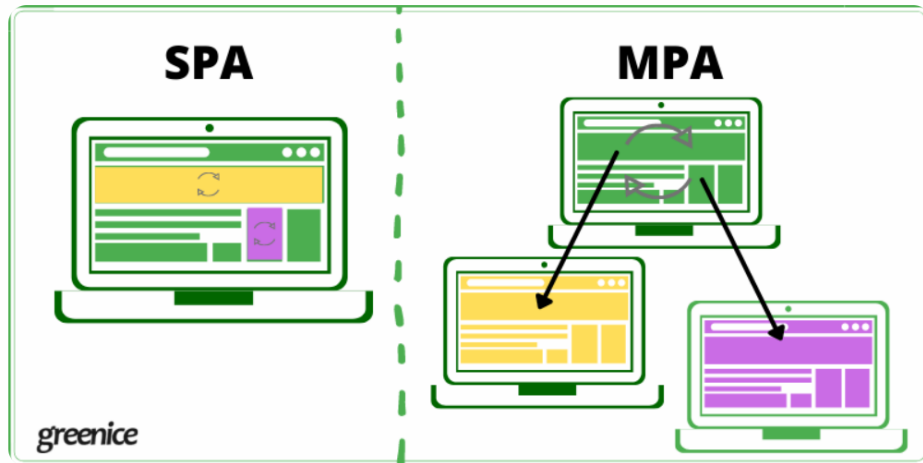


SPA VS MPA



MPA(Multi Page Application)은 여러 개의 Page로 구성된 Application이다.

새로운 페이지를 요청할 때마다 서버에서 렌더링된 정적 리소스(HTML, CSS, JavaScript)가 다운로드 된다.

페이지 이동하거나 새로고침하면 전체 페이지를 다시 reloadable된다.

▶ MPA는 SSR(Server Side Application) 방식으로 렌더링한다.

SPA(Single Page Application)는 한 개(Single)의 Page로 구성된 Application이다.

웹 애플리케이션에 필요한 모든 정적 리소스를 최초 한 번에 다운로드한다.

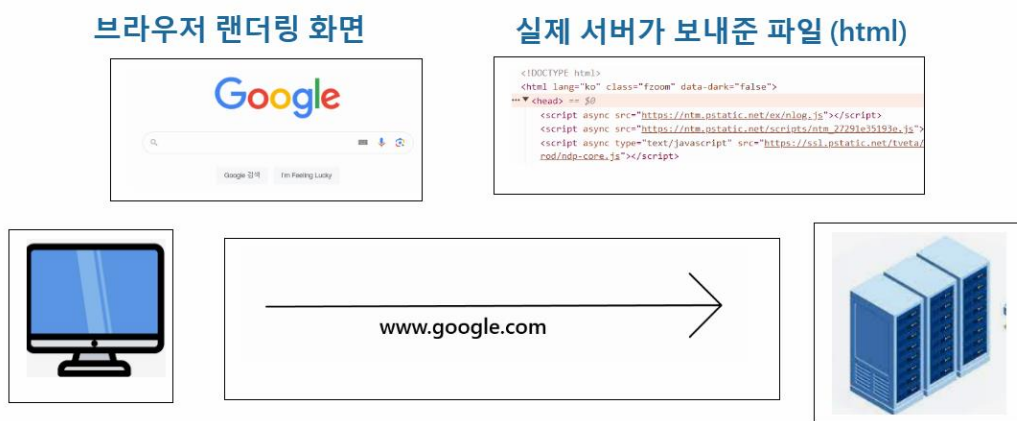
새로운 페이지 요청이 있을 때, 페이지 갱신에 필요한 데이터만 전달 받아서 변경이 필요한 부분만 갱신한다. 전체 페이지 reloadable 안된다.

▶SPA는 CSR(Client Side Rendering) 방식으로 렌더링한다.

SSR VS CSR

Server, Client 어느 사이드에서 정적 콘텐츠가 만들어지는가에 따라 나뉜다.

SSR



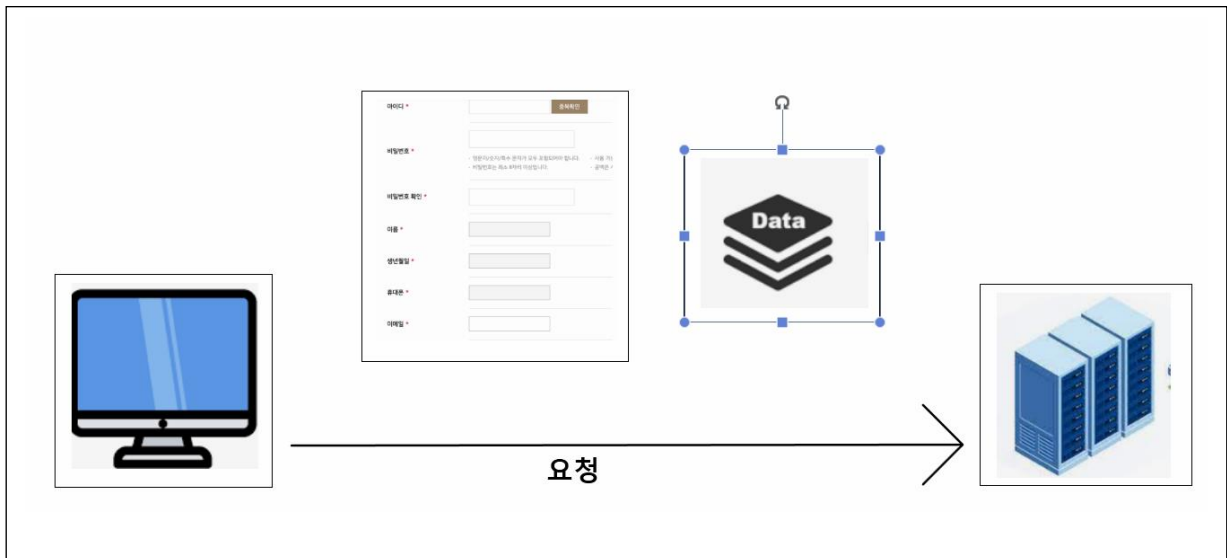
서버가 보내준 html파일을 브라우저가 분석을 해서 화면을 보여준다.

서버가 클라이언트에 모든 Content가 다 들어있는 파일을 보내준것이다.

이것이 바로 서버에서 렌더링이 진행되었다고 하며 바로 Server Side Rendering이다.

일반적으로 JSP, Django기술이 바로 SSR기술이다.

JQuery



CSR 기술이 보편화된 계기는 jQuery Ajax에서 찾아볼 수 있다.

요즘은 SPA으로 빨리 넘어가는 경향이 있어서 jQuery를 많이 사용하지 않는 추세는 있지만 사실 jQuery기술만으로도 한때는 멋진 Front UI를 구사하던 때가 있었다.

특히 jQuery 비동기 기술인 Ajax를 사용하면 서버에서 데이터(Text, Json)만 받아와 클라이언트에서 렌더링 해줄 수 있다. 이런 방식은 화면의 부분만을 갱신할 수 있어 전체페이지를 다시 불러오지 않아도 된다.

CSR

기술은 여기서 만족하지 않고 더 나아간다.

한 부분만을 렌더링(부분적인 갱신) 하기 위해서 서버에서 데이터를 가져오는 것이 아니라 웹 사이트 정적인 문서 전체를 렌더링 하는 방법을 사용한다.

본격적으로 Single Page Application 프로젝트가 시작된다.



FaceBook이 만든 React와 Community로 만들어진 Vue가 업계에서 가장 많이 사용된다.

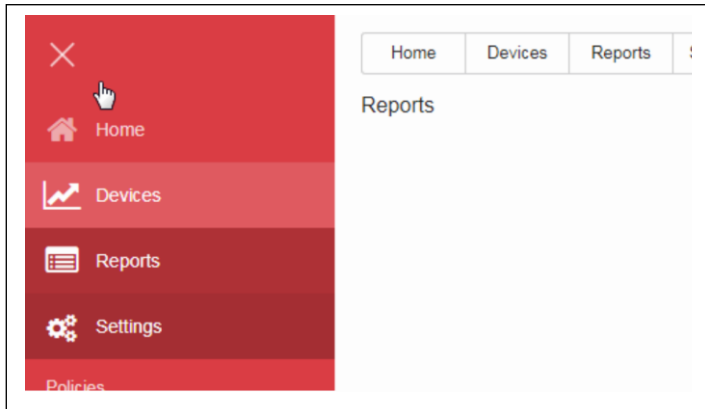
- 1) 서버가 빈 뼈대만 있는 html파일을 클라이언트에 전달.
- 2) js 파일들이 다 다운로드 되어지고 root부분에 그려질 페이지(js)가 injecting된다.
- 3) virtual dom이 생성
- 4) 초기로딩 속도가 느리고 이후 페이지 변경시에는 구동속도는 빠름

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>React App</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

서버에서 빈 html코드를 받아서 클라이언트에서 모든 렌더링이 이루어진다.

React, Angular, Vue는 다 CSR 기술을 지원한다.

CSR은 SPA를 기반으로 하는 렌더링 방식이기 때문에 페이지가 이동하거나 이벤트를 이용해서 다른 페이지의 내용을 보는 경우 전체 페이지 업로드 없이 내용이 바뀌는 부분은 새로 그려주는 방식이다.



메뉴바를 클릭할때마다 SSR 기술과 CSR이 페이지를 갱신하는 방법이 다르게 된다.

이 부분에서 CSR기술이 매우 효율적이다.

SSR은 새롭게 변경되는 부분과 변경할 필요가 없는 부분까지를 포함해서 전체 페이지를 다 만들어서 페이지를 클라이언트로 보내준다.

시간도 오래 걸리고 만들 필요가 없는 부분까지 다 만들어야 하는 번거로움이 따른다.

CSR은 변할 필요 없는 부분은 전 페이지에서 이미 렌더링이 된 상태라서 가지고 있고 새로 추가되는 부분의 내용은 비동기 통신을 이용해 데이터만 가져와 클라이언트에서 그 부분만 다시 그려준다. 속도와 효율적인 측면에서 CSR이 좋다.

CSR 단점

첫 로딩이 느리다.

root부분에 그려질 페이지가 injecting된다. 이때 렌더링할 파일들을 다 불러온다.

서버에서 올 때는 빈 페이지. 그리고 root부분에서 js파일이 연결되고 화면이 그려진다.

이때 필요한 js파일들이 다 로딩되어진다.

SSR은 요청하는 페이지만 서버에서 전달받으면 된다 (index.html)

여기서 다음 요청을 하면 다시 서버에 그 페이지를 전달받는 식이다.(login.html)

하지만 CSR은 index.html , login.html 페이지를 화면에 띄워야 한다면 여기에 필요한 js코드가 첫 로딩할때부터 다 있어야 한다. 처음에 js파일을 로딩하는 시간이 꽤 걸린다.

SEO 최적화가 잘 안된다.

검색어를 입력하면 우리 사이트가 엔진에서 검색대상이 되어야 하는데, 이런 부분에서 부족하다.

CSR기술은 처음 요청했을 때 index.html 페이지에 아무 내용이 뜨질 않는다. blank page가 나온다.

검색엔진이 이런 페이지를 보면 아무것도 없는 사이트라 생각하고 바로 버려버린다.

우리가 입력한 검색 키워드가 서치엔진에서 걸리기 힘든 구조이다.

Goole엔진은 js읽어들이는 기술을 보완해서 검색을 가능하게 해줬다라고 하는데 아직까지 완벽하진 않다.

해결책, 보완개선책

Next.js

SSR을 React로 할 수 있다.

리액트를 사용해서 서버에서 모든 것이 다 렌더링이 된 HTML 파일을 클라이언트로 전송할 수 있다.

첫 요청을 하면 SSG(Static Side Generation)으로 첫 화면(html)을 보여준다. 그 다음부터는 html파일이 CSR과 동일하게 작업을 이어나가는 방식이다.

첫 요청일때만 SSR을 하고 첫 페이지를 응답 받은 다음에 링크로 다른 요청을 했을 때 그때부터는 CSR방식으로 동작하게 된다. 그리고 이런 방식을 자동으로 다 해결해준다.

Isomorphic App(아이소몰픽 앱) : SSR,CSR의 장점을 다 모아놓은 애플리케이션이다. 리액트는 원래 CSR 기반이지만 nextjs를 활용하면 리액트로 SSR 기능이 사용 가능하다.

CSR 방식이 더 적합한 경우

1. 개인정보 혹은 특수한 정보만으로 이뤄진 사이트는 검색엔진에 노출될 필요없다.
2. 사용자와의 상호작용이 많은 경우도 마찬가지.

SSR 방식이 더 적합한 경우

1. 기업 홈페이지 혹은 병원, 기타 등등..검색 엔진에서 노출이 잘 되어야 하는 곳