

### ● 쿠키와 세션의 차이점을 설명해라

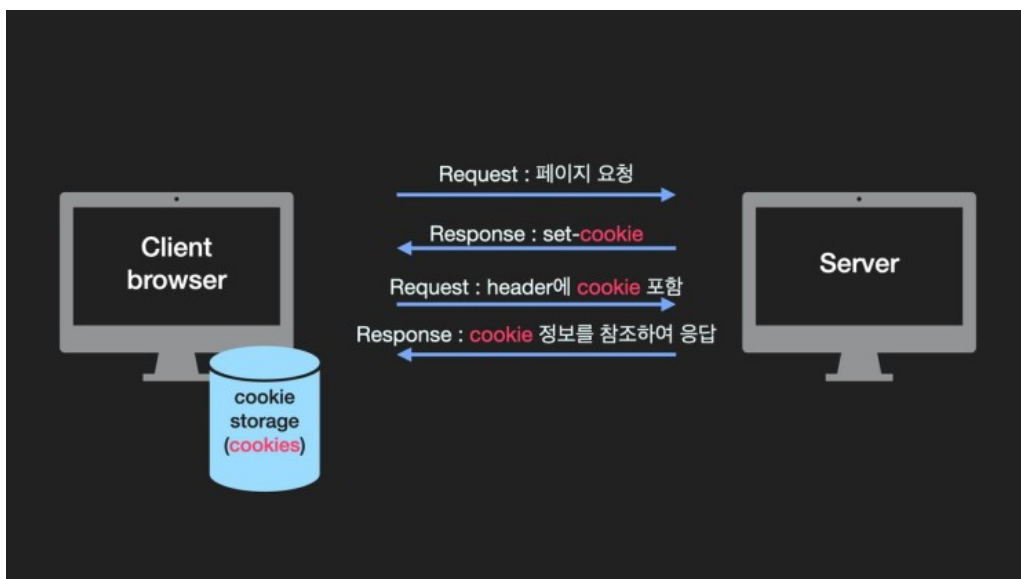
▶ 쿠키는 클라이언트, 즉, 브라우저, 로컬에 key-value 쌍으로 저장되는 데이터 파일로 유효시간 내에서는 브라우저가 종료되어도 계속 유지됩니다. 반면 세션은 브라우저가 종료될 때 삭제되고, 서버에서 직접 해당 세션을 삭제할 수도 있기 때문에 쿠키보다 보안성이 좋습니다. 또한 서버에 데이터를 저장하므로 서버 용량이 허용하는 한에서 제한 없이 데이터를 저장할 수 있다는 장점이 있습니다. 하지만 세션은 서버의 자원을 사용하기 때문에 서버의 부하가 커지고 서버가 느려질 수 있다는 단점이 있습니다. 때문에 세션 대신 쿠키를 사용하면 서버 자원의 낭비를 방지하여 웹사이트 속도를 높일 수 있습니다. 쿠키의 사용 예로는 쇼핑물의 장바구니 기능, 로그인 시 아이디와 비밀번호 저장(또는 자동 로그인), 팝업에서 "오늘 더 이상 이 창을 보지 않음" 체크 등이 있습니다.

### ● 쿠키와 세션을 사용하는 이유

▶ 쿠키와 세션을 사용하는 이유는 HTTP의 connectionless(비연결성), stateless(비상태성)라는 특징 때문입니다. 클라이언트가 요청(request)을 했을 때 그 요청에 맞는 응답(response)을 서버가 보낸 후 연결을 끊고 (단, request header에 keep-alive 값을 통해 connection을 끝내지 않고 유지할 수 있습니다), 서버는 클라이언트에 대한 상태 정보를 유지하지 않기 때문에 알 수 없게 됩니다. 만약 쿠키와 세션을 사용하지 않는다면 이런 비연결성, 비상태성 특징 때문에 어떤 사이트에 로그인을 했음에도 페이지를 이동할 때마다 계속 로그인을 해야됩니다. 또한 쿠키와 세션을 사용하면 로그인을 할때, 아이디 비밀번호 저장을 하여 다음번에 재방문 할 때에도 해당 아이디와 비밀번호를 자동으로 입력하도록 할 수 있고, 이 외에도 쇼핑물의 장바구니 기능과 팝업에서 "오늘 더이상 이 창을 보지 않음" 등의 편의성을 제공할 수 있게 됩니다.

### ● 쿠키(cookie)

▶ 쿠키의 생성과 저장은 구현에 따라 다르지만 원리는 동일합니다. 먼저 서버가 클라이언트로부터 요청을 받았을 때, 클라이언트에 관한 정보를 토대로 쿠키를 구성하고, 클라이언트에게 보내는 응답의 header에 set-cookie를 통해서 쿠키를 담아서 response와 함께 보냅니다. 클라이언트가 응답을 받으면, 브라우저는 쿠키를 쿠키 디렉터리에 저장합니다. 그리고 그 이후에는 사용자가 따로 작업을 하지 않아도 client가 request(요청)을 할 때마다 브라우저가 request header에 cookie를 포함시켜서 서버로 보내고, 서버는 쿠키 정보를 참조하여 응답하는 방식입니다. 이렇게 쿠키는 클라이언트(브라우저)에서 key-value 쌍으로 로컬에 저장되는 데이터 파일입니다. 유효시간 내에서는 브라우저가 종료되어도 계속 유지됩니다.



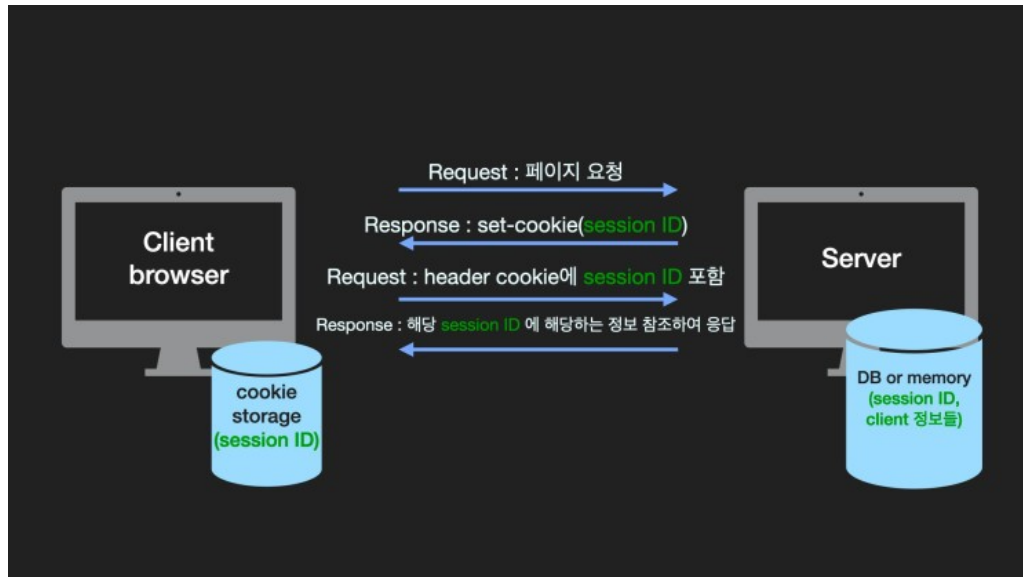
1) client가 request로 페이지를 요청합니다.

2) 서버는 request에 대한 response를 해주면서 클라이언트의 정보를 담은 cookie를 set-cookie를 통해서 response와 함께 보냅니다.

- 3) client browser의 cookie storage에 쿠키가 저장됩니다.
- 4) 그 이후에 client가 request를 할 때마다, request header에 cookie를 포함시켜서 server로 보내면,
- 5) server은 cookie 정보를 참조하여 응답합니다.

#### ● 세션(session)

▶ 세션은 기본적으로 쿠키를 이용하여 구현이 됩니다. 클라이언트를 구분하기 위해 각 클라이언트에게 session ID를 부여하고, 클라이언트는 쿠키에 session ID를 저장해 둡니다. 사용자 정보를 브라우저에 저장하는 쿠키와 달리 세션은 서버측에 저장하여 관리합니다. 세션은 유효시간을 두어 일정 시간 응답이 없다면 끊을 수 있고, 브라우저가 종료될 때까지 인증상태를 유지할 수 있습니다. 사용자 정보를 서버에 두기 때문에 쿠키보다 보안은 좋지만 서버 자원을 차지하기 때문에 서버에 과부하를 줄 수 있고, 성능 저하의 요인이 될 수 있습니다.



- 1) client가 request로 페이지를 요청합니다.
- 2) 서버는 request에 대한 response를 해주면서 클라이언트 ID를 set-cookie를 통해서 response와 함께 보냅니다.
- 3) 클라이언트가 응답을 받으면 session ID가 client browser의 cookie storage에 저장됩니다.
- 4) 그 이후부터 클라이언트 브라우저가 요청할 때 request header cookie에 session ID를 포함시켜서 서버로 전송합니다. (쿠키 정보를 보내지 않고 세션 아이디만 보내므로 보안에 더 유리)
- 5) 서버는 해당 session ID에 해당하는 클라이언트 정보들을 DB나 memory에서 저장해 두고, session ID가 담긴 요청이 올 때마다 해당 세션 아이디에 해당하는 정보들을 참조하여 응답합니다.

Resources: inflearn 개발남 노씨