

React

React 라우터(router)

- 다른 주소에 다른 화면을 보여주는 것
- 리액트 라우팅 라이브러리
 - React-router(react-router-dom)
 - Reach-router
 - Next.js
- 라우터 적용 과정
 - 컴포넌트 만들기
 - Route 컴포넌트로 특정 주소에 컴포넌트 연결
- Link 컴포넌트
 - 클릭하여 다른 주소로 이동시켜 주는 컴포넌트
 - Link 컴포넌트는 <a 태그로 이루어져 있지만 페이지 <a 태그와 달리 전환을 방지
- 서브라우트
 - 라우트 내부에 또 라우트를 정의하는 것
- History
 - 특정 버튼 눌렀을 때 뒤로 가거나, 로그인 후 화면전환, 다른 페이지로 이 탈하는 것 방지

Context API

- 전역적으로 사용할 데이터가 있을 때 유용한 기능
 - 사용자 로그인 정보, 앱 환경설정, 테마 등...
 - 리덕스, 리액트 라우터, styled-components 등이 context library를 기반으로 구현되어 있음
 - 리액트는 props 를 사용하여 데이터를 부모 컴포넌트에서 자식 컴포넌트로 전송하는 형식인데 자식 컴포넌트가 데이터를 필요로 하지 않고 손자 컴포넌트가 데이터를 필요로 할 경우 자식 컴포넌트는 단지 데이터를 전달만 하는 불필요한 코드를 작성하게 됨
 - 데이터의 공급자(Provider)와 소비자(Consumer)를 정의하고 데이터가 필요한 컴포넌트만 사용할 수 있게 구현
 - 하위 컴포넌트가 여러 개인 구조에서 유용
 - 몇번째 하위 컴포넌트인지 상관 없이 하위 컴포넌트에서 임포트해 필요한 데이터 사용
 - props 는 데이터가 부모에서 자식 컴포넌트로 단방향으로만 이동할 수 있다. Context를 사용하여 자식 컴포넌트에서 부모 컴포넌트 데이터를 변경 가능

Redux

- Context 와 마찬가지로 데이터를 필요한 컴포넌트에서만 요청해 사용할 수 있게 함
- 부모 컴포넌트에서 생성한 데이터를 컴포넌트 외부의 store에서 관리
 - 컴포넌트의 상태 업데이트 관련 로직을 다른 파일로 분리하여 관리 가능
- 컴포넌트의 위치에 상관없이 store에 접근해 데이터를 사용하고 변경 가능
- 전역 상태를 관리할 때 효과적
- Redux의 데이터 변경 과정
 - Store > Component > Action > Reducer > Store
 - Props를 store에 담아 컴포넌트로 전달하면 전달받은 컴포넌트에서 store에 접근할 수 있으며, 컴포넌트에서 dispatch 함수를 사용하여 store 데이터를 변경 가능
 - dispatch 함수는 reducer에 action을 전달하고 reducer에서는 action에 작성된 작업 내용을 읽어 store 데이터를 변경
 - 컴포넌트에서 dispatch 함수가 실행되면 reducer 함수는 action 값을 참조하여 작업을 실행

Redux 개념 정리

- 액션(Action)
 - 상태에 어떠한 변화가 필요한 경우 액션이 발생함
- 액션 생성 함수
 - 액션 객체를 만들어 주는 함수
- 리듀서(reducer)
 - 변화를 일으키는 함수
 - 액션을 만들어서 발생시키면 리듀서가 현재 상태와 전달받은 액션 객체를 파라미터로 받아온다. 그리고 두 값을 참고하여 새로운 상태를 만들어 반환
- 스토어(store)
 - 프로젝트에 리덕스를 적용하기 위해 만든다.
 - 한 프로젝트는 한 개의 스토어만 가진다.
 - 스토어 안에는 현재 애플리케이션 상태와 리듀서가 들어가 있고, 중요한 내장 함수를 가지고 있다
- 디스패치(dispatch)
 - 스토어의 내장 함수
 - 액션을 발생시키는 역할: dispatch(action)
- 구독(subscribe)
 - 스토어의 내장 함수
 - Subscribe 함수 내에 리스너 함수를 파라미터로 넣어서 호출하면 리스너 함수가 액션이 디스패치 되어 상태가 업데이트 될 때마다 호출됨

Component와 PureComponent

- 두 컴포넌트 모두 props와 state의 변경에 따라 render() 함수를 호출하지만 변경 기준이 서로 다름
- Component
 - 비교대상이 완전히 동일하지 않은 경우 변경 발생으로 간주
- PureComponent
 - 비교 대상의 값을 비교해 동일하지 않으면 변경 발생으로 간주
 - 변경 발생으로 불필요한 rendering() 함수 실행을 줄이고 페이지 성능 향상 가능
 - Shallow-equal 패키지
 - PureComponent에서 state 값의 변경을 비교하는 것과 같은 기능을 함수 제공
 - shallowEqualArrays() 함수

비동기 작업

- 작업을 동기적으로 처리하는 경우 요청이 끝날 때 까지 기다리는 동안 중지 상태가 되므로 다른 작업을 수행할 수 없음
- 비동기적으로 처리하는 경우 웹어플리케이션이 멈추기 않기 때문에 동시에 여러가지 요청을 처리할 수 있고 기다리는 과정에서 다른 함수도 호출 가능
- 자바스크립트에서 비동기 작업 수행할 때 주로 콜백(Callback) 함수를 사용
- Promise
 - 콜백 안에 또다른 콜백이 들어가는 콜백 지옥(callback helling)의 코드가 만들어지지 않도록 ES6에 새롭게 도입된 기능
 - Resolve, then, reject
 - .then을 사용하여 그 다음 작업을 설정하므로 콜백 지옥이 현성되지 않음

리덕스 미들웨어

- 리덕스를 사용하여 비동기 작업을 관리해야 할 경우 미들웨어를 사용하여 효율적인 상태 관리 가능
- 리덕스 미들웨어를 비동기 작업 관리 절차
 - 작업환경 준비
 - 미들웨어 만들기
 - Redux-logger 사용
 - 비동기 작업 관리
- 리덕스 미들웨어는 액션을 디스패치 했을 리듀서에서 처리하기에 앞서 사전에 지정된 작업 실행
- 액션과 리듀서의 중간자 역할
 - 액션 -> 미들웨어 -> 리듀서 -> 스토어
 - 전달 받은 액션의 콘솔 출력
 - 전달 받은 액션 정보를 기반으로 액션 취소
 - 다른 종류의 액션을 추가로 디스패치

Server side rendering

- UI를 서버에서 렌더링 하는 것
- 사용자가 웹서비스를 방문했을 때 서버 쪽에서 **초기 렌더링**을 대신하고 사용자가 html을 전달받으면 그 내부에 렌더링된 결물이 표시됨
- 장점
 - SPA는 검색엔진 크롤러와 같이 자바스크립트가 실행되지 않는 환경에서는 페이지가 제대로 나타나지 않으므로 서버에서 트라이언트 대신 렌더링을 해주면 검색 엔진이 페이지의 내용을 제대로 수집 가능
 - 초기 렌더링 성능 향상
 - 자바스크립트 파일 다운로드가 완료되지 않은 시점에도 html 상에 사용자가 볼 수 있는 콘텐츠가 있어서 대기 시간이 최소화되고 사용자 경험 (UX)도 향상됨
- 단점
 - 브라우저 할 일을 서버가 대신하므로 서버 리소스가 사용됨
 - 수 많은 사용자가 동시 접속할 경우 서버에 과부하 발생
 - 캐싱과 밸런싱으로 성능 최적화 필요

JWT를 통한 인증

- JWT(JSON Web Token)
 - 데이터가 JSON으로 이루어져 있는 토큰
 - 두 개체가 서로 안전하게 정보를 주고받을 수 있도록 웹 표준으로 정의된 기술
- 세션 기반 인증과 토큰 기반 인증
 - 세션 기반 인증
 - 사용자 로그인 – 서버에서 사용자 정보 조회 – 세션 id 발급
 - 발급된 id는 브라우저의 쿠키에 저장
 - 사용자가 다른 요청을 보낼 때 마다 서버는 세션 저장소에서 세션을 T 조회한 후 로그인 여부 결정
 - 토큰 기반 인증
 - 토큰은 로그인 이후 서버가 만들어 주는 문자열이며 이 문자열 안에 사용자의 로그인 정보가 들어 있고 해당 정보가 서버에서 발급되었음을 증명하는 서명 포함
 - 서명이 포함되어 있으므로 무결성(정보가 변경 또는 위조되지 않았음을 의미하는 성질) 보장
 - 서버에서 사용자 로그인 정보를 기억하기 위해 사용하는 리소스가 적은 장점

React에서 bootstrap 사용

- 프론트엔드 디자인을 쉽게 구현할 수 있도록 해주는 html, css, js 프레임워크인 bootstrap을 react에서 사용 가능
- Reactstrap
 - Alerts
 - Badge
 - Breadcrumbs
 - Button Dropdown
 - Button group
 - Buttons
 - Card
 - Carousel
 - Collapse
 - Fade
 - Form
 - Jumbotron
 - Input group
 - List Group
 - Modal
 - Navbar
 - Pagination
 - Popovers
 - Progress
 - Spinner
 - Table
 - Tab
 - sweetalert2

React 이벤트 처리

- Html 과 달리 이벤트에 camel case 사용
 - Onclick -> onClick
- 이벤트 종류
 - onClick
 - onChange
 - onMouseMove
 - onMouseOver
 - onMouseOut
 - onKeyDown, onKeyPress, onKeyUp
 - onSubmit