

CSR vs SSR

브라우저 렌더링

CSR과 SSR은 브라우저 렌더링 방식이기 때문에 이에 대해 알기 전에 **브라우저 렌더링**에 대해 알아보려 한다. 브라우저 렌더링이란 브라우저가 서버로부터 요청해 받은 내용을 브라우저 **화면(View)**에 표시해주는 작업을 말한다. 즉, 브라우저가 서버로부터 HTML, CSS, JavaScript 문서를 전달 받아 브라우저 엔진이 각 문서를 해석해 브라우저 화면에 그려주는 것이다. 예를 들어 크롬 주소창에 naver.com 이라 검색한다면 브라우저는 네이버 웹사이트에 대한 정보를 서버로부터 받아 브라우저 화면에 네이버 홈페이지를 그려주게 된다. 그리고 이러한 브라우저 렌더링은 **CSR(Client-Side Rendering)** 과 **SSR(Server-Side Rendering)** 로 나누어진다.

CSR

CSR에 대해 알기 전에 SPA 를 먼저 알아야 한다. SPA란 Single Page Application 의 약자로, **최초 한 번 페이지 전체를 로딩한 이후부터 데이터만을 변경해 사용하는 단일 페이지로 구성된 웹 애플리케이션**을 의미한다. 즉, 화면 구성에 필요한 모든 HTML을 클라이언트가 가지고 있고, 서버 측에는 필요한 데이터만 요청해 JSON으로 받기 때문에 기존 애플리케이션에 비해 화면 구성 속도가 빠르다. 첫 로딩 시에만 전체를 로딩하기 때문에 첫 로딩 시 속도만 좀 느리며, 이후 동작은 매우 빠르다는 장점이 있다. 그리고 이러한 렌더링 방식을 CSR 이라고 한다.

CSR은 클라이언트 측에서 최초에 1번 서버에서 전체 페이지를 로딩하여 보여준다. 그 이후에는 사용자의 요청이 있을 때마다, 자원을 서버에서 제공하고, 클라이언트는 이를 해석 후 렌더링 하는 방식이다. 한 번 다시 CSR 과정을 보며 장단점을 파악해보려 한다.

1. 처음 웹 서버에 요청할 때, 데이터가 없는 문서를 반환한다.
2. HTML 및 static 파일들이 로드된다. 이때, 데이터가 있다면 데이터 또한 서버에 요청하여 화면에 나타낸다.

3. 브라우저가 서버에 HTML과 static 파일을 요청 후 로드되면, 사용자와의 상호작용에 따라 JavaScript를 통해 동적으로 렌더링을 한다.
4. 필요에 따라 데이터를 서버에 요청해 받아와 렌더링 한다.

CSR 장점

기존 서버 측에서 전체 페이지를 다시 읽어 들이는 것이 아닌 사용자의 행동에 따라 필요한 부분만 다시 읽어들이기 때문에 빠른 상호작용이 가능하다.

CSR 단점

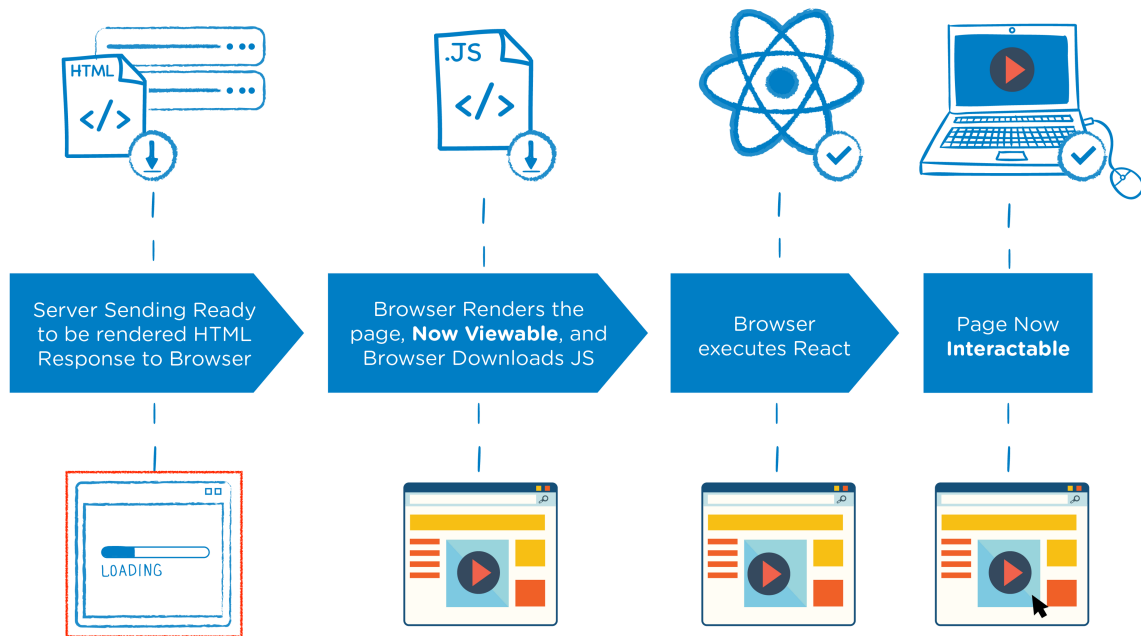
- 초기 구동 속도가 느리다.
 - CSR은 모든 자원을 다운로드 이후 브라우저에서 렌더링 하기 때문에 초기 구동 속도가 느리다.
- SEO가 어렵다.
 - CSR 방식으로 이루어진 웹 페이지에서는 화면을 생성하기 위해 반드시 JavaScript를 실행시켜야 한다. 하지만, 대부분 웹 크롤러 봇들은 JavaScript 파일을 실행시키지 못하기 때문에 HTML 에서만 콘텐츠들을 수집하게 되고 CSR 페이지를 빈 페이지로 인식하게 된다.
- 보안 문제
 - CSR은 쿠키 말고는 사용자 정보를 저장할 공간이 마땅치 않다.

SSR

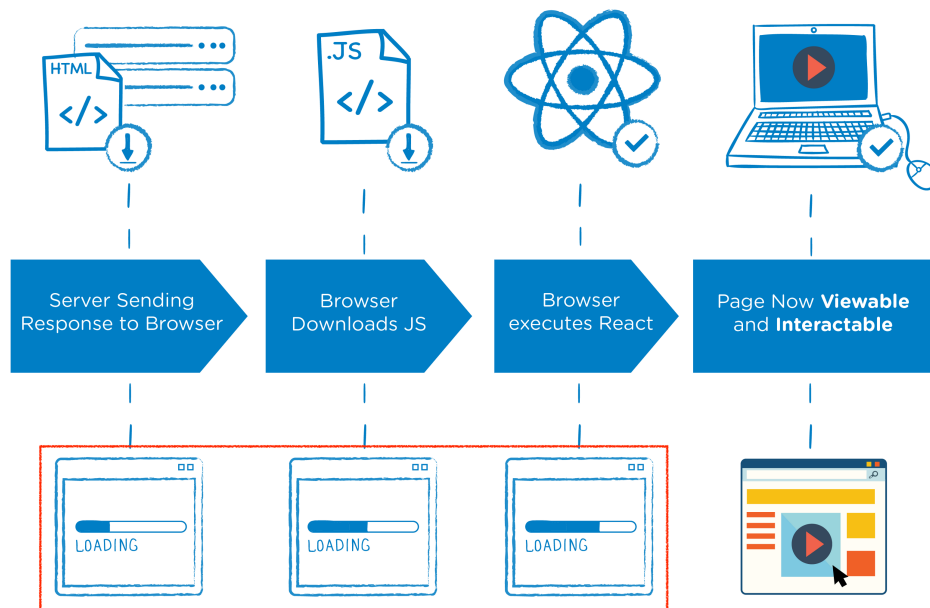
SSR은 서버에서 사용자에게 보여줄 페이지를 모두 구성하여 사용자에게 페이지를 보여주는 방식이다. SPA가 도입 되기 전 웹은 JSP/Serverlet 의 아키텍처에서 SSR 방식을 사용했다.

SSR을 사용하면 모든 데이터가 매핑된 서비스 페이지를 클라이언트에게 바로 보여줄 수 있다. 서버를 이용해 페이지를 구성하기 때문에 클라이언트에서 구성하는 CSR 보다 페이지를 구성하는 속도는 늦어지지만 **전체적으로 사용자에게 보여주는 콘텐츠 구성이 완료되는 시점은 빨라진다는** 장점이 있다. 더불어 **SEO 또한 쉽게 구성이 가능하다.**

SSR



CSR



출처 : [The Benefits of Server Side Rendering Over Client Side Rendering](#)

위 사진에서 보는 것처럼 SSR은 CSR보다 빠른 초기 로딩 속도를 보인다. 또한 위에서 말했 다시피 대부분의 웹 크롤러 봇은 HTML 기반으로 콘텐츠를 수집하고, SSR 방식은 서버에서 HTML 을 만들어 바로 화면에 뿌려주기 때문에 SEO 최적화 또한 가능하다.

다만 단점은, 초기 로딩 이후 페이지 이동 시 속도가 다소 느리다는 것이다. 페이지 이동 시 마다 클라이언트가 서버에서 필요한 데이터를 요청하고 서버가 응답해주는 방식이다 보니 속도가 다소 느리다.

SSR이 주목받는 이유

위에서 말했 다시피 SSR의 문제들을 해결하기 위해 나온게 CSR 방식이다. 그럼에도 불구하고 현재 프론트엔드 생태계는 다시 SSR 쪽으로 기울고 있다. 그 이유가 무엇일까

첫 번째는 바로 CSR의 느린 초기 로딩 속도이다. 예를 들어 페이지를 로딩하기 위해 필요한 사진 데이터가 1000장이라고 생각해보자. CSR의 경우 해당 데이터를 모두 다운로드 하기 전까지 화면을 보여주지 못한다. 그렇게 되면 사용자는 상당한 시간 동안 하얀 화면을 볼 수 밖에 없다.

두 번째는 SEO 최적화이다. 난 이게 가장 큰 문제라고 보는데 기업의 입장에서 검색 엔진에 내 서비스가 뜨지 않는다는 것은 큰 문제이다.