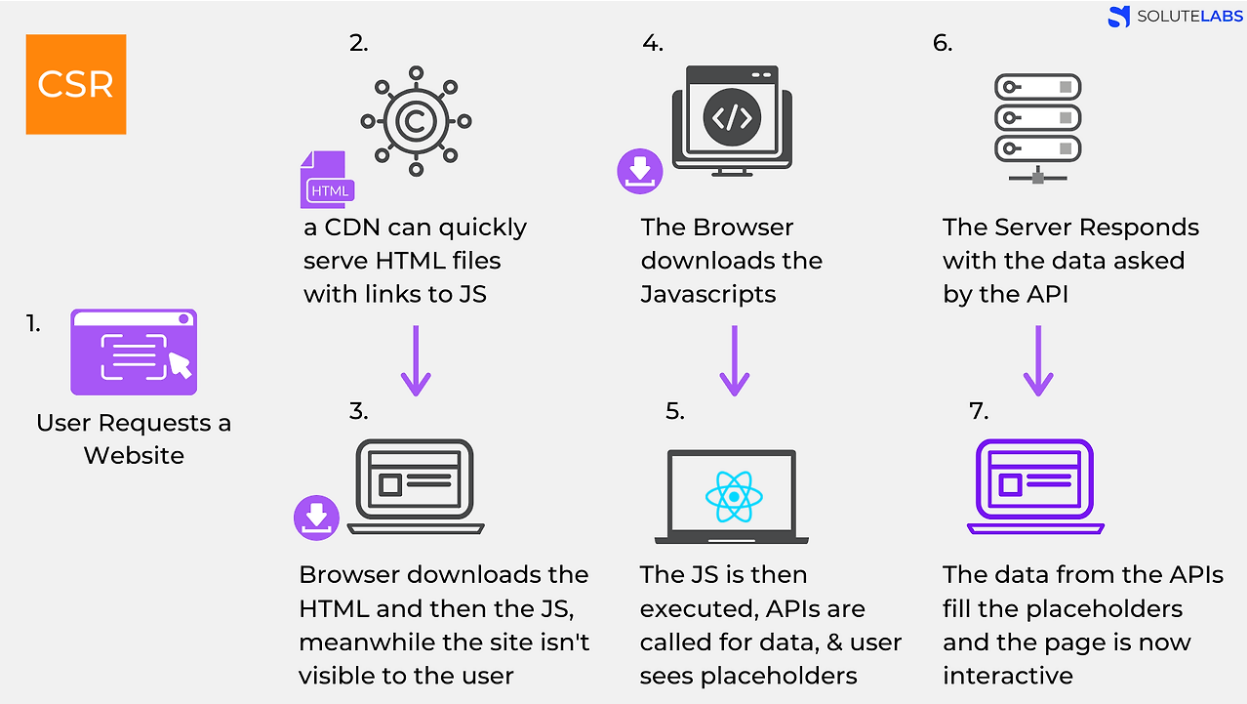
[개발환경] CSR, SSR

구현 어려움 정도	개념
구현 완료	☑
기간	@December 11, 2024
정리 완료	☑

CSR

- Client Side Rendering
- 말 그대로 SSR과 달리 렌더링이 클라이언트 쪽에서 일어남
- 즉, 서버는 요청을 받으면 클라이언트에 HTML과 JS를 보내준다.
- 클라이언트는 그것을 받아 렌더링을 시작한다.

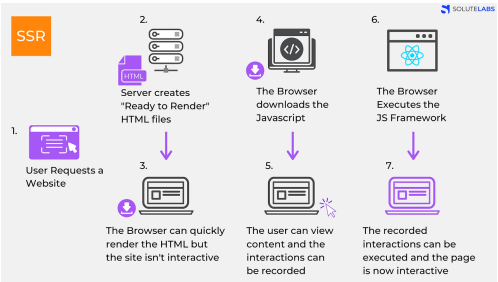


1. 유저가 웹사이트에 요청을 보냄
2. CDN이 HTML 파일과 JS로 접근할 수 있는 링크를 클라이언트로 보냄
3. 클라이언트는 HTML과 JS를 다운로드 받는다.
   이 때, SSR과는 달리 유저는 아무것도 볼 수 없음
4. 생략
5. 다운로드가 완료된 JS가 실행된다. 데이터를 위한 API가 호출됨
6. 서버가 API로 부터의 요청에 응답
7. API로 부터 받아온 data를 placeholder 자리에 넣어줌
   이제 페이지는 상호작용이 가능해짐

즉, 서버에서 처리 없이 클라이언트로 보내주기 때문에 자바스크립트가 모두 다운로드되고 실행이 끝나기 전까지는 사용자가 볼 수 없음

SSR

- Server Side Rendering
- 서버쪽에서 렌더링 준비를 끝마친 상태로 클라이언트에 전달하는 방식



1. 유저가 웹사이트에 요청을 보냄
2. 서버는 'Ready to Render'
   즉, 즉시 렌더링 가능한 HTML 파일을 만든다.
   리소스 체크, 컴파일 후 완성된 HTML 컨텐츠로 만든다.
3. 클라이언트에 전달되는 순간, 이미 렌더링 준비가 되어있기 때문에 HTML은 즉시 렌더링된다.
   그러나, 사이트 자체는 조작 불가능(Javascript가 읽히기 전이기 때문)
4. 클라이언트가 자바스크립트를 다운로드 받는다.
5. 다운로드 받아지고 있는 사이에 유저는 컨텐츠는 볼 수 있지만 사이트를 조작할 수 없다. 이 때의 사용자 조작을 기억하고 있는다.
6. 브라우저가 Javascript 프레임워크를 실행한다.
7. Javascript까지 성공적으로 컴파일 됐기 때문에 기억하고 있던 사용자 조작이 실행되고 이제 웹 페이지는 상호 작용이 가능해진다.

즉, 서버가 이미 **렌더 가능한 상태**로 클라이언트에 전달되기 때문에 Javascript가 다운로드 되는 동안 사용자는 무언가를 보고 있을 수 있다.

CSR과 SSR의 차이점

1. 웹페이지 로딩시간

웹 페이지 로딩의 종류는 두 가지로 나눌 수 있다.

웹 사이트의 가장 첫 페이지를 로딩하는 것과 나머지를 로딩하는 것

CSR

HTML과 CSS, 모든 스크립트를 한 번에 불러온다.

SSR

필요한 부분의 HTML과 스크립트를 불러오게 된다.

평균적으로 SSR이 **초기 로딩속도가 빠르다.**

나머지 로딩 시간

첫 페이지를 로딩한 후, 사이트의 다른 곳으로 이동하는 식의 동작을 가정해보면 CSR은 이미 첫 페이지 로딩할 때 나머지 부분을 구성하는 코드를 받아왔기 때문에 다른 페이지(나머지) 로딩 시간이 빠르다.

반면 SSR은 첫 페이지를 로딩한 과정을 정확히 다시 실행하기 때문에 느리다.

2. SEO 대응

검색 엔진은 자동화된 로봇인 '크롤러'로 웹 사이트를 읽는다.

CSR은 자바스크립트를 실행시켜 동적으로 컨텐츠가 생성되기 때문에 자바스크립트가 실행되어야 metadata가 바뀌었다.

(이전 크롤러들은 Javascript를 실행시키지 않았기 때문에 SEO 최적화가 필수적이였다.)

SSR은 애초에 서버 사이트에서 컴파일되어 클라이언트로 넘어오기 때문에 크롤러에 대응하기 용이하다.

3. 서버 자원 사용

SSR이 서버 자원을 더 많이 사용한다. 매번 서버에 요청을 하기 때문이다.

반면 CSR은 클라이언트에 일감을 몰아주기 때문에 서버에 부하가 적다.

참고

https://hahahoho5915.tistory.com/52