

쿠키 vs 세션 vs 웹스토리지(로컬, 세션)

HTTP 프로토콜의 한계를 보완하기 위해 **쿠키** 또는 **세션**을 사용한다. 기본적으로 HTTP 프로토콜 환경은 **비연결지향, 무상태**한 특성을 가지기 때문에 요청만으로 서버는 클라이언트를 구별할 수 없다.



비연결지향

HTTP 는 먼저 클라이언트가 요청을 서버에 보내면, 서버는 클라이언트에게 요청에 맞는 응답을 보내고 TCP/IP 연결을 끊는 특성이다.



무상태

연결을 끊는 순간 클라이언트와 서버의 통신이 끝나며 상태 정보를 유지하지 않는 특성이다.

쿠키

- 클라이언트에 저장되는 키와 같이 들어있는 작은 파일이다.
- 클라이언트의 상태 정보를 **로컬에 저장했다가** 참조한다.
- 클라이언트에 **300개**까지 쿠키 저장이 가능하며, **하나의 도메인당 20개의 값**만 가질 수 있고, 하나의 **쿠키값은 4KB** 까지 저장이 가능하다.
- Response Header에 **Set-Cookie** 속성을 사용하면 클라이언트에 쿠키를 만들 수 있다.

- 쿠키는 사용자가 따로 요청하지 않아도 브라우저가 Request 시에 Request Header를 넣어 **자동으로 서버에 전송**한다.

세션

- 사용자 정보를 파일 브라우저에 저장하는 쿠키와 달리 세션은 **서버 측에서 관리**한다.
- 서버에서 클라이언트를 구분하기 위해 세션 ID를 부여하며, 웹 브라우저가 서버에 접속해 **브라우저를 종료할 때까지 인증 상태를 유지**한다.
- 접속 시간에 제한을 두어 일정 시간 응답이 없다면 정보가 유지되지 않게 설정이 가능하다.
- 데이터를 서버에 두기 때문에 **쿠키보다 보안이 좋지만, 사용자가 많아질수록 서버 메모리를 많이 차지**하게 된다.

쿠키 vs 세션

- 데이터 저장 위치 : 쿠키는 클라이언트, 세션은 서버
- 보안 : 저장위치 때문에 쿠키는 스니핑에 당할 우려가 있지만, 세션은 쿠키를 이용해 세션 아이디만 저장하고 서버에서 처리하기 때문에 보안성은 세션이 더 좋다
- 라이프 사이클 : 쿠키는 브라우저를 종료해도 만료기간이 남아있으면 존재, 세션은 브라우저 종료 시 만료기간에 상관없이 종료
- 속도는 쿠키가 세션보다 빠르다

웹 스토리지

- 클라이언트에 데이터를 저장할 수 있도록 **HTML5** 부터 추가된 저장소

- 간단한 **key-value** 스토리지 형태
- 쿠키와 달리 자동 전송의 위험성이 없다
- 오리진(Origin)(도메인, 프로토콜, 포트) 단위로 접근이 제한되는 특성 덕분에 **CSRF**로부터 안전하다.
- 쿠키보다 **큰 저장 용량을 지원**한다. (모바일 2.5MB, 데스크탑 5~10MB)
- 서버가 HTTP 헤더를 통해 스토리지 객체를 조작할 수 없다. (웹 스토리지 객체 조작은 JavaScript 내에서만 수행이 가능하다.)
- 오직 **문자형 데이터 타입만 지원**한다.
- **로컬 스토리지**와 **세션 스토리지**가 있으며, 같은 Storage 객체를 상속하기 때문에 메서드가 동일하다.

로컬 스토리지

- 사용자가 데이터를 지우지 않는 이상, 브라우저나 OS를 종료해도 계속 브라우저에 남아 있다. (영구성)
- 단, 동일한 브라우저를 사용할 때만 해당
- 지속적으로 필요한 데이터 저장 (자동 로그인 등)

세션 스토리지

- 데이터가 오리진 뿐만 아니라 브라우저 탭에도 종속되기 때문에, 윈도우나 브라우저 탭을 닫을 경우 제거
- 일시적으로 필요한 데이터 저장(일회성 로그인 정보, 입력폼 저장 등)