What is, How Work

REACT ROUTER

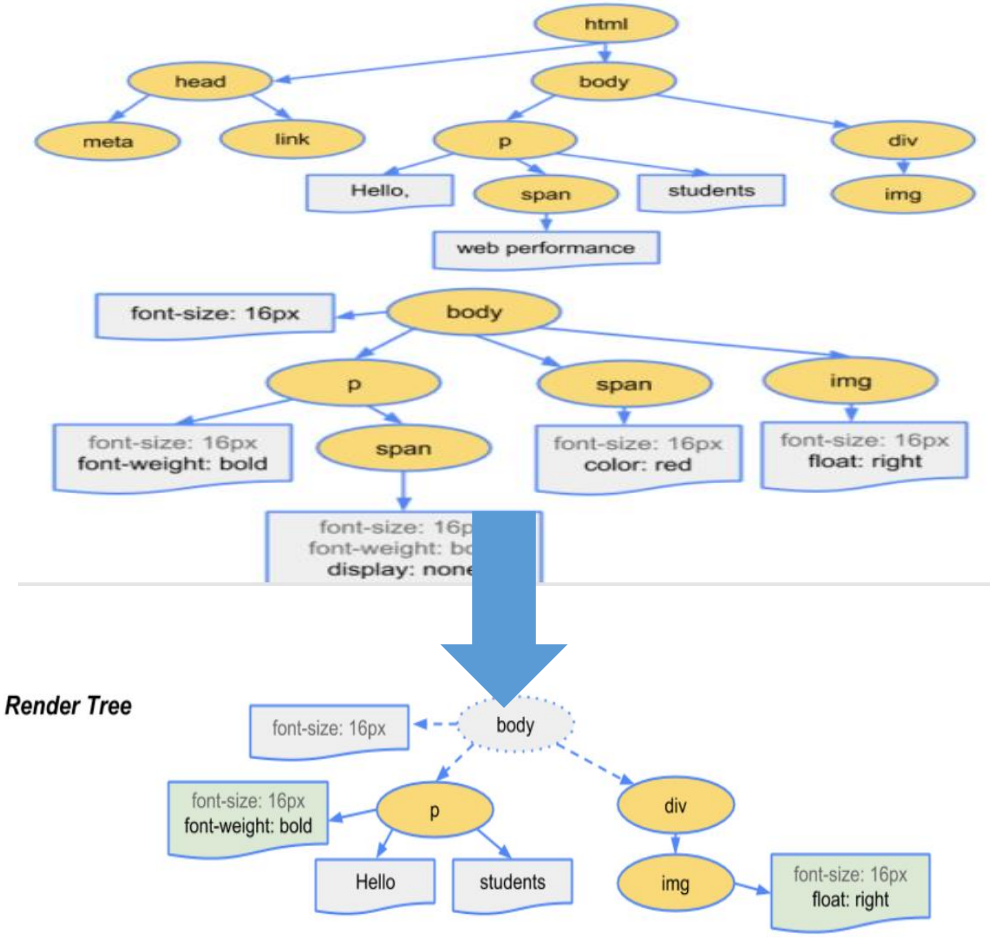
2021.06.17 (목) 권용빈

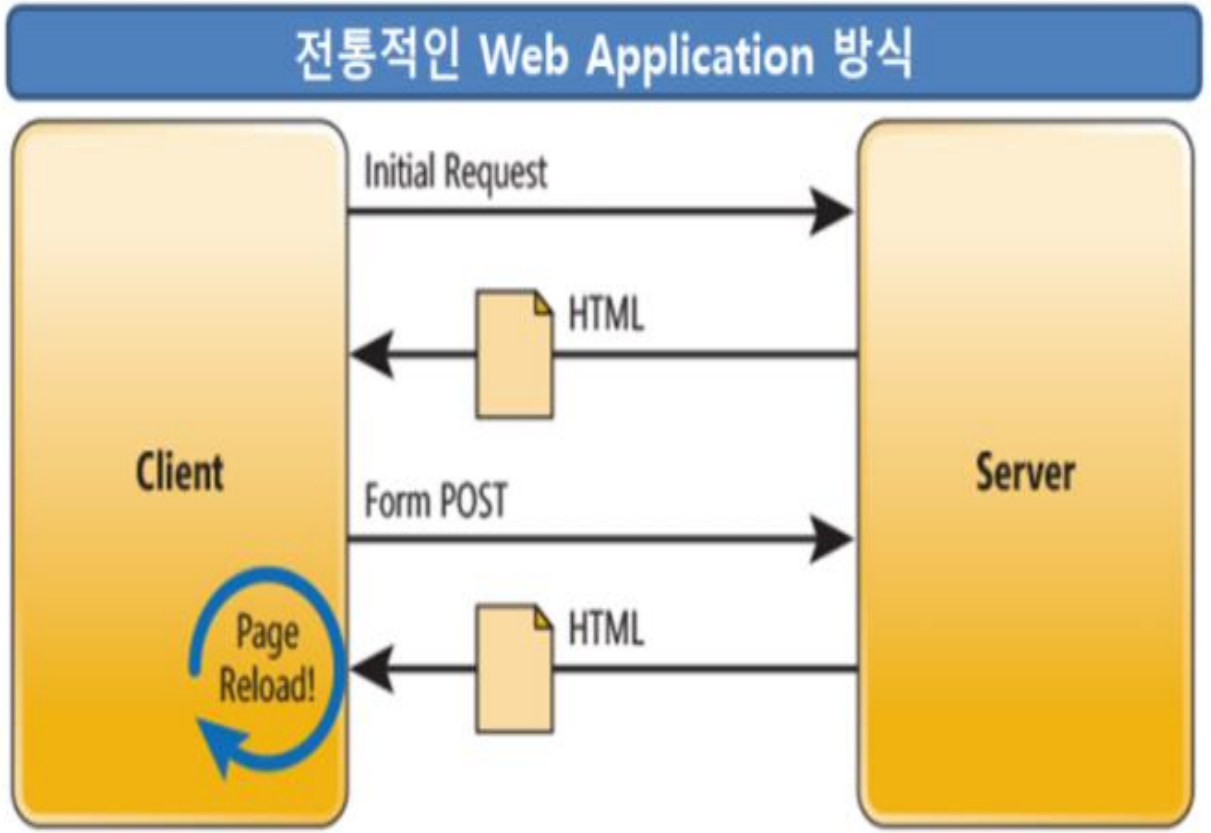
평범한 방식의 웹 페이지 렌더링

서버로부터 받은 HTML과 CSS 파일을 다운로드 받는다.

이때 다운받은 **HTML은 DOM**을 생성하고, **CSS는 CSSOM**을 생성한다.

그리고 DOM과 CSSOM을 이용하여 Render Tree를 생성한다.





클라이언트가 서버로 새로운 페이지를 요청 할 때마다 정적 리소스 및 새로운 페이지를 다시 렌더링 하므로 비효율적이다.

변경이 필요 없는 부분까지 포함하여 전체 페이지를 렌더링한다.

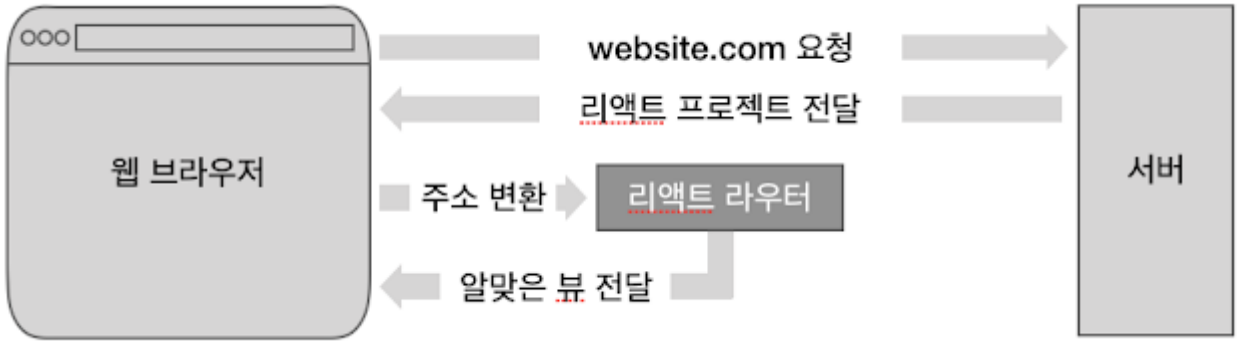
이를 보완한 SPA (Single Page Application)

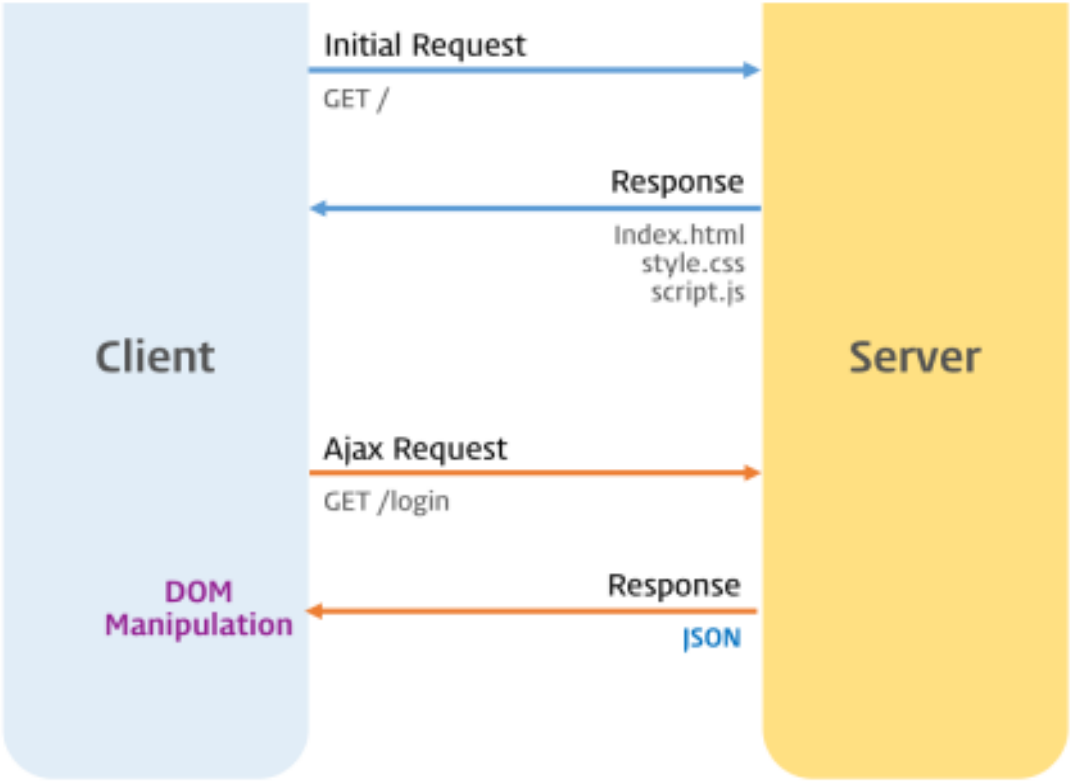
가장 전통적인 애플리케이션 방식은 페이지를 요청 할 때마다 정적 리소스 및 새로운 페이지를 다시 렌더링 하므로 비효율적이다.

웹에 필요한 모든 정적 리소스들을 최초에 한번 다운로드 한다.

사용자의 요청 및 URL에 따라서 동적으로 UI를 교체해준다.

페이지가 갱신 되었을 때, 새로운 데이터만 받아서 렌더링이 하기 때문에 처리과정이 효율적이다.





그에 따른 SPA 의 단점

모든 정적 리소스를 한번에 다운로드 하기 때문에, 프로젝트의 규모가 커지면 초기 로딩속도가 느려질 수 있다.

이를 보완한 점이 코드 스플리팅! (나중에 발표하게 된다면 설명!)

참고: https://ko.reactjs.org/docs/code-splitting.html

React-router-dom의 키워드

BrowserRouter Switch Route Redirect 등등…

BrowserRouter

- HTML5의 **history API**를 사용하여 UI를 업데이트 한다.
- HashRouter: url에 포함되어 있는 '#' 이상의 정보를 가져와서 렌더링 해주는 방식

Switch

- 감싸준 Route 목록 중, url path와 일치하는 첫번째 컴포넌트 만을 렌더링한다.

Route

- Path에 따라서 보여줄 컴포넌트를 정의해준다.
- Exact props를 활성화 해주면 정확히 해당 path에서만 렌더링이 이루어진다.

Redirect

- 일치하는 Route 목록이 없을 때, 어느 페이지로 Redirect 시킬지 정한다.

React-router-dom의 키워드
Link NavLink

Link
- 클릭시 SPA 내의 원하는 경로로 이동하는 컴포넌트.
- <a> 태그는 외부 링크로 이동할 때 쓰이는 태그이기에 새로 고침이 발생하지만, Link는 그렇지 않다.

NavLink
- Link와 기능은 동일하다.
- 만약 설정했던 URL이 활성화 되면 특정 스타일을 지정해줄 수 있다. (activeStyle Props 제공)

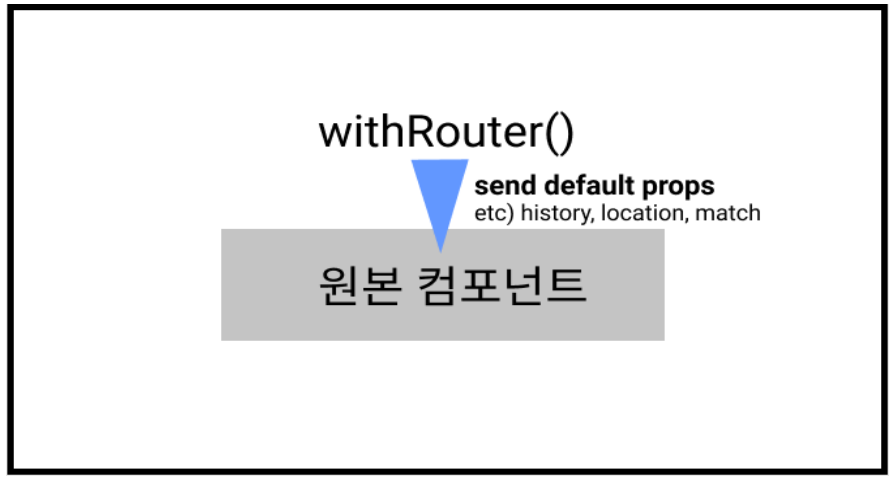
```
const Test = (): JSX.Element => {
  const activeStyle: CSSProperties = useMemo(() => {
    return {
      color: 'blue',
      border: '1px solid blue',
    };
  }, []);

  return (
    <div>
      <NavLink exact to='/' activeStyle={activeStyle}>홈 화면</NavLink>
      <NavLink exact to='/write' activeStyle={activeStyle}>글 쓰기</NavLink>
      <NavLink exact to='/member' activeStyle={activeStyle}>멤버 목록</NavLink>
      <NavLink exact to='/logout' activeStyle={activeStyle}>로그아웃</NavLink>
    </div>
  );
};
```

React-router-dom의 유용한 hooks 모음

useHistory useLocation useParams …

공통적인 특징

- Props history, match, location을 사용할 때 해야 하는 withRouter를 이용한 hoc 과정을 거치지 않아도 된다.
- withRouter hoc는 라우터에 의해서 호출된 컴포넌트가 아니라도, 해당 객체들에 접근하기 위해서 필요하다.



useHistory

- 일반적으로 Props에서 가져올 수 있는 history 객체를 hooks를 이용해서 가져 올 수 있다.
- SPA 내의 링크로 이동하려고 할 때 사용한다.
- Props history를 대체하여 사용 가능하다.

useLocation

- 현재 URL 이름 및 Query String의 정보를 가져올 때 유용하다.
- Props location을 대체하여 사용 가능하다.

useParams

- 현재 URL의 Path Parameter의 정보를 가져올 때 유용하다.
- 이를 사용하기 위해선 Route에 동적 URL이 필요하다.
- Props match를 대체하여 사용 가능하다.

```
<Route
  exact
  path='/question-form/:idx'
  component={QuestionFormPage}
/>
```

```
const { idx } = useParams<IPageParam>();
return Number(idx);
```