 4TB My Cloud Personal Network Attached Storage
If you need an external hard drive this is a great choice. Connect it to your broadband and you can access your files anywhere in the world. I can view my media on my Xbox. I can access it from any device in my house. Looks good
- Canon EOS REBEL T7i EF-S 18-55
This is the camera loads of Vloggers recommended buy. The quality of the

picture and video is good. Worth buying

- Rode VMGO Video Mic GO
This is the best camera MIC you can buy. Good sound quality. Not bad price
- Rode Podcaster USB Dynamic Microphone
This USB MIC plugs into my PC/Mac. Great sound quality. I recommend this MIC for anyone wanting to do voice and screen-sharing
- Xbox One X 1TB Console
I'm an Xbox guy. Xbox One is the best games and media player. FACT!
- Echo Dot
Its fun :)
- Samsung Galaxy S7 Edge
My first smart phone was an iPhone. After trying a Samsung I switched and never looked back. Camera on this phone is great. Battery life is good.
- Fitbit Charge 2
I never thought I'd buy a smart-watch but I love this. If you want to become more active, a great challenge is to try and walk 10,000 steps a day. This watch helps you to do that. Since I've brought it I'm on a 50 day streak.
- Fitbit Aria 2 Wifi + Bluetooth Smart Scales
I never thought I'd buy a smart-watch but I love this. If you want to become more active, a great challenge is to try and walk 10,000 steps a day. This watch helps you to do that. Since I've brought it I'm on a 50 day streak
- TomTom Go 510 5 Inch Sat Nav With World Maps
Need a Satnav? This is worth considering. Comes with speed camera detection, which 'may' have saved me on a few occasions!
- Dyson AM04 Hot + Cool Heater/Table Fan
This is expensive, but it is the coolest looking fan/heater on the market. In the summer acts as a fan. In the winter a heater

My Code

Below gives an overview of all the repos that you can find on my Github account

React Projects

- Todo App
- Portfolio
- Expense Tracker Website Allow users to add/edit/remove expenses. Integrates with firebase
- React Sample Website

Node Projects

- Node Examples

My NPM Packages

- https-status-lookup

Misc Projects

- CSS Playground
- HackDays
- Packages Playground

Episerver Projects

- Episerver BaseBuild

- Episerver Goodies
- EpiServerBlogSampleSite
- DeploymentTool
- EpiServerDonutCaching
- Episerver10DisplayOptions
-
-

Umbraco Projects

- Umbraco SampleSite
- Surface Controller Example