NETWORK

쿠키 & 세션 알아보기

HTTP에서 상태를 관리하는 법

고종환

목차

1.쿠키

쿠키

: 웹 서버가 확인할 목적으로 브라우저에 저장하는 작은 텍스트 파일

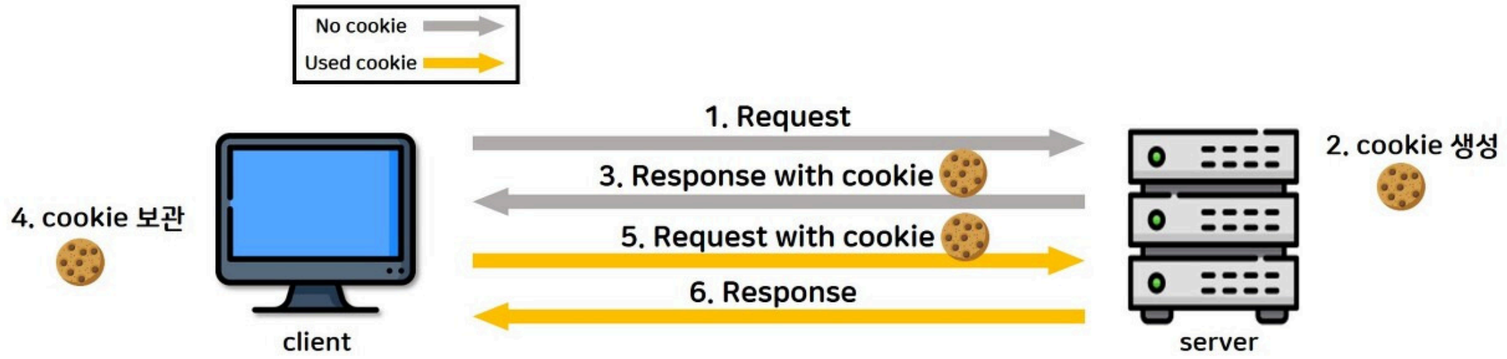
- Stateless한 HTTP 통신에서 상태를 저장하기 위한 기술
- 상태 정보를 클라이언트에 저장하는 형태
- Key-Value 형태의 값을 갖는다.
- 4KB의 용량 제한
- 사용자가 스스로 변경 가능

쿠키의 종류

- 세션 쿠키 vs. 영구 쿠키
  - 세션 쿠키: 브라우저가 지속되는 동안 유지
  - 영구 쿠키: 정해진 기간 또는 영구적으로 유지되는 쿠키

- 퍼스트 파티 쿠키 vs. 서드 파티 쿠키
  - 퍼스트 파티 쿠키: 같은 도메인에서 생성된 쿠키
  - 서드 파티 쿠키: 다른 도메인에서 생성된 쿠키

쿠키 동작 방식



클라이언트가 목적에 따라 스스로 쿠키를 만들 수도 있다. (테마 설정)

쿠키 사용방법

서버(Set-Cookie)

```
Set-Cookie: <cookie-name>=<cookie-value>
```

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: theme=dark
Set-Cookie: lang=ko

[page content]
```

브라우저(Cookie)

```
GET /sample_page.html HTTP/1.1
Host: www.example.org
Cookie: theme=dark; lang=ko
```

쿠키를 쓰는 3가지 목적

세션 관리

서버에 저장해야할
로그인, 장바구니, 게임 스코어 등의 정보 관리

(예시 : 로그인 유지, 장바구니에 담은 제품 유지 등)

개인화

사용자 선호, 테마 등
사용자의 개인 세팅을
저장, 관리

(예시 : 다크모드 사용, 언어 설정, 메뉴 순서 최적화)

트래킹

분석 및 광고 개제를 위해
웹사이트 내
사용자 행동 기록, 관리

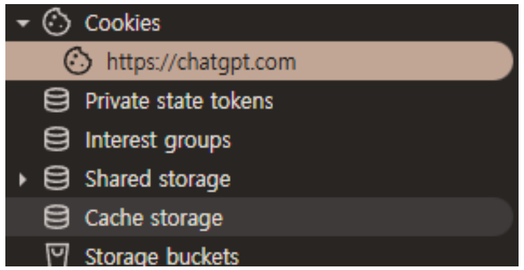
(예시 : 분석데이터 수집, 리타게팅 광고에 기여)

(참고) 서드 파티 쿠키

: 접속한 웹 페이지의 소유자가 아닌 다른 도메인에서 사용하는 쿠키

- <iframe>, <script> 등의 외부 리소스 요청을 통해 다른 도메인의 쿠키가 활용된다.
- 주로 마케팅 및 광고 목적으로 사용되며, 사용자의 행동이 쿠키에 의해 추적된다.

- 크롬에서 개인정보 보호를 이유로 서드파티 쿠키 지원을 중단하려고 했으나... 광고업계의 반발로 결국 24년 7월경에 이 계획을 철회했다.



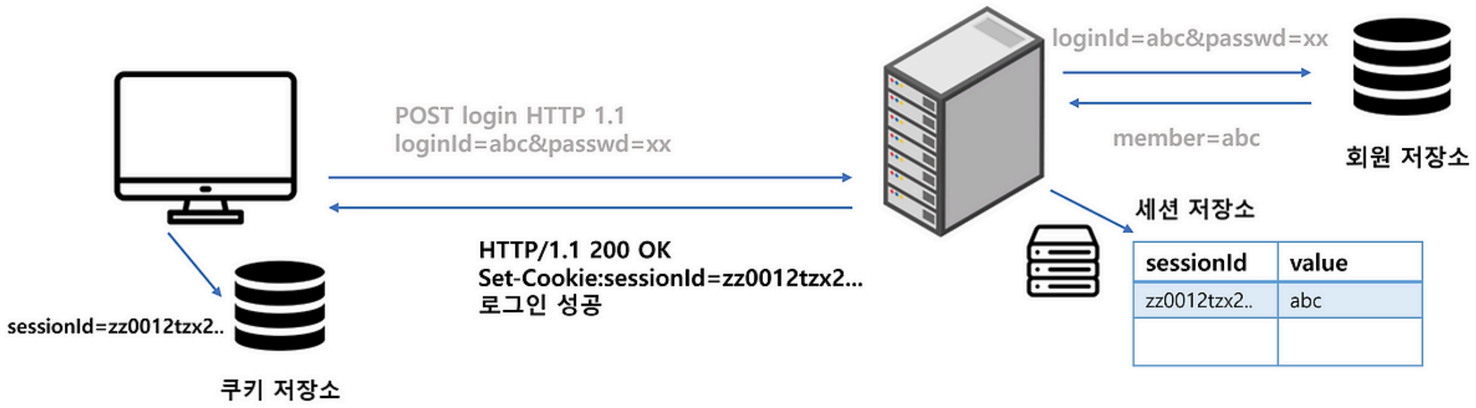
https://velog.io/@teammint/%EC%84%9C%EB%93%9C%ED%8C%8C%ED%8B%B0-%EC%BF%A0%ED%82%A4%EB%A5%BC-%EC%95%8C%EC%95%84%EB%B3%B4%EC%9E%90-1

2. 세션

세션

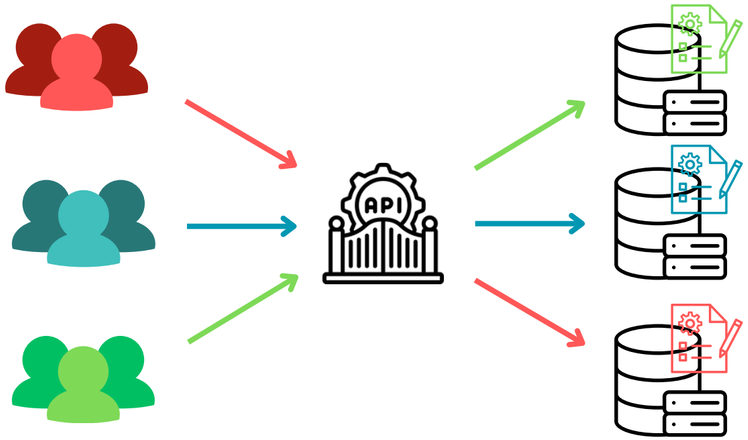
: 서버에서 세션 객체를 만들고 클라이언트와 세션 ID를 주고받으며 상태를 유지하는 방식

- 쿠키 그 자체에 민감한 정보(ID, PW)를 담는 것을 보완
- 익명의 사용자에게 장바구니 등의 상태 저장 가능
- 상태 정보를 서버에 저장

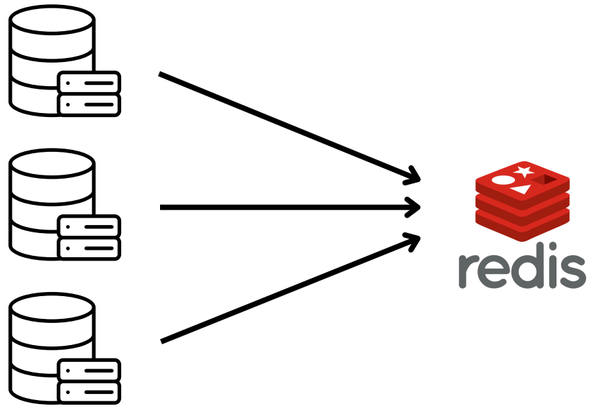


분산 서버에서 세션을 어떻게 관리하는가?

Session Replication

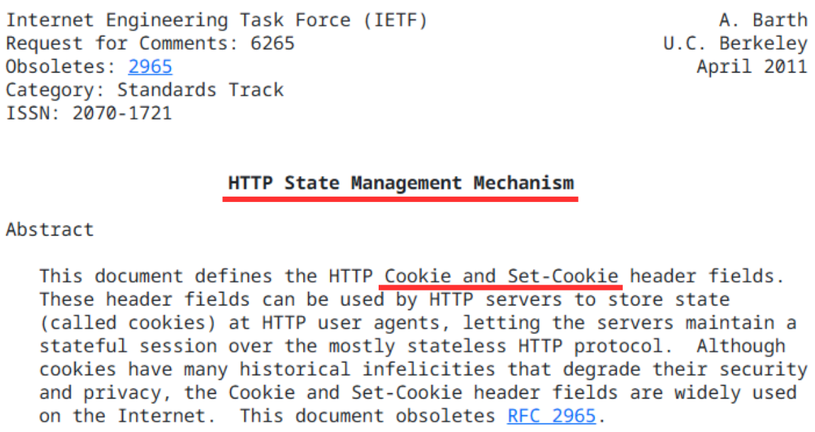


Sticky Session



Centralized Session

주의: 세션은 HTTP 프로토콜의 일부가 아닙니다



Internet Engineering Task Force (IETF)
Request for Comments: 6265
Obsoletes: 2965
Category: Standards Track
ISSN: 2070-1721

A. Barth
U.C. Berkeley
April 2011

HTTP State Management Mechanism

Abstract

This document defines the HTTP Cookie and Set-Cookie header fields. These header fields can be used by HTTP servers to store state (called cookies) at HTTP user agents, letting the servers maintain a stateful session over the mostly stateless HTTP protocol. Although cookies have many historical infelicities that degrade their security and privacy, the Cookie and Set-Cookie header fields are widely used on the Internet. This document obsoletes RFC 2965.

- **쿠키**: HTTP의 **무상태성**을 보완하기 위해 **HTTP 프로토콜 명세에 공식적으로 포함된 기능**입니다. HTTP 헤더에 **Cookie**와 **Set-Cookie** 필드를 두는 방식이 표준화되어 있습니다.
- **세션**: 웹 개발자들이 **쿠키의 보안 취약성**을 보완하고자 **서버 측에서 구현한 로직**입니다. 세션은 HTTP 프로토콜 자체가 아닌, 그 위에 얹어진 애플리케이션 계층의 개념입니다.

즉, 쿠키는 **HTTP의 상태 저장을 위한 표준**이며,
세션은 **서버에서 상태를 관리하는 방식**입니다.

3. 무상태성

HTTP의 무상태성(Stateless)

: 서버에서 클라이언트의 상태를 저장하지 않는 성질

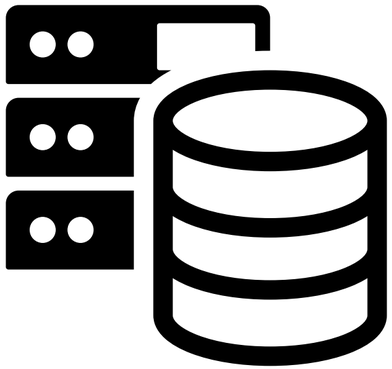
HTTP is a stateless application-level protocol...(RFC 7231)

Stateless의 특징

- **독립적인 요청**: 클라이언트로부터 들어오는 모든 요청을 완전히 독립적으로 간주합니다. 각 요청은 그 자체로 완결된 하나의 작업단위이며, 이전 요청이나 다음 요청과 연결되지 않습니다.

Statelesss의 강점

서버 부하 감소



서버 확장성

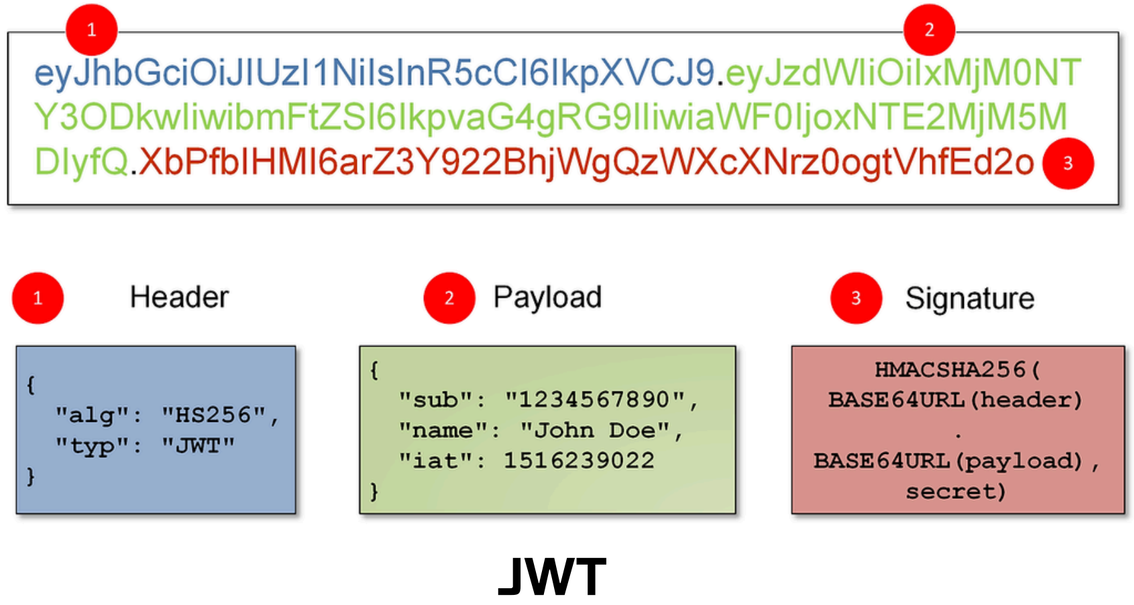
가용성

시스템의 규모가 커질수록 Stateless의 중요성은 커진다!

서버가 Stateless하다면
매번 로그인 없이 사용자 인증을 유지할 수 있을까?

무상태 토큰 기반 인증

: 서버에서 발급한 토큰으로 세션 없이 사용자를 식별



- 로그인 성공시, 서버가 토큰 발급
- 서명을 통해 토큰이 유효한지 확인하여 사용자를 식별
- ex) JWT, PASETO, SAML

- 특징
  - 세션ID와 다르게 그 자체로 사용자 식별 가능(self-contained)
  - 높은 이식성
  - 유효기간을 짧게 설정하여 보안 강화

그럼에도 불구하고...

완전한 Stateless 서버는
로그아웃, 토큰 무효화 등의 기능이 불가능

-> 현실적으로 완전한 Stateless 대신 Stateful을 혼합한 구조가 많다.

정리: 인증 방식 비교

	Only 쿠키	세션	무상태 토큰
저장위치	브라우저	서버	브라우저/앱
서버의 상태성	stateful	stateful	stateless
확장성	높음	**낮음**	매우 높음
보안(인증)	**민감 정보 유출 가능**	비교적 안전	**탈취 시, 무효화 불가**
사용 사례	사용 X	전통적인 웹서비스	RESTful API, MSA, 모바일앱

감사합니다