Hanbit RealTime 114

처음 시작하는 리액트

UI를 위한 자바스크립트 라이브러리 ReactJS

톰 핼럿, 리차드 펠드만,
시몬 회벡, 칼 미켈슨,
존 비비, 프랑키 반야르디 지음 /
곽현철, 김훈민 옮김

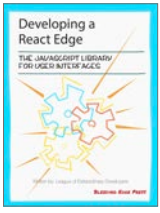
한빛미디어 Hanbit Media, Inc.

Hanbit RealTime 114

처음 시작하는 리액트

UI를 위한 자바스크립트 라이브러리 ReactJS

톰 핼럿, 리차드 펠드만,
시몬 회벡, 칼 미켈슨,
존 비비, 프랑키 반야르디 지음 /
곽현철, 김훈민 옮김



이 도서는
Developing a React Edge(BLEEDING EDGE PRESS)의
번역서입니다

한빛미디어 Hanbit Media, Inc.

표지 사진 **김재영**

이 책의 표지는 김재영 님이 보내 주신 풍경사진을 담았습니다.
리얼타임은 독자의 시선을 담은 풍경사진을 책 표지로 보여주고자 합니다.

사진 보내기 **ebookwriter@hanbit.co.kr**

처음 시작하는 리액트 UI를 위한 자바스크립트 라이브러리 ReactJS

초판발행 2016년 7월 25일

지은이 톰 핼렛, 리차드 펠드만, 시몬 회벡, 칼 미켈슨, 존 비비, 프랑키 반야르디 / **옮긴이** 곽현철, 김훈민 / **펴낸이** 김태헌
펴낸곳 한빛미디어(주) / **주소** 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부
전화 02-325-5544 / **팩스** 02-336-7124
등록 1999년 9월 30일 제10-1779호
ISBN 978-89-6848-775-0 15000 / **정가** 17,000원

총괄 전태호 / **책임편집** 김창수 / **기획·편집** 김상민
디자인 표지·내지 여동일, 조판 최송실
마케팅 박상용, 송경석, 변지영 / **영업** 김형진, 김진불, 조유미

이 책에 대한 의견이나 오탈자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주십시오.
한빛미디어 홈페이지 www.hanbit.co.kr / **이메일** ask@hanbit.co.kr

지금 하지 않으면 할 수 없는 일이 있습니다.
책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요.
한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

저자 소개

지은이_ **톰 핼럿**

샌프란시스코에 있는 실시간 비디오 플랫폼인 Tout.com의 Ruby/JavaScript 시니어 엔지니어이다. Jasmine을 이용한 React 애플리케이션 테스트를 도와주는 Jasmine-react의 제작자이기도 하다. 수중 폴로를 좋아하고, 아내와 아들과 함께 시간을 보낸다.

지은이_ **리차드 펠드만**

샌프란시스코에 있는 교육 기술 회사인 NoRedInk에서 리드 프론트엔드 엔지니어로 일하고 있다. 함수형 프로그래밍의 지지자이자, 강연자다. 일반적인 JavaScript 객체와 배열에 하위호환성을 갖는 이뮤터블 데이터 구조를 제공하는 오픈 소스 라이브러리인 *seamless-immutable*의 제작자이기도 하다.

지은이_ **시몬 회벡**

로드아일랜드주 프로비던스에 있는 Swipely에서 시니어 UI 엔지니어로 일하고 있다. 프로비던스 JS 밋업 그룹의 공동주최자이고, 보스턴의 Startup Institute에서 JavaScript를 가르치기도 했다. JavaScript를 이용한 기능적 유저 인터페이스를 만들고, cssarrowplease.com 같은 사이드 프로젝트에 시간을 할애하고 있다.

지은이_ **칼 미켈슨**

LockedOn에서 시니어 PHP/JavaScript 엔지니어로 일하면서 아름답고 강력한 부동산 소프트웨어를 만들고 있다. Karl은 새로운 기술에 대한 열정이 있고, 새로운 방법으로 일하기 위해 공부하는 것을 즐긴다. 자신의 웹사이트인 karlmikko.com에서 그를 찾을 수 없다면, 아내와 함께 암벽등반을 하고 있거나 커피를 즐기고 있을 것이다.

지은이_ **존 비비**

Dave Ramsey의 디지털 개발팀에서 애플리케이션을 개발하고 있다. 웹과 iOS를 위해 사용자를 대하는 기술에 집중하고 있다. Final Cut Pro와 Motion에 사용하는 플러그인과 PHP 웹서비스를 만들기도 했다. Beebe가 예술과 코드에 관한 언어를 섞는 날은 좋은 날이다. 그는 독서광이고, 사진을 좋아하며, 매일 아내의 기대 이상을 달성하기 위해 노력하고 있다.

지은이_ **프랑키 반야르디**

여러 고객을 위해 사용자 경험을 만드는 시니어 프론트엔드 개발자이다. 여가는 StackOverflow(FakeRainBrigand)와 IRC(GreenJello)에 올라오는 질문에 답해주는 한편 작은 프로젝트도 즐긴다.

역자 소개

옮긴이_ **곽현철**

NHN Technology Services에서 UI 개발자로 일하다가 지금은 티켓몬스터에서 프론트엔드 개발자로 일하고 있다. 좋은 동료가 되겠다는 핑계로 개발 욕심보다 개그 욕심이 많아서 주변에 웃음을 전하느라 바쁜, 조금 재미있는 사람. 아마 지금도 어디선가 농담을 던지고 있을 것이다.

옮긴이_ **김훈민**

NHN Technology Services에서 근무하던 시절 본격적으로 프론트엔드 개발에 입문하여 지금은 네이버에서 스마트에디터3를 개발하고 있다. 잠들어버린 http://huns.me 블로그를 운영하고 있으며 테스트 프로세스에 관심이 많다.

요즘 웹 프론트엔드는 그야말로 춘추전국시대다. 수많은 도구, 라이브러리, 프레임워크가 오늘 쏟아지고 내일 사라진다. SPA를 위한 Full Framework로서 한 시대를 풍미했던 Backbone은 벌써 낡은 기술이 되었다. Backbone이 지나간 자리는 구글을 등에 업은 Angular가 차지했다. Angular는 한때 천하를 통일할 가장 유력한 후보였다. 엄청난 기세로 개발자들의 마음을 사로잡으며 높은 점유율을 차지하는 데 성공했지만, 고질적인 성능 이슈와 완전히 새로운 버전인 Angular 2 발표로 인한 역풍 등으로 요즘은 기세가 한풀 꺾인 모습이다.

React는 Angular 이후, 최근 프론트엔드 영역에서 가장 많은 관심을 받는 기술이다. 처음 React가 등장했을 때 커뮤니티의 반응은 호의적이지 않았다. 많은 것이 불편했다. Virtual DOM, JSX 같은 개념은 낯설었고, 페이지 단위로 개발하는 데 익숙해 있던 개발자들은 컴포넌트 단위로 사고하기 힘들어했다. 더군다나 HTML과 JavaScript를 한 곳에 묶어두는 React의 개발 방식은 기능과 구조라는 두 개의 관심사를 분리하라는 오랜 불문율에 어긋났다.

하지만 Facebook은 포기하지 않았고, 전 세계 수억 명이 사용하는 자사 제품의 곳곳에 React를 심었다. 끊임없이 React의 철학을 이야기하며 설득했다. 시간이 지나면서 React에 호감을 보이는 개발자들이 늘기 시작했다. Airbnb, 넷플릭스, Atlassian, BBC 같은 거대 기업이 React를 자사 프로젝트에 사용했다. 분위기가 달라졌다. 때마침 국내에도 React 바람이 불었다. React Native의 등장이 한 몫 했다. "Learn once, Write everywhere"라는 캐치프레이즈에 많은 개발자가 설렜다. 그렇게 React는 프론트엔드를 넘어 모바일 앱까지 조금씩 스며들었다. React 관련 커뮤니티가 문을 열었고, React를 주제로 하는 블로그 포스트가 하나둘 등장했다. 국내외에서 부지런히 영역을 넓힌 React는, 이제 설계부터 개발까지 다양한 플러그인과 라이브러리를 가진 차세대 프론트엔드 개발 도구로 많은 사랑을 받고 있다.

이 책은 React를 소개하지만, React를 처음 접하는 사람을 위한 책은 아니다. 간단한 튜토리얼 정도는 해봤으나 React의 장점이 뭔지 아직 아리송하다는 사람에게 더 적합하다. 6명의 저자가 다양한 주제로 React의 가치와 철학을 이야기한다. 여러 관점에서 바라본 React를 간접 경험해 볼 수 있다. 하지만 저자가 여러 명이다 보니 읽다 보면 전체 내용이 하나의 줄기로 매끄럽게 연결되어 있지 못한 듯한 느낌을 받을 수 있다. 사용자들이 처음부터 끝까지 직접 따라 해 볼 수 있을 만한 예제가 부족하다는 점도 아쉽다. 대신 React를 지탱하는 철학이 무엇인지, 어떻게 활용할 수 있는지를 고민하면서 이 책을 읽을 것을 추천한다. 설치 방법에 대한 설명이나 간단한 튜토리얼은 React 공식 문서(https://facebook.github.io/react/docs/getting-started.html)를 추천한다. 영어가 어렵다면 한국어로 번역한 사이트(http://reactkr.github.io/react/docs/getting-started-ko-KR.html)도 있다.

원서가 나온 후, 책을 번역하고 출간하기까지 여타의 사정으로 인해 시간이 지연되어 책이 쓰일 당시 0.11이었던 React의 버전이 이제는 0.14 RC까지 올라가 버렸다. 따라서 책에 있는 내용을 최신 버전에 그대로 적용할 수 없는 경우가 있다. 번역서라는 특성상 원문을 수정하기가 쉽지 않기 때문에 논의 끝에 React 0.14 RC를 반영하는 역자 주석을 추가하기로 했다. 또한, 원서에는 없는 React 릴리즈 로그를 부록으로 수록했다. 릴리즈 로그를 보면 지원을 중단하는 API와 React가 그동안 변해온 과정을 알 수 있다. React를 맛은 봤지만, 그 맛이 정확히 뭔지 잘 모르겠다는 분들이 이 책으로 인해 React와 더 친해질 수 있기를 바란다.

끝으로 전문 직업인으로서 배울 점이 많은 존경하는 상훈 님, 개발자로서 성장하는 데 많은 영감과 지식을 준 대선이 형, 방황하던 내 인생에 반전을 만들어준 태훈 님, 지칠 때마다 끊임없이 열정을 불어넣어 주는 우영이, 함께 번역하고 교정하느라 고생한 능력자 현철 님, 한결같이 옆에서 나를 응원해주는 지원이에게 고맙다는 말을 전한다. 이

책을 선택한 모든 독자의 앞날에 축복이 가득하길 바라며 서문을 마친다.

_김훈민

최근 몇 년간 Javascript 개발 환경의 눈부신 발전만큼, HTML/CSS를 이용한 UI 개발도 빠르게 변화해왔다. Bootstrap 같은 프레임워크의 사용은 이제 흔한 일이 되었고, BEM, OOCSS 같은 CSS 방법론들이 UI를 바라보는 새로운 시야를 제공하고 있다. 이런 영향으로 UI 개발도 페이지 또는 화면 단위의 개발 방식에서 탈피하여, 컴포넌트라고 부를 만한 재사용성이 높은 UI를 바탕으로 한 스타일 가이드 기반의 개발로 점차 모양새가 바뀌고 있다.

React는 이런 새로운 물결과 잘 어울린다. 컴포넌트와 상태를 중심으로 UI를 개발하면, 화면 구성을 위한 복잡한 논리 구조를 단순하게 풀어낼 수 있다. React에 대한 이해는 UI의 분리를 위해 선행되어야 할 생각의 분리에도 도움을 준다. React가 가져다주는 이런 단편적인 생각의 전환만으로도 자신의 생산성이 높아지는 것을 느끼게 될 것이다.

끝으로 엉성한 번역을 함께 살펴봐 주고 글로 만들어준 훈민 님, 번역의 길에 다리를 놓아준 상훈 님, 개발의 길을 열어준 형국이형, 고락을 함께 하고 있는 표준화개발유닛 동료들, 그리고 늘 나를 웃게 해주는 영원이에게 감사 인사를 전한다.

_곽현철

React는 무엇이고 왜 써야 할까?

React는 Facebook에서 내부적으로 개발한 JavaScript 라이브러리로 2013년에 오픈 소스로 공개되었다. 웹에서 상호작용하는 사용자 인터페이스를 만들기 위한 라이브러리이다. React는 브라우저 DOM을 다루는 새로운 방법을 소개했다. 확장성과 새로운 기능의 개발을 위해 수동으로 DOM을 갱신하고 어렵게 각 상태를 추적하는 노력은 이제 옛날이야기가 되었다.

React는 매우 새로운 방법으로 DOM을 다룬다. 아무 때나 선언적으로 사용자 인터페이스를 정의할 수 있다. React는 데이터가 변경되었을 때 어떤 부분의 DOM을 갱신할지 신경 쓰지 않는다. 언제든지 최소한의 DOM 수정 때문에 전체 애플리케이션을 재 렌더링한다.

이 책에서 얻을 수 있는 것

React는 현재의 방법들에 도전하는 새롭고 흥미로운 개념을 소개한다. 이 책은 이런 모든 개념을 살펴보고, 이런 개념들이 유용한 이유를 설명해준다. 단일 페이지 애플리케이션SPAs, Single Page Applications를 만드는 데 특히 도움이 될 것이다.

React는 애플리케이션의 "view"에만 집중한다. 서버 통신이나 코드 조직에는 전혀 관심을 두지 않는다. 이 책에서는 React를 이용해서 완전한 애플리케이션을 만들기 위한 모범 예제와 대체 도구도 설명한다.

이 책을 읽기 위해 알아야 할 것

이 책의 내용을 이해하기 위해서는 JavaScript와 HTML을 다뤄본 경험이 있어야 한다. 프레임워크의 종류와 상관없이 단일 페이지 애플리케이션을 다뤄본 경험이 있다

면 더욱 도움이 될 것이다. 물론 필수적인 것은 아니다.

소스 코드와 예제 애플리케이션

이 책에서는 예제 애플리케이션으로 설문조사 생성기Survey Builder를 인용한다. 전체 코드는 Github의 저장소 https://github.com/backstopmedia/bleeding-edge-sample-app에서 확인할 수 있다.

버전 이슈에 대한 역자주

이 책은 React v.0.11.1~v.0.12를 기준으로 집필되었다. Github 저장소에 있는 샘플 예제 코드는 React v.0.11.1을 사용한다. 이 책을 번역하는 현재(2015. 9. 8) 가장 안정화된 최신 버전은 v.0.13.3으로, 버전이 업데이트하면서 인터페이스나 정책이 바뀐 부분이 있어 책에 나와 있는 내용을 최신 버전에 그대로 적용할 수 없는 경우가 있다. 독자의 혼란을 막기 위해 최신 버전에서 호환되지 않는 부분마다 역자주를 삽입했고, 부록에 릴리즈 노트를 수록했으니 참고하기 바란다.

차례

chapter 1

React 소개

1.1 배경

웹 애플리케이션이 막 등장한 시절에는 클라이언트가 페이지 전체를 서버에 요청하는 방식으로만 웹 애플리케이션을 개발할 수 있었다. 이렇게 개발한 애플리케이션은 브라우저에서 일어나는 이벤트를 처리할 필요가 없었기 때문에 구조가 아주 단순했다.

PHP 같은 언어를 사용하면 이런 형태의 애플리케이션을 쉽게 만들 수 있었다. 기능 PHP로 컴포넌트functional components를 만들면, 재사용성과 가독성이 높은 코드를 쉽게 작성할 수 있었고, PHP는 이런 단순함 덕분에 많은 인기를 얻었다.

하지만 이 방식으로는 사용자에게 멋진 경험을 줄 수 없었다. 사용자가 무언가를 할 때마다 매번 서버에 요청을 보내고 응답을 기다려야 했다. 심지어 서버로부터 응답이 오면 지금까지 페이지에서 사용자가 작업한 내용은 모두 날아가 버렸다.

더 나은 사용자 경험을 제공하기 위해서 사람들은 브라우저에서 애플리케이션을 렌더링하는 JavaScript 기반 라이브러리를 만들기 시작했다. 단순한 HTML 템플릿부터 애플리케이션 전체를 제어하는 시스템까지, 다양한 방법으로 DOM을 제어하는 라이브러리가 등장했다. 다양한 JavaScript 라이브러리를 이용하면서 애플리케이션은 점점 더 크고 복잡해졌다. 길게 헝클어진 실처럼 꼬여있는 이벤트로 무장한 애플리케이션의 동작을 설명하기란 쉽지 않다. 그래서 앞에서 이야기한 PHP의

경우에 비해, 요즘의 클라이언트 애플리케이션은 만들기가 매우 어려워졌다.

React는 Facebook이 사용하는 PHP프레임워크인 XHP를 대체하기 위해 시작되었다. PHP 프레임워크인 XHP는 새로운 요청이 들어올 때마다 전체 페이지를 렌더링한다. 이 작업을 클라이언트에서 처리하기 위해 React가 탄생했다.

React는 기본적으로 복잡한 상태 변화를 잘 관리할 수 있게 해주는 상태 시스템state machine이다. React의 관심사는 오직 두 가지다.

1. DOM 업데이트
2. 이벤트 응답

React는 Ajax, 라우팅, 저장소, 데이터 구조에 관심이 없으며 MVCModel-View-Controller 프레임워크가 아니다. 굳이 MVC를 가지고 이야기하자면, MVC에서 V를 담당한다고 볼 수 있다. 가벼운 React는 다양한 시스템에 자유롭게 어울릴 수 있다. 실제로 몇몇 인기 있는 MVC 프레임워크가 View를 렌더링할 때 React를 사용한 적이 있다.

JavaScript로 DOM을 가져오고 갱신하는 작업은 느리므로 상태가 바뀔 때마다 페이지 전체를 렌더링하는 애플리케이션은 성능이 떨어질 수밖에 없다. React는 가상 DOMvirtual DOM을 이용해서 DOM을 읽지 않고 갱신할 수 있는 아주 강력한 렌더링 시스템을 가지고 있다.

React의 핵심은 렌더링 함수다. 이 함수는 현재의 상태 값을 결과 페이지를 나타내는 가상 표현 객체virtual representation으로 변환한다. 마치 고성능 3D 게임 엔진 같다. 상태 변경을 감지한 React는 이 함수를 실행해서 새로운 가상 표현 객체를 만든 다음에 이것을 DOM에 전달해 새로운 상태를 반영한다.

언뜻 보면, JavaScript가 필요할 때 개별 요소를 갱신하는 것보다 느릴 것 같다. 하지만 React는 현재의 가상 표현 객체와 새로운 가상 표현 객체의 차이를 아주

효과적으로 비교하는 알고리즘을 가지고 있다. 이 알고리즘을 이용해서 DOM을 최소한으로 변경한다.

리플로우reflow나 불필요한 DOM 조작mutation 같이 흔한 성능 저해 요소를 최소화함으로써 성능을 끌어올린다.

인터페이스가 커질수록, 하나의 상호 작용interaction이 다른 상호 작용을 연쇄적으로 부르는 경우가 많아진다. 이런 연쇄 호출을 적절히 처리하지 않으면 애플리케이션의 성능이 급격히 떨어진다. 심지어, 최종 상태에 도달할 때까지 같은 DOM 엘리먼트를 몇 번이고 다시 변경하는 경우도 있다.

React 가상 표현 객체는 단일 패스에서 최소한의 변경을 수행함으로써 이런 문제를 최소화한다. 이를 통해 애플리케이션의 유지 보수성도 높인다. 사용자 입력이나 외부의 변경에 의한 상태 변화가 있다는 사실을 React에게 알려주기만 하면 나머지는 React가 알아서 처리한다. 개발자가 프로세스를 세세히 관리할 필요가 없다.

React는 모든 이벤트를 하나의 이벤트 핸들러에 위임delegate함으로써 이벤트 핸들러가 여러 개일 때 발생할 수 있는 성능 저하의 위험을 줄인다.

이 책에서 사용하는 설문조사 생성기Survey Builder 예제는 Github 저장소(https://github.com/backstopmedia/bleeding-edge-sample-app)에서 자세히 볼 수 있다.

1.2 개요

이 책은 React를 이용한 최신 개발 기법을 크게 네 가지 주제로 나눠서 다룬다.

컴포넌트 생성 및 구성

1장부터 7장까지는 React 컴포넌트를 생성하고 구성하는 방법을 설명한다. 여기에서는 React 사용방법을 배운다.

1) React 소개

1장은 배경과 이 책의 전체 개요를 설명하고 React를 소개한다.

2) JSX와 기본 React 구성요소 사용하기

JSXJavaScript XML를 이용하면 JavaScript 코드 안에 XML 스타일의 문법 코드를 작성할 수 있다. JSX를 사용하는 방법과 이를 이용해서 기본적인 React 컴포넌트를 만드는 방법을 학습한다. React 컴포넌트를 개발할 때 반드시 JSX를 같이 사용해야 하는 것은 아니지만, 권장하고 싶은 방법이라는 생각에 이 책에 있는 대부분의 예제는 JSX를 사용해서 작성했다.

3) React 구성요소의 라이프 사이클

React는 렌더링 과정 중에 자주 컴포넌트를 생성하고 제거하며, 컴포넌트 라이프 사이클에 접근할 수 있는 다양한 함수를 제공한다. 이 라이프 사이클을 잘 이해하면 애플리케이션 메모리 누수를 방지할 수 있다.

4) React의 데이터 흐름

React가 컴포넌트 트리에 데이터를 어떻게 전달하는지, 어떤 데이터를 변경해도 안전한지 잘 알아둬야 한다. React는 속성props과 상태state를 아주 분명하게 구분한다. 이 장은 속성과 상태가 무엇인지 알아보고, 컴포넌트 개발 시에 이 둘을 제대로 사용하는 방법을 설명한다.

5) 이벤트 핸들링

React의 이벤트 처리는 선언적이다. 이벤트 처리는 동적 UI를 구성하는 데 중요하므로 완벽하게 익히는 게 좋다. 다행히 React를 이용하면 이벤트 처리를 아주 간단하게 할 수 있다.

6) 컴포넌트의 구성

React를 이용하면 특정 작업을 수행하는 작지만 정교한 컴포넌트를 만들 수 있

다. 이렇게 만든 컴포넌트를 구성해서 오케스트레이션 레이어orchestration layers[01]를 만든다. 이 장은 한 컴포넌트가 다른 컴포넌트를 사용하는 방법을 설명한다.

7) React 믹스인

여러 React 컴포넌트가 사용하는 공통 기능을 공유하는 방법인 믹스인을 사용하면 컴포넌트를 더 작게 만들 수 있어서 관리하기 편하다.

고급 주제

기본을 배웠으니 이제 더 수준 높은 주제로 넘어간다. 8장부터 13장까지 React 개발 기술을 연마하여 더 훌륭한 React 컴포넌트를 만들어 본다.

8) React에서 DOM에 접근하기

기존 JavaScript 라이브러리를 사용하거나 컴포넌트를 더 깊게 제어하기 위해서 때로는 React 가상 DOM이 아닌 진짜 DOM을 이용해야 할 때가 있다. 이 장은 안전하게 DOM에 접근할 수 있는 React 컴포넌트의 라이프 사이클이 어디인지, DOM 제어를 언제 해제하여 메모리 누수를 막아야 하는지를 설명한다.

9) React로 폼 요소 다루기

HTML 폼 엘리먼트는 사용자 입력을 받는 대표적인 방법이다. HTML 폼 엘리먼트는 상태를 갖는다. React를 이용하면 놀라운 방법으로 폼의 상태를 React 컴포넌트에 전달할 수 있다.

10) 애니메이션

CSS를 이용하면 고성능 애니메이션을 만들 수 있다. React는 CSS 애니메이션 처리를 도와준다. 이 장은 React를 이용해서 CSS 애니메이션을 처리하는 방법을 설명한다.

---

01 **역자주_** 소프트웨어 개발에서 오케스트레이션 레이어(Orchestration Layer, OL)는 모델링한 데이터 요소와 기능을 구체화하는 추상 레이어를 말한다. 작게 나눈 React 컴포넌트를 실제 제품에 적용하기 위해서는 개별 컴포넌트가 서로 조화롭게 동작하여 하나의 콘텐츠를 표현할 수 있게 제어하는 단계가 필요하다. 이 책에서 말하는 오케스트레이션 레이어는 이 단계를 의미한다.

11) 성능 개선과 컴포넌트

React 가상 DOM은 뛰어난 성능을 보여주지만, 언제나 그렇듯이 개선할 부분이 있다. React는 변경이 없는 컴포넌트를 브라우저가 다시 렌더링 하는 것을 막음으로써 애플리케이션의 속도를 비약적으로 높인다.

12) 서버 사이드 렌더링

많은 애플리케이션이 SEO를 적용한다. React는 Node.js처럼 브라우저가 아닌 환경에서도 문자열로 렌더링할 수 있다. 서버 측 렌더링을 이용하면 애플리케이션의 시작 페이지 로딩 시간을 줄일 수 있다. 서버와 클라이언트 렌더링 방식을 함께 사용하는 것은 어려울 수 있다. 이 장에서는 두 가지 렌더링 방식을 함께 적용하는 전략을 설명하고, 서버 측 렌더링을 처리할 때 발생할 수 있는 복잡한 상황을 자세히 알아본다.

13) React 패밀리의 다른 JavaScript 라이브러리 사용하기

Facebook은 React 외에도, React와 함께 사용할 수 있는 여러 오픈 소스 개발 도구를 계속해서 공개하고 있다. 이 장을 학습하면 이런 라이브러리를 React 패밀리에 잘 적용하기 위한 통찰을 얻을 수 있다.

React를 위한 도구 다루기

React는 뛰어난 개발, 테스트 도구를 제공한다. 이 도구를 사용하면 견고한 애플리케이션을 만들 수 있다. 14, 15장에서 개발자 도구와 테스트 방법을 알아본다.

14) 개발자 도구

React 애플리케이션의 규모가 커지면 배포를 위해 코드 패키징 과정을 자동화할 필요가 있으며 코드 디버깅이 점점 어려질 것이다. 이 장에서는 이런 고민을 덜어주는 React 애플리케이션 패키징 도구를 살펴본다. 그리고 더 쉬운 디버깅을 위해 구글 크롬 플러그인으로 React 컴포넌트를 시각화하는 방법도 알아본다.

15) React를 위한 테스트 코드 작성하기

애플리케이션의 규모가 점점 커지는 상황에서, 기존 코드에 버그를 추가하지 않으려면 테스트 코드를 작성하는 것이 좋다. 테스트 코드는 더 나은 모듈화 코드를 작성하는 데 도움이 된다. 이 장에서는 React 컴포넌트의 모든 부분을 테스트하는 방법을 알아본다.

React 활용하기

마지막 장은 React를 활용하는 데 있어서 중요한 부분을 살펴보고, 미처 생각해 보지 못한 다른 사용 사례를 설명한다.

16) Architectural patterns

React는 "MVC" 중에서 "V"만을 제공한다. 그래서 다른 프레임워크나 시스템에 매우 유연하게 적용할 수 있다. 이 장은 React를 이용해서 규모가 큰 애플리케이션을 설계하는 과정을 설명한다.

17) React의 다른 사용 사례

React는 웹 환경에 초점을 맞추고 있지만, 웹이 아니더라도 JavaScript를 지원하는 환경이라면 어디든 사용할 수 있다. 이 장에서는 전통적인 웹 환경이 아닌 곳에서 React를 사용하는 방법을 알아본다.