

Test-Driven Development (TDD) Using React.js and ES6 Eğitimi

Eğitim Hakkında

Test-Driven Development (TDD) Using React.js and ES6 Eğitiminde katılımcılar ön uç web geliştirmedeki en güncel araçlar, teknikler ve teknolojilerden bazılarıyla uygulamalı deneyim kazanacaklar.

Test-Driven Development (TDD) Using React.js and ES6 Eğitiminde, modern ön uç geliştirmeye eksiksiz bir giriş niteliğindedir. React'i öğretirken modern JavaScript geliştirmenin sözdizimini, kurallarını ve en iyi uygulamalarını da öğreneceksiniz. Bu süreçte katılımcılar, React/Redux uygulaması oluştururken TDD ve özellikle Jest frameworkünü kullanarak React ve Redux uygulamalarının nasıl test edileceğini öğreneceksiniz.

Neler Öğreneceksiniz

- Learn how to install, configure and use modern web tooling.
- Understand test-driven development.
- Learn to create test suites using Jasmine.
- Write ES6 code and compile it using Babel.
- Understand what React.js is and what problem it solves.
- Explore the basic architecture of a React.js application.
- Gain a deep knowledge of React.js components and JSX.
- Build a working application that uses React.js components.
- Use Redux for maintaining state in a React.js application.

Eğitim İçeriği

React Quickstart with Create-React-App

Development Ecosystem

- Code Editors and IDEs
- Lab 1: Installing and Configuring WebStorm IDE
- Node.js
- EventEmitter
- Node Streams
- Node Modules
- Lab 2 - Getting Started with Node.js
- Git
- What is Version Control

- History of Git
- What is Git?
- 3 States of Git
- Git Workflow
- Lab 3 - Version Control With Git
- Command Prompt
- Know Your Shell

Reproducible Builds

- Why Automate Your Build?
- Build Requirements
- npm
- Lab 4 - Initialize npm
- node_modules
- package.json
- npm install
- Lab 5 - Using npm as a Build Tool
- Lab 6 - Managing External Dependencies

Static Code Analysis

- Lint tools
- Configuring ESLint
- ESLint: What Can Be Configured?
- ESLint Rules
- Lab 7 - Automate Linting
- Lab 8 - Configure a Local Web Server
- Browser Development Tools

Test-Driven Development

- Goal of TDD
- The TDD Cycle
- Red, Green, Refactor
- Assertions
- JavaScript Testing Frameworks
- JS Exception Handling
- Jasmine Overview
- How Jasmine Works
- Test Suites
- Specs
- Expectations
- Matchers
- TDD vs. BDD

- Lab 9 – Get Started with Jasmine
- Lab 10 - TDD Practice
- Automated Cross-browser Testing
- Karma
- Lab 11 - In-browser Testing with Karma

Modularity

Why is Modularity Important?

- CommonJS
- RequireJS
- ES6 Modules
- Front-end Modules
- Manage Modules Manually
- Front End Package Management with npm

Building and Refactoring

- Building the dist directory
- webpack
- Lab 12: Deploying with Webpack
- Lab 13 README Update and Refactoring

ES2015 (ES6)

- Variable Scoping with const and let
- let vs. var
- Block-scoped Functions
- Arrow Functions
- Default Parameter Handling
- Rest Parameter
- Spread Operator
- Template Literals
- Enhanced Object Properties
- Array Matching
- Object Matching
- Symbol Primitive
- User-defined Iterators
- For-Of Operator
- Creating and Consuming Generator Functions
- Class Definition
- Class Declaration
- Class Expressions
- Class Inheritance
- Advanced JavaScript Topics

- “use strict”
- Understanding this
- 4 Rules of this
- What is this?
- Implicit Binding
- Explicit Binding
- new Binding
- window Binding
- Array.map()
- Promises
- Promises vs. Event Listeners
- Why use Promises?
- Demo: Callback vs. Promise
- Using Promises
- Babel
- Lab 14: Transpiling with Babel
- Lab 15: Converting to ES6

The Document Object Model

- What is the DOM?
- Understanding Nodes
- EventTarget
- DOM Events
- Other Events
- Element
- Manipulating HTML with the DOM
- Manipulating HTML with JQuery
- Manipulating HTML with React

Introduction to React.js

- What is React.js
- Imperative API vs. Declarative API
- Imperative vs. Declarative Screen updates
- One-way Data Flow
- Virtual DOM
- Virtual DOM vs. HTML DOM
- State Machines
- Lab 16, Part 1: Hello, React!
- Understanding Components
- React.render()
- ReactDOM
- React Development Process
- Props vs. State
- Setting Initial State

- super()
- Lab 16, Parts 2-3: Your first Component

JSX

- What is JSX?
- Using JSX
- JSX is not Exactly HTML
- Using React with JSX
- Using React without JSX
- Expressions in JSX
- Precompiled JSX
- Lab 17 - HTML to JSX

React Components

- Creating Components
- Pure Functions
- Benefits of Pure Functions
- F.I.R.S.T
- Single Responsibility
- Communication Between Components
- Props
- Ref Callback
- Lab 18: Passing Props
- Styles in React
- Style Components
- Lab 19: Style in React
- Forms
- Forms Have State
- Form Events
- Controlled Components
- Uncontrolled Components
- Lab 20: Controlling the Form
- Stateless Functional Components
- Lab 21: Refactoring the App
- Component Life-Cycle Events
- Life-Cycle Methods
- Mount/Unmount
- Mount/Unmount Life-Cycle Methods
- Data Life-Cycle Methods
- Component Life Cycle
- Events
- Lab 22: Life Cycle and Events
- Composition
- Reusable Components

- Presentational Components
- Container Components
- PropTypes
- Lab 23: PropTypes
- Testing React Components
- What to Test in a React Component
- Jest
- Mocking
- Snapshot Testing
- TestUtils
- Enzyme
- Shallow Rendering
- Lab 24: Testing React Components
- Lab 24.5: Testing with Jest and Enzyme
- Lab 25: Multiple Components
- React Router
- Lab 26: React Router
- Lab 27: React Router, Part 2

Flux and Redux

- Flux
- Flux Flow
- Flux Action
- Flux Dispatcher
- Flux Stores
- EventEmitter
- Redux
- Stores & Immutable State Tree
- Redux Actions
- Reducers
- Things You Should Never Do in a Reducer
- Reducer Composition
- Redux Store
- Redux Pros and Cons
- Lab 28: Redux Thermometer
- Lab 29: Implementing Redux
- React AJAX Best Practices
- Redux with Ajax
- What is Redux Middleware?
- What is Middleware Good For?
- Thunk
- Redux Saga
- Using Sagas
- Lab 30: SwimCalc App
- Lab 31: Redux Middleware
- create-react-app
- Lab 32: create-react-app and enzyme

- Relay and GraphQL