Cookie vs Session

Cache : 뭐야 나도 껴줘요

0. HTTP의 특징

(1) Stateless 프로토콜

- 클라이언트의 상태 정보를 가지지 않는 서버 처리 방식으로 동작
- 클라이언트와 첫번째 통신에서 데이터를 주고 받아도, 두번째 통신에서 이전 데이터를 유지 X

(2) Connectionless 프로토콜

- 클라이언트가 서버에 요청(Request)을 했을 때,그 요청에 맞는 응답(Response)을 보낸 후 연결을 끊는 처리 방식으로 동작
- HTTP 1.1 버전에서 연결을 유지하고, 재활용 하는 기능이 Default 로 추가되긴함

But, 실제로는 데이터 유지가 필요한 경우가 많다.

정보가 유지되지 않는다면?

- 매번 다시 로그인
- 상품을 선택했는데 구매 페이지로 넘어가니까 상품이 없음
- 매번 새로운 사이트를 들어온 듯한 기분 > new!

따라서, Stateful 경우를 대처하기 위해서 쿠키와 세션을 사용

쿠키와 세션의 가장 큰 차이점은 **정보의 저장 위치**

- 쿠키는 **클라이언트(= 사용자브라우저)** 에 저장
- 세션은 **서버** 에 저장

쿠키와 세션을 간단히 말하자면

HTTP 프로토콜을 사용하는 인터넷 사용자가 어떤 웹사이트를 방문할 경우,
사용자와 서버 사이의 연결을 확인하기 위한 정보

1. 쿠키 Cookie

서버가 클라이언트의 요청을 식별하는데 사용

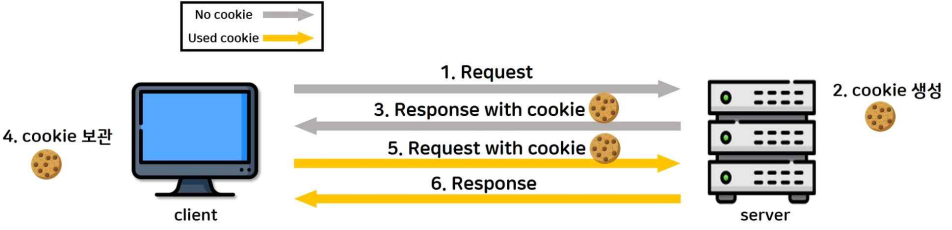


그림 출처 : https://raonctf.com/essential/study/web/cookie_connection

- **사용자의 브라우저에 저장**되고, 통신할 때 HTTP 헤더에 포함되는 텍스트 데이터 파일

- 이름, 값 만료기간(지정 가능), 경로 정보가 있고 키와 값으로 구성

- 사용자를 구분하는데 매우 유용

- 해당 사용자의 컴퓨터를 사용한다면 누구나 쿠키에 입력된 값을 쉽게 확인 가능

→ 보안성 낮기 때문에 민감한 정보들을 저장하는데 사용하지 않음

어디에 사용될까?

1. 세션 ID 관리
2. 사용자 선호 및 테마
3. 사용자 행동 기록 및 분석
4. 자동 로그인 유지
5. 위시 리스트 저장
6. 팝업 보지 않기 등등

쿠키 제약 조건

- 클라이언트는 총 300개의 쿠키 저장 가능
- 하나의 도메인 당 20개의 쿠키 가질 수 있음

→ 20개가 넘으면 가장 적게 사용되는 것부터 삭제

- 하나의 쿠키는 4KB (4096byte) 저장 가능

2. 세션 Session

브라우저가 서버에 연결되어 있는 동안 유지하는 데이터 집합

(서버에 저장되는 쿠키)

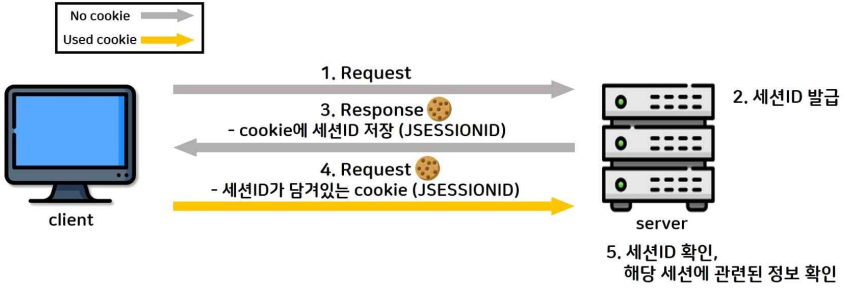


그림 출처 : https://raonctf.com/essential/study/web/session_connection

- **서버 내부에 저장**되며, 저장된 값은 반영구적

- 사용자가 특정 시간동안 사용되지 않을 경우 폐기될 수 있는 정보

- 사용자 로컬이 아닌 서버에 직접 저장되므로, 세션 내의 데이터를 탈취하는 것이 어려움

→ 보안성이 비교적 높아 중요한 데이터를 저장 시 사용

✓ 민감한 정보 관리, 사용자의 비밀번호 및 개인정보

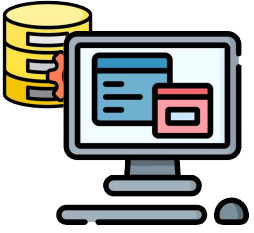
3. 쿠키와 세션의 차이점 간단 비교!

	쿠키	세션
저장위치	클라이언트 (웹 브라우저)	서버 (서버 메모리/데이터베이스)
보안	누구든 고칠 수 있기 때문에 보안 취약	서버에 저장되어 비교적 보안성이 좋음
라이프 사이클	만료 기간을 지정, 브라우저 종료 시에도 유지	브라우저 종료 시 삭제 (기간 지정이 가능하긴 함)
속도	빠르다	느리다
저장	문자열만 저장	문자열 뿐만 아니라 객체까지 저장 가능

4. 캐시 Cache

리소스 파일들의 임시 저장소

컴퓨터 과학에서 캐시는 데이터나 값을 미리 복사해놓는 임시 장소를 가리킴



같은 웹 페이지에 접속할 때 사용자의 PC에서 로드 (서버를 거치지 않아도 됨)

이전에 사용되었던 데이터는 다시 사용될 가능성이 높다

→ 다시 사용될 확률이 있는 데이터들을 빠르게 접근 가능한 저장소에 저장

→ 페이지 로딩 속도 개선

이미지, 비디오, 오디오, CSS, JS 등

캐시의 종류

[Local Cache]

- Local 장비 내에서만 사용되는 캐시로, Local 장비의 Resource를 이용한다.
- Local에서만 작동하기 때문에 속도가 빠르다.
- Local에서만 작동하기 때문에 다른 서버와 데이터 공유가 어렵다.

[Global Cache]

- 여러 서버에서 Cache Server에 접근하여 사용하는 캐시로 분산된 서버에서 데이터를 저장하고 조회할 수 있다.
- 네트워크를 통해 데이터를 가져오므로, Local Cache에 비해 상대적으로 느리다.
- 별도의 Cache서버를 이용하기 때문에 서버 간의 데이터 공유가 쉽다.

놀랍게도 끝~

감사합니다