

1장 리액트 시작 033

1.1 왜 리액트인가? 034

1.1.1 리액트 이해 036

1.2 리액트의 특징 039

1.2.1 Virtual DOM 039

1.2.2 기타 특징 042

1.3 작업 환경 설정 043

1.3.1 Node.js와 npm 043

1.3.2 yarn 046

1.3.3 에디터 설치 048

1.3.4 Git 설치 051

1.3.5 create-react-app으로 프로젝트 생성하기 051

2장 JSX 055

2.1 코드 이해하기 056

2.2 JSX란? 059

2.3 JSX의 장점 060

2.3.1 보기 쉽고 익숙하다 060

2.3.2 더욱 높은 활용도 061

2.4 JSX 문법 061

2.4.1 감싸인 요소 062

2.4.2 자바스크립트 표현 064

2.4.3 If 문 대신 조건부 연산자 066

2.4.4 AND 연산자(&&)를 사용한 조건부 렌더링 067

2.4.5 undefined를 렌더링하지 않기 069

2.4.6 인라인 스타일링 070

2.4.7 class 대신 className 072

2.4.8 꼭 달아야 하는 태그 073

2.4.9 주석 075

2.5 ESLint와 Prettier 적용하기 077

2.5.1 ESLint 077

2.5.2 Prettier 079

2.6 정리 082

3장 컴포넌트 083

3.1 클래스형 컴포넌트 085

3.2 첫 컴포넌트 생성 087

3.2.1 src 디렉터리에 MyComponent.js 파일 생성 087

3.2.2 코드 작성하기 088

3.2.3 모듈 내보내기 및 불러오기 091

3.3 props 092

3.3.1 JSX 내부에서 props 렌더링 092

3.3.2 컴포넌트를 사용할 때 props 값 지정하기 092

3.3.3 props 기본값 설정: defaultProps 093

3.3.4 태그 사이의 내용을 보여 주는 children 094

3.3.5 비구조화 할당 문법을 통해 props 내부 값 추출하기 096

3.3.6 propTypes를 통한 props 검증 097

3.3.7 클래스형 컴포넌트에서 props 사용하기 102

3.4 state 103

3.4.1 클래스형 컴포넌트의 state 104

3.4.2 함수형 컴포넌트에서 useState 사용하기 111

3.5 state를 사용할 때 주의 사항 115

3.6 정리 116

4장 이벤트 핸들링 119

4.1 리액트의 이벤트 시스템 121

4.1.1 이벤트를 사용할 때 주의 사항 122

4.1.2 이벤트 종류 123

4.2 예제로 이벤트 핸들링 익히기 123

- 4.2.1 컴포넌트 생성 및 불러오기 124
- 4.2.2 onChange 이벤트 핸들링하기 125
- 4.2.3 임의 메서드 만들기 130
- 4.2.4 input 여러 개 다루기 133
- 4.2.5 onKeyPress 이벤트 핸들링 136

4.3 함수형 컴포넌트로 구현해 보기 137

4.4 정리 140

5장 ref: DOM에 이름 달기 141

5.1 ref는 어떤 상황에서 사용해야 할까? 143

- 5.1.1 예제 컴포넌트 생성 145
- 5.1.2 App 컴포넌트에서 예제 컴포넌트 렌더링 146
- 5.1.3 DOM을 꼭 사용해야 하는 상황 147

5.2 ref 사용 147

- 5.2.1 콜백 함수를 통한 ref 설정 147
- 5.2.2 createRef를 통한 ref 설정 148
- 5.2.3 적용 149

5.3 컴포넌트에 ref 달기 151

- 5.3.1 사용법 151
- 5.3.2 컴포넌트 초기 설정 152
- 5.3.3 컴포넌트에 메서드 생성 154
- 5.3.4 컴포넌트에 ref 달고 내부 메서드 사용 155

5.4 정리 156

6장 컴포넌트 반복 157

6.1 자바스크립트 배열의 map() 함수 159

- 6.1.1 문법 159
- 6.1.2 예제 159

6.2 데이터 배열을 컴포넌트 배열로 변환하기 160

- 6.2.1 컴포넌트 수정하기 160
- 6.2.2 App 컴포넌트에서 예제 컴포넌트 렌더링 161

6.3 key 162

- 6.3.1 key 설정 163

6.4 응용 164

- 6.4.1 초기 상태 설정하기 164
- 6.4.2 데이터 추가 기능 구현하기 165
- 6.4.3 데이터 제거 기능 구현하기 168

6.5 정리 170

7장 컴포넌트의 라이프사이클 메서드 171

7.1 라이프사이클 메서드의 이해 172

7.2 라이프사이클 메서드 살펴보기 175

- 7.2.1 render() 함수 175
- 7.2.2 constructor 메서드 176
- 7.2.3 getDerivedStateFromProps 메서드 176
- 7.2.4 componentDidMount 메서드 176
- 7.2.5 shouldComponentUpdate 메서드 176
- 7.2.6 getSnapshotBeforeUpdate 메서드 177
- 7.2.7 componentDidUpdate 메서드 177
- 7.2.8 componentWillUnmount 메서드 178
- 7.2.9 componentDidCatch 메서드 178

7.3 라이프사이클 메서드 사용하기 179

- 7.3.1 예제 컴포넌트 생성 179
- 7.3.2 App 컴포넌트에서 예제 컴포넌트 사용 181
- 7.3.3 에러 잡아내기 184

7.4 정리 188

8장 Hooks 189

8.1 useState 190

8.1.1 useState를 여러 번 사용하기 192

8.2 useEffect 194

8.2.1 마운트될 때만 실행하고 싶을 때 195

8.2.2 특정 값이 업데이트될 때만 실행하고 싶을 때 196

8.2.3 뒷정리하기 197

8.3 useReducer 200

8.3.1 카운터 구현하기 200

8.3.2 인풋 상태 관리하기 202

8.4 useMemo 204

8.5 useCallback 208

8.6 useRef 210

8.6.1 로컬 변수 사용하기 211

8.7 커스텀 Hooks 만들기 213

8.8 다른 Hooks 214

8.9 정리 215

9장 컴포넌트 스타일링 217

9.1 가장 흔한 방식, 일반 CSS 219

9.1.1 이름 짓는 규칙 221

9.1.2 CSS Selector 221

9.2 Sass 사용하기 223

9.2.1 utils 함수 분리하기 227

9.2.2 sass-loader 설정 커스터마이징하기 228

9.2.3 node_modules에서 라이브러리 불러오기 232

9.3 CSS Module 234

9.3.1 classnames 238

- 9.3.2 Sass와 함께 사용하기 240
- 9.3.3 CSS Module이 아닌 파일에서 CSS Module 사용하기 241

9.4 styled-components 241

- 9.4.1 Tagged 템플릿 리터럴 245
- 9.4.2 스타일링된 엘리먼트 만들기 246
- 9.4.3 스타일에서 props 조회하기 247
- 9.4.4 props에 따른 조건부 스타일링 248
- 9.4.5 반응형 디자인 249

9.5 정리 252

10장 일정 관리 웹 애플리케이션 만들기 253

10.1 프로젝트 준비하기 255

- 10.1.1 프로젝트 생성 및 필요한 라이브러리 설치 255
- 10.1.2 Prettier 설정 255
- 10.1.3 index.css 수정 256
- 10.1.4 App 컴포넌트 초기화 256

10.2 UI 구성하기 257

- 10.2.1 TodoTemplate 만들기 258
- 10.2.2 TodoInsert 만들기 262
- 10.2.3 TodoListItem과 TodoList 만들기 266

10.3 기능 구현하기 270

- 10.3.1 App에서 todos 상태 사용하기 270
- 10.3.2 항목 추가 기능 구현하기 274
- 10.3.3 지우기 기능 구현하기 281
- 10.3.4 수정 기능 284

10.4 정리 287

11장 컴포넌트 성능 최적화 289

11.1 많은 데이터 렌더링하기 290

11.2 크롬 개발자 도구를 통한 성능 모니터링 292

- 11.3** 느려지는 원인 분석 294
- 11.4** React.memo를 사용하여 컴포넌트 성능 최적화 295
- 11.5** onToggle, onRemove 함수가 바뀌지 않게 하기 296
 - 11.5.1 useState의 함수형 업데이트 296
 - 11.5.2 useReducer 사용하기 300
- 11.6** 불변성의 중요성 302
- 11.7** TodoList 컴포넌트 최적화하기 304
- 11.8** react-virtualized를 사용한 렌더링 최적화 305
 - 11.8.1 최적화 준비 306
 - 11.8.2 TodoList 수정 307
 - 11.8.3 TodoListItem 수정 308
- 11.9** 정리 310

12장 immer를 사용하여 더 쉽게 불변성 유지하기 311

- 12.1** immer를 설치하고 사용법 알아보기 313
 - 12.1.1 프로젝트 준비 313
 - 12.1.2 immer를 사용하지 않고 불변성 유지 313
 - 12.1.3 immer 사용법 316
 - 12.1.4 App 컴포넌트에 immer 적용하기 318
 - 12.1.5 useState의 함수형 업데이트와 immer 함께 쓰기 320
- 12.2** 정리 322

13장 리액트 라우터로 SPA 개발하기 323

- 13.1** SPA란? 324
 - 13.1.1 SPA의 단점 325
- 13.2** 프로젝트 준비 및 기본적인 사용법 326
 - 13.2.1 프로젝트 생성 및 라이브러리 설치 327
 - 13.2.2 프로젝트에 라우터 적용 327

- 13.2.3 페이지 만들기 328
- 13.2.4 Route 컴포넌트로 특정 주소에 컴포넌트 연결 329
- 13.2.5 Link 컴포넌트를 사용하여 다른 주소로 이동하기 331

13.3 Route 하나에 여러 개의 path 설정하기 333

13.4 URL 파라미터와 쿼리 334

- 13.4.1 URL 파라미터 335
- 13.4.2 URL 쿼리 337

13.5 서브 라우트 339

13.6 리액트 라우터 부가 기능 342

- 13.6.1 history 342
- 13.6.2 withRouter 345
- 13.6.3 Switch 349
- 13.6.4 NavLink 350

13.7 정리 352

14장 외부 API를 연동하여 뉴스 뷰어 만들기 353

14.1 비동기 작업의 이해 354

- 14.1.1 콜백 함수 356
- 14.1.2 Promise 357
- 14.1.3 async/await 358

14.2 axios로 API 호출해서 데이터 받아 오기 359

14.3 newsapi API 키 발급받기 363

14.4 뉴스 뷰어 UI 만들기 366

- 14.4.1 NewsItem 만들기 367
- 14.4.2 NewsList 만들기 369

14.5 데이터 연동하기 371

14.6 카테고리 기능 구현하기 374

- 14.6.1 카테고리 선택 UI 만들기 375
- 14.6.2 API를 호출할 때 카테고리 지정하기 380

14.7 리액트 라우터 적용하기 382

14.7.1 리액트 라우터의 설치 및 적용 382

14.7.2 NewsPage 생성 383

14.7.3 Categories에서 NavLink 사용하기 384

14.8 usePromise 커스텀 Hook 만들기 386

14.9 정리 390

15장 Context API 391

15.1 Context API를 사용한 전역 상태 관리 흐름 이해하기 392

15.2 Context API 사용법 익히기 394

15.2.1 새 Context 만들기 394

15.2.2 Consumer 사용하기 395

15.2.3 Provider 397

15.3 동적 Context 사용하기 399

15.3.1 Context 파일 수정하기 399

15.3.2 새로워진 Context를 프로젝트에 반영하기 400

15.3.3 색상 선택 컴포넌트 만들기 403

15.4 Consumer 대신 Hook 또는 static contextType 사용하기 406

15.4.1 useContext Hook 사용하기 406

15.4.2 static contextType 사용하기 407

15.5 정리 411

16장 리덕스 라이브러리 이해하기 413

16.1 개념 미리 정리하기 414

16.1.1 액션 415

16.1.2 액션 생성 함수 415

16.1.3 리듀서 416

16.1.4 스토어 416

16.1.5 디스패치 417

16.1.6 구독 417

16.2 리액트 없이 쓰는 리덕스 417

- 16.2.1 Parcel로 프로젝트 만들기 418
- 16.2.2 간단한 UI 구성하기 419
- 16.2.3 DOM 레퍼런스 만들기 420
- 16.2.4 액션 타입과 액션 생성 함수 정의 421
- 16.2.5 초깃값 설정 422
- 16.2.6 리듀서 함수 정의 422
- 16.2.7 스토어 만들기 424
- 16.2.8 render 함수 만들기 424
- 16.2.9 구독하기 425
- 16.2.10 액션 발생시키기 426

16.3 리덕스의 세 가지 규칙 427

- 16.3.1 단일 스토어 427
- 16.3.2 읽기 전용 상태 427
- 16.3.3 리듀서는 순수한 함수 427

16.4 정리 428**17장 리덕스를 사용하여 리액트 애플리케이션 상태 관리하기 429****17.1 작업 환경 설정 430****17.2 UI 준비하기 431**

- 17.2.1 카운터 컴포넌트 만들기 432
- 17.2.2 할 일 목록 컴포넌트 만들기 433

17.3 리덕스 관련 코드 작성하기 436

- 17.3.1 counter 모듈 작성하기 437
- 17.3.2 todos 모듈 만들기 439
- 17.3.3 루트 리듀서 만들기 442

17.4 리액트 애플리케이션에 리덕스 적용하기 442

- 17.4.1 스토어 만들기 443
- 17.4.2 Provider 컴포넌트를 사용하여 프로젝트에 리덕스 적용하기 443
- 17.4.3 Redux DevTools의 설치 및 적용 444

17.5 컨테이너 컴포넌트 만들기 446

- 17.5.1 CounterContainer 만들기 447
- 17.5.2 TodosContainer 만들기 453

17.6 리덕스 더 편하게 사용하기 457

17.6.1 redux-actions 457

17.6.2 immer 462

17.7 Hooks를 사용하여 컨테이너 컴포넌트 만들기 464

17.7.1 useSelector로 상태 조회하기 464

17.7.2 useDispatch를 사용하여 액션 디스패치하기 465

17.7.3 createStore를 사용하여 리덕스 스토어 사용하기 466

17.7.4 TodosContainer를 Hooks로 전환하기 467

17.7.5 useActions 유틸 Hook을 만들어서 사용하기 468

17.7.6 connect 함수와의 주요 차이점 469

17.8 정리 470

18장 리덕스 미들웨어를 통한 비동기 작업 관리 471

18.1 작업 환경 준비 472

18.2 미들웨어란? 476

18.2.1 미들웨어 만들기 477

18.2.2 redux-logger 사용하기 480

18.3 비동기 작업을 처리하는 미들웨어 사용 481

18.3.1 redux-thunk 482

18.3.2 redux-saga 502

18.4 정리 519

19장 코드 스플리팅 521

19.1 자바스크립트 함수 비동기 로딩 525

19.2 React.lazy와 Suspense를 통한 컴포넌트 코드 스플리팅 528

19.2.1 state를 사용한 코드 스플리팅 528

19.2.2 React.lazy와 Suspense 사용하기 531

19.2.3 Loadable Components를 통한 코드 스플리팅 533

19.3 정리 536

20장 서버 사이드 렌더링 537

20.1 서버 사이드 렌더링의 이해 538

- 20.1.1 서버 사이드 렌더링의 장점 539
- 20.1.2 서버 사이드 렌더링의 단점 539
- 20.1.3 서버 사이드 렌더링과 코드 스플리팅 충돌 540

20.2 프로젝트 준비하기 541

- 20.2.1 컴포넌트 만들기 541
- 20.2.2 페이지 컴포넌트 만들기 543

20.3 서버 사이드 렌더링 구현하기 545

- 20.3.1 서버 사이드 렌더링용 엔트리 만들기 545
- 20.3.2 서버 사이드 렌더링 전용 웹팩 환경 설정 작성하기 546
- 20.3.3 빌드 스크립트 작성하기 553
- 20.3.4 서버 코드 작성하기 555
- 20.3.5 정적 파일 제공하기 558

20.4 데이터 로딩 562

- 20.4.1 redux-thunk 코드 준비하기 563
- 20.4.2 Users, UsersContainer 컴포넌트 준비하기 567
- 20.4.3 PreloadContext 만들기 570
- 20.4.4 서버에서 리덕스 설정 및 PreloadContext 사용하기 572
- 20.4.5 스크립트로 스토어 초기 상태 주입하기 574
- 20.4.6 redux-saga 코드 준비하기 577
- 20.4.7 User, UserContainer 컴포넌트 준비하기 581
- 20.4.8 redux-saga를 위한 서버 사이드 렌더링 작업 584
- 20.4.9 usePreloader Hook 만들어서 사용하기 587

20.5 서버 사이드 렌더링과 코드 스플리팅 589

- 20.5.1 라우트 컴포넌트 스플리팅하기 590
- 20.5.2 웹팩과 babel 플러그인 적용 591
- 20.5.3 필요한 청크 파일 경로 추출하기 593
- 20.5.4 loadableReady와 hydrate 595

20.6 서버 사이드 렌더링의 환경 구축을 위한 대안 597

- 20.6.1 Next.js 597
- 20.6.2 Razzle 598

20.7 정리 598

21장 백엔드 프로그래밍: Node.js의 Koa 프레임워크 599

21.1 소개하기 600

- 21.1.1 백엔드 600
- 21.1.2 Node.js 600
- 21.1.3 Koa 601

21.2 작업 환경 준비 602

- 21.2.1 Node 설치 확인 602
- 21.2.2 프로젝트 생성 602
- 21.2.3 ESLint와 Prettier 설정 604

21.3 Koa 기본 사용법 608

- 21.3.1 서버 띄우기 608
- 21.3.2 미들웨어 609

21.4 nodemon 사용하기 616

21.5 koa-router 사용하기 617

- 21.5.1 기본 사용법 618
- 21.5.2 라우트 파라미터와 쿼리 619
- 21.5.3 REST API 621
- 21.5.4 라우트 모듈화 622
- 21.5.5 posts 라우트 생성 623

21.6 정리 634

22장 mongoose를 이용한 MongoDB 연동 실습 635

22.1 소개하기 636

- 22.1.1 문서란? 637
- 22.1.2 MongoDB 구조 639
- 22.1.3 스키마 디자인 639

22.2 MongoDB 서버 준비 640

- 22.2.1 설치 640
- 22.2.2 MongoDB 작동 확인 642

22.3 mongoose의 설치 및 적용 642

22.3.1 .env 환경변수 파일 생성 643

22.3.2 mongoose로 서버와 데이터베이스 연결 644

22.4 esm으로 ES 모듈 import/export 문법 사용하기 645

22.4.1 기존 코드 ES Module 형태로 바꾸기 648

22.5 데이터베이스의 스키마와 모델 651

22.5.1 스키마 생성 651

22.5.2 모델 생성 654

22.6 MongoDB Compass의 설치 및 사용 655**22.7 데이터 생성과 조회 657**

22.7.1 데이터 생성 657

22.7.2 데이터 조회 660

22.7.3 특정 포스트 조회 662

22.8 데이터 삭제와 수정 663

22.8.1 데이터 삭제 663

22.8.2 데이터 수정 665

22.9 요청 검증 667

22.9.1 ObjectId 검증 667

22.9.2 Request Body 검증 669

22.10 페이지네이션 구현 673

22.10.1 가짜 데이터 생성하기 673

22.10.2 포스트를 역순으로 불러오기 676

22.10.3 보이는 개수 제한 677

22.10.4 페이지 기능 구현 677

22.10.5 마지막 페이지 번호 알려 주기 678

22.10.6 내용 길이 제한 680

22.11 정리 682**23장 JWT를 통한 회원 인증 시스템 구현하기 683****23.1 JWT의 이해 684**

23.1.1 세션 기반 인증과 토큰 기반 인증의 차이 684

23.2 User 스키마/모델 만들기 686

23.2.1 모델 메서드 만들기 687

23.2.2 스택 메서드 만들기 688

23.3 회원 인증 API 만들기 689

23.3.1 회원가입 구현하기 690

23.3.2 로그인 구현하기 694

23.4 토큰 발급 및 검증하기 696

23.4.1 비밀번호 설정하기 696

23.4.2 토큰 발급하기 697

23.4.3 토큰 검증하기 699

23.4.4 토큰 재발급하기 702

23.4.5 로그아웃 기능 구현하기 704

23.5 posts API에 회원 인증 시스템 도입하기 705

23.5.1 스키마 수정하기 705

23.5.2 posts 컬렉션 비우기 706

23.5.3 로그인했을 때만 API를 사용할 수 있게 하기 707

23.5.4 포스트 작성 시 사용자 정보 넣기 708

23.5.5 포스트 수정 및 삭제 시 권한 확인하기 709

23.6 username/tags로 포스트 필터링하기 712

23.7 정리 714

24장 프론트엔드 프로젝트: 시작 및 회원 인증 구현 715

24.1 작업 환경 준비하기 719

24.1.1 설정 파일 만들기 720

24.1.2 라우터 적용 721

24.1.3 스타일 설정 725

24.1.4 Button 컴포넌트 만들기 726

24.1.5 리덕스 적용 729

24.2 회원가입과 로그인 구현 732

24.2.1 UI 준비하기 732

24.2.2 리덕스로 폼 상태 관리하기 751

24.2.3 API 연동하기 758

- 24.2.4 회원가입 구현 769
- 24.2.5 로그인 구현 776
- 24.2.6 회원 인증 에러 처리하기 778

24.3 헤더 컴포넌트 생성 및 로그인 유지 787

- 24.3.1 헤더 컴포넌트 만들기 787
- 24.3.2 로그인 상태를 보여 주고 유지하기 795
- 24.3.3 로그아웃 기능 구현 804

24.4 정리 808

25장 프론트엔드 프로젝트: 글쓰기 기능 구현하기 809

25.1 에디터 UI 구현하기 810

25.2 에디터 하단 컴포넌트 UI 구현하기 813

- 25.2.1 TagBox 만들기 813

25.3 리덕스로 글쓰기 상태 관리하기 821

- 25.3.1 EditorContainer 만들기 822
- 25.3.2 TagBoxContainer 만들기 826
- 25.3.3 글쓰기 API 연동하기 830

25.4 정리 835

26장 프론트엔드 프로젝트: 포스트 조회 기능 구현하기 837

26.1 포스트 읽기 페이지 구현하기 838

- 26.1.1 PostViewer UI 준비하기 838
- 26.1.2 API 연동하기 842

26.2 포스트 목록 페이지 구현하기 847

- 26.2.1 PostList UI 준비하기 847
- 26.2.2 포스트 목록 조회 API 연동하기 855
- 26.2.3 HTML 필터링하기 861
- 26.2.4 페이지네이션 구현하기 866

26.3 정리 871

27장 프론트엔드 프로젝트: 수정/삭제 기능 구현 및 마무리 873

27.1 포스트 수정 874

- 27.1.1 PostActionButtons 컴포넌트 만들기 874
- 27.1.2 수정 버튼 클릭 시 글쓰기 페이지로 이동하기 878

27.2 포스트 삭제 887

27.3 react-helmet-async로 meta 태그 설정하기 892

27.4 프로젝트 마무리 897

- 27.4.1 프로젝트 빌드하기 897
- 27.4.2 koa-static으로 정적 파일 제공하기 898
- 27.4.3 더 할 수 있는 작업 899

27.5 정리 902

28장 그다음은? 903

28.1 리액트 관련 커뮤니티 904

- 28.1.1 국내 커뮤니티 904
- 28.1.2 해외 커뮤니티 905

28.2 책의 연장선 905

28.3 사이드 프로젝트 생성 905

찾아보기 906