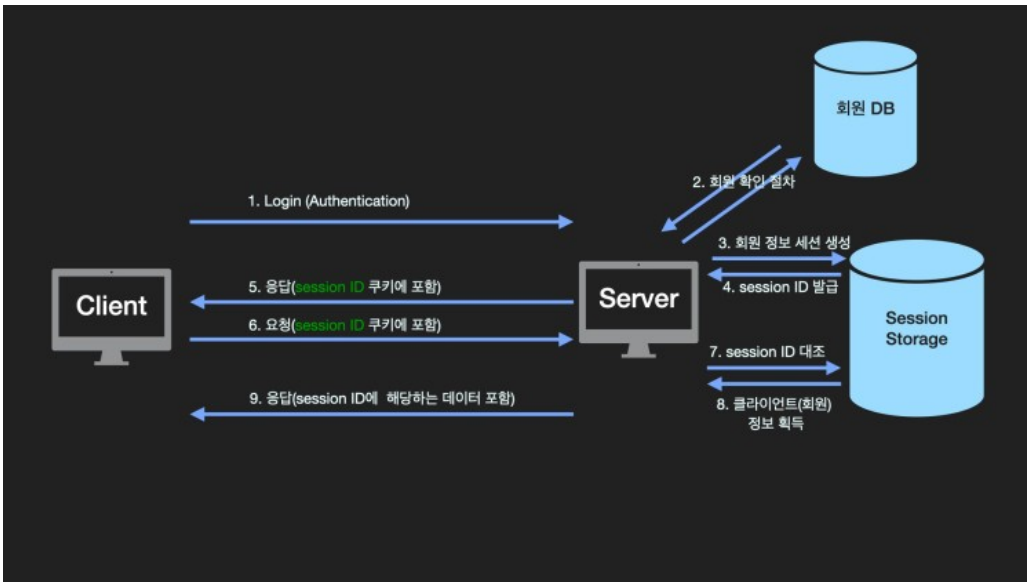
● 쿠키와 세션을 이용한 로그인 방식을 화이트보드에 설명해라

▶



HTTP는 비연결성, 비상태성의 특성을 지니기 때문에 서버는 클라이언트가 로그인을 했더라도 이후 요청 때 해당 클라이언트가 로그인을 했었는지 알 수 없습니다. 하지만 쿠키와 세션을 활용한다면 로그인 상태를 유지할 수 있게 됩니다.

쿠키와 세션을 통한 인증(authentication)과 인가(authorization):
인증(authentication)은 사용자가 누구인지 확인하는 절차입니다. 회원가입과 로그인 과정이 인증의 대표적인 예시입니다. 인가(authorization)는 사용자가 요청하는 것에 대한 권한이 있는지를 확인하는 절차입니다.

세션을 통한 인증, 인가(auth)의 절차는 다음과 같습니다.

1. 클라이언트가 로그인을 하면 서버는 회원 DB에 있는 회원 정보를 대조하여 인증(authentication)을 합니다. 인증(authentication)은 사용자가 누구인지 확인하는 절차입니다. 회원가입과 로그인 과정이 인증의 대표적인 예시입니다.
2. 그런 다음 서버는 회원 정보(클라이언트 정보)를 세션 저장소(session storage)에 생성하고 session ID를 발급합니다.
3. 그런 다음 서버는 발급한 session ID를 http response header 쿠키에 담아서 보냅니다.
4. 클라이언트는 session ID를 쿠키 저장소에 저장하고, 이후에 http request를 보낼 때마다 헤더 쿠키에 session ID를 담아서 보냅니다.
5. 서버에서는 쿠키에 담겨져서 온 session ID에 해당하는 회원 정보를 세션 저장소에서 가져옵니다. 이 절차를 인가(authorization)이라고 합니다. 인가(authorization)는 사용자가 요청하는 것에 대한 권한이 있는지를 확인하는 절차입니다. 해당 Session ID를 가진 클라이언트가 접근할 수 있는 정보들만 섹션 저장소에서 가져오는 것입니다.
6. 마지막으로 서버는 응답 메시지에 session ID에 해당하는 회원 정보를 바탕으로 처리된 데이터를 담아서 클라이언트에 보냅니다. 이를 통해 클라이언트 정보에 따라 맞춤 응답을 할 수 있게 됩니다.

사용자가 로그인을 하면 서버는 session ID를 쿠키로 클라이언트에게 보냅니다. 클라이언트가 session ID를 요청 시마다 헤더에 담아서 보내면, 서버는 session ID에 해당하는 클라이언트 정보를 세션 저장소에서 가져옵니다. 이를 통해 클라이언트 정보에 따라 맞춤 응답을 할 수 있게 됩니다. session ID만 쿠키에 담겨서 요청을 보내기 때문에 요청 때마다 사용자 정보를 쿠키에 담아서 전송하는 것보다 안전합니다 (요청 시마다 아이디와 비밀번호를 계속해서 보내

는 것이 아니라, session ID만 보내면 되기 때문입니다). 하지만 서버에서 세션 저장소를 사용하여 사용자 데이터를 저장해야 되기 때문에 추가적인 저장공간을 필요로 합니다. 또한 session ID만 노출되어 악의를 가진 다든 사용자가 이를 이용해 서버에 요청하면 서버는 구별해낼 수 있는 방법이 없습니다. 이를 Session hijacking 이라고 합니다. 해결책으로는 HTTPS 를 사용하거나 session에 짧은 주기로 만료시간을 설정하는 방법이 있습니다 (주기마다 다시 인증을 해야 하는 방식입니다).

또한 세션과 쿠키를 이용한 로그인 방식은 Load Balancing 및 서버 효율성 관리 및 확장이 어려워질 수 있다는 단점이 있습니다. 여러 대의 서버를 사용하는 시스템의 경우, 유저 로그인 시 해당 유저는 처음 로그인했던 서버로만 요청을 보내도록 설정해야 하기 때문입니다.

Resources: inflearn 개발남 노씨