

SPA, CSR, SSR



SSR

Server Side Rendering의 약자로 말 그대로 서버쪽에서 렌더링 준비를 마친 상태로 클라이언트에 전달하는 방식을 의미합니다.

기본적으로 User가 웹사이트 요청을 보내면 서버는 바로 화면에서 렌더링이 가능한 html 파일을 만들고, 클라이언트는 해당 파일을 받아 화면에 즉시 렌더링을 시켜줍니다. 이 상황에서는 아직 JavaScript가 읽히기 전이기 때문에 사이트 조작은 불가능합니다.

다음으로 클라이언트가 JavaScript를 다운받고, 브라우저가 JavaScript를 실행시킵니다.

어찌됐든 JS가 다운되기 전에도 사용자들은 실제 화면을 보고 있을 수 있다는 장점이 있습니다.



CSR

Client Side Rendering의 약자로 SSR과 달리 렌더링이 클라이언트 쪽에서 일어납니다. 즉, 서버는 웹사이트 요청을 받으면 클라이언트에 html과 js를 보내주고, 클라이언트는 이를 받아 렌더링을 시작합니다.

이때, CSR의 경우 첫 페이지 로드 시에 필요한 모든 html과 css, js를 모두 불러오기 때문에 첫 페이지 로딩 시간이 길니다. 또한, 로드 되는 동안에는 사용자가 화면을 볼 수가 없습니다.

하지만 첫 로드에서 모든 리소스를 받아오기 때문에 이후 페이지 전환에서는 보다 짧은 로딩 시간을 가집니다.



차이점

1. 웹 페이지 로딩 시간

- 우선 가장 큰 차이는 웹 페이지 로딩 시간입니다. 웹 페이지 로딩은 크게 2가지로 나뉘는데 첫 번째는, 첫 페이지 로딩 시간, 두 번째는 나머지 로딩 시간입니다.
- 위에서 말했듯이 CSR의 경우 첫 페이지 로딩 시간이 길지만 나머지 로딩 시간이 짧습니다.
- 반대로 SSR의 경우 페이지 전환시마다 서버에 페이지를 요청하기 때문에 첫 페이지 로딩 시간은 짧지만 나머지 로딩 시간이 길습니다.

2. SEO 대응

- 다음으로 검색 엔진 대응입니다. 검색 엔진은 자동화 로봇인 크롤러로 웹 사이트들을 읽습니다. CSR은 자바스크립트를 실행시켜 동적으로 콘텐츠가 생성되기 때문에 JS가 실행되어야 metadata가 바뀌었습니다. 물론 최근엔 많이 바뀌고 있다고는 하지만 이전까지의 크롤러들은 JS를 실행시키지 않았기 때문에 SEO 최적화가 필수적이었습니다.

3. 서버 자원 사용

- SSR의 경우 매번 서버에 페이지를 요청하기 때문에 서버 자원을 더 많이 사용합니다.



트렌드

리액트가 나오면서 CSR은 엄청난 부흥기를 맞았습니다. 하지만 CSR의 여러 단점들은 시대가 바뀌면서 사람들이 첫 페이지 대기 시간에 민감해지고, 기업 입장에서 SEO 최적화가 되지 않는 것은 너무나 큰 단점이 되었고 원래 SSR 이후 CSR이 나왔지만 다시 SSR로 복귀하게 되었습니다.

하지만 SSR과 CSR을 동시에 사용할 수 있다면 이라는 생각이 나오게 되어 둘을 모두 사용하는게 트렌드가 되었고 리액트의 경우 Next.js 라는 리액트 프레임워크의 등장으로 인해 SSR과 CSR을 동시에 쉐길 수 있게 되었습니다.