**쿠키와 세션 개념**

**HTTP의 특징과 쿠키와 세션을 사용하는 이유**

* HTTP 프로토콜의 특징이자 약점을 보완하기 위해서 사용함.
* HTTP 프로토콜 환경에서는 서버는 클라이언트가 누구인지 확인해야함. 그 이유는 HTTP 프로토콜이 connectionless, stateless한 특성이 있기 때문이다.

|  |
| --- |
| **Connectionless** |
| 클라이언트가 요청을 한 후 응답을 받으면 그 연결을 끊어 버리는 특징. HTTP는 먼저 클라이언트가 request를 서버에 보내면, 서버는 클라이언트에게 요청에 맞는 response를 보내고 접속을 끊는 특징이 있다. |
| **Stateless** |
| 통신이 끝나면 상태를 유지하지 않는 특징. 연결을 끊는 순간 클라이언트와 서버의 통신이 끝나며 상태 정보는 유지하지 않는 특성이 있다. 쿠키와 세션은 위의 두가지 특징을 해결하기 사용한다. |

정리하자면, 쿠키와 세션을 사용했을 경우, 한번 로그인을 하면 어떠한 방식에 의해서 그 사용자에 대한 인증을 유지하게 되는 것이다.

**쿠키(Cookie)**

쿠키란?

* 쿠키는 클라이언트(브라우저)로컬에 저장되는 키와 값이 들어 있는 작은 데이터 파일이다.
* 사용자 인증이 유효한 시간을 명시할 수 있으며, 유효 시간이 정해지면 브라우저가 종료되어도 인증이 유지된다는 특징이 있다.
* 쿠키는 클라이언트의 상태 정보를 로컬에 저장했다가 참조한다.
* 클라이언트에 300개까지 쿠키 저장 가능, 하나의 도메인당 20개의 값만 가질 수 있다. 하나의 쿠키값은 4KB까지 저장한다.

쿠키의 구성요소

* 이름: 각각의 쿠키를 구별하는 데 사용되는 이름
* 값: 쿠키의 이름과 관련된 값
* 유효시간: 쿠키의 유지시간
* 도메인: 쿠키를 전송할 도메인
* 경로: 쿠키를 전송할 요청 경로

쿠키의 동작 방식

1. 클라이언트가 페이지를 요청
2. 서버에서 쿠키를 생성
3. HTTP 헤더에 쿠키를 포함시켜 응답한다.
4. 브라우저가 종료되어도 쿠키 만료 기간이 있다면 클라이언트에서 보관하고 있음
5. 같은 요청을 할 경우 HTTP 헤더에 쿠키를 함께 보냄
6. 서버에서 쿠키를 읽어 이전 상태 정보를 변경할 필요가 있을 때 쿠키를 업데이트 하여 변경된 쿠키를 HTTP 헤더에 포함시켜 응답

쿠키의 사용 예)

* 방문 사이트에서 로그인 시, “ 아이디와 비밀번호를 저장하시겠습니까?”
* 쇼핑몰 장바구니 기능(배달의 민족 장바구니 기능)
* 자동로그인, 팝업에서 “오늘 더 이상 이 창을 보지 않음” 체크, 쇼핑몰의 장바구니

**Php cookie 생성방법**

* setcookie([쿠키이름], [값], [만료시간 / 초 단위], [경로]);
* 예제) setcookie("cookie", "setcookie", time() + 3600, "/");

**Php cookie 이용방법**

* $\_COOKIE["[쿠키이름]"];
* 예제) echo $\_COOKIE["cookie"];

**Php cookie 삭제방법**

* setcookie([쿠키이름], "", 0, [경로]);
* 예제)  setcookie("cookie", "", 0, "/");

**세션(Session)**

세션이란?

* 세션은 쿠키를 기반하고 있지만, 사용자 정보 파일을 브라우저에 저장하는 쿠키와 달리 세션은 서버 측에서 관리한다.
* 서버에서는 클라이언트를 구분하기 위해 세션 ID를 부여하며 웹 브라우저가 서버에 접속해서 브라우저를 종료할 때까지 인증상태를 유지힌다.
* 사용자에 대한 정보를 서버에 두기 떄문에 쿠키보다 보안에 좋지만, 사용자가 많아질 수록 서버 메모리를 많이 차지하게 된다.
* 동시 접속자 수가 많은 웹 사이트인 경우 서버에 과부하를 주게 되므로 성능 저하의 요인이 된다.
* 클라이언트가 Request를 보내면, 해당 서버의 엔진이 클라리언트에게 유일한 ID를 부여하는데 이것이 세션 ID다.

세션의 동작 방식

1. 클라이언트가 서버에 접속 시 세션 ID를 발급받는다
2. 클라이언트는 세션 ID에 대한 쿠키를 사용해서 저장하고 가지고 있는다.
3. 클라이언트는 서버에 요청할 때 , 이 쿠키의 세션 ID를 서버에 전달해서 사용한다
4. 서버는 세션 ID를 전달 받아서 별다른 작업 없이 세션 ID로 세션에 있는 클라이언트 정보를 가지고 온다.
5. 클라이언트 정보를 가지고 서버 요청을 처리하여 클라이언트에게 응답한다.

세션의 특징

* 보안 면에서 쿠키보다 우수
* 사용자가 많아질수록 서버 메모리 많이 차지하게 된다.
* 세션 ID로 클라이언트 구분해서 클라이언트 요구에 맞는 서비스를 제공

|  |
| --- |
| 쿠키와 세션의 차이   * 쿠키는 서버의 자원을 전혀 사용하지 않으며, 세션은 서버의 자원을 이용한다. 보안 면에서 세션이 더 우수하며, 요청 속도는 쿠키가 세션보다 더 빠르다   세션을 사용하면 좋은데 왜 쿠키를 사용하는가? -> 세션은 서버의 자원을 사용하기 때문에 무분별하게 만들다보면 서버의 메모리가 감당할 수 없어질 수가 있고 속도가 느려질 수 있기 때문이다. |

**Php 세션 사용방법**

Session\_start() -> session\_start()함수는 소스 코드의 최상단에 위치시켜 다른 코드가 그 위에 오지 않도록 처리해야 한다.

**Php 세션 생성 및 수정**-> $\_SESSION["[세션명]"] = "값";

**Php session 삭제** ->   session\_unset();