



쿠키와 세션

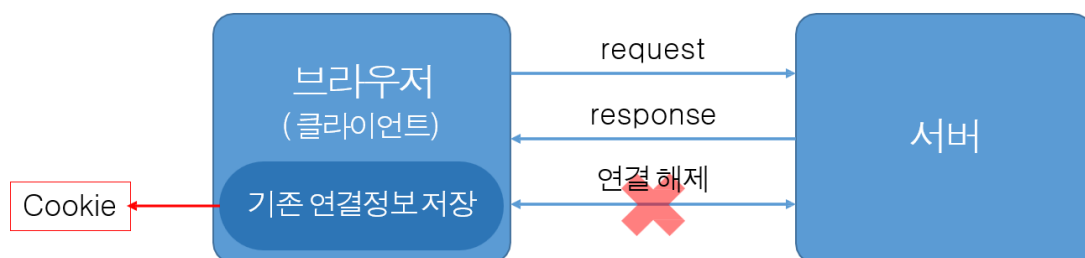
쿠키란 무엇인가?

세션이란 무엇인가?

쿠키 VS 세션

Application 내장 객체

쿠키란 무엇인가?

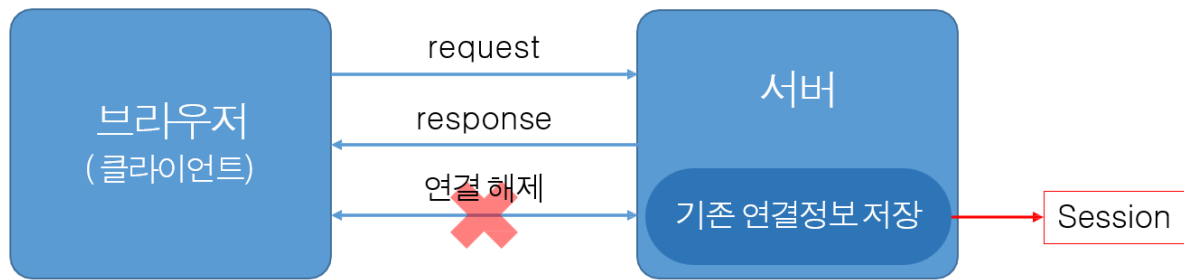


- 쿠키(Cookie)는 웹 서버가 클라이언트(브라우저)에게 정보를 저장하고 관리하기 위해 사용되는 작은 텍스트 파일입니다.
- **클라이언트 측에서 로컬에 저장**되며, 클라이언트가 웹 서버에 요청을 보낼 때 해당 정보를 함께 전송하여 상태를 유지하거나 필요한 데이터를 저장하는 데 사용됩니다.

- 쿠키는 클라이언트에 의해 생성되어 브라우저에 저장되며, 웹 서버로의 요청이 있을 때마다
해당 쿠키가 웹 서버로 전송 됩니다.
- 서버는 전송된 쿠키를 읽어 클라이언트의 정보를 확인하고, 필요한 경우 서버 측에서 데이터를 수정하거나 새로운 쿠키를 생성하여 클라이언트에게 전송할 수 있습니다.
- 쿠키는 다양한 용도로 사용될 수 있습니다. 예를 들어,
 - 로그인 정보를 유지하기 위해 사용자의 인증 상태를 확인하거나,
 - 사용자의 선호 언어, 테마, 글꼴 등의 개인화된 설정을 저장하거나,
 - 사용자의 방문 기록을 추적하거나,
 - 광고 타겟팅을 위해 사용될 수 있습니다.
- 쿠키는 보안과 개인 정보 보호 측면에서 주의해야 합니다. 중요한 개인 정보를 저장하지 않도록 주의해야 하며, 필요한 경우 암호화나 안전한 전송을 위한 HTTPS를 사용하는 등 보안을 강화해야 합니다. 또한, 사용자의 개인 정보를 수집하거나 추적하는 쿠키의 경우에는 개인 정보 보호 관련 법률 및 규정을 준수해야 합니다.

메서드	기능
1. setMaxAge()	쿠키의 유효시간을 설정합니다.
2. setPath()	쿠키사용의 유효디렉토리를 설정합니다.
3. setValue()	쿠키의 값을 설정합니다.
4. setVersion()	쿠키 버전을 설정합니다.
5. getMaxAge()	쿠키 유효기간 정보를 얻습니다.
6. getName()	쿠키의 이름을 얻습니다.
7. getPath()	쿠키사용의 유효디렉토리 정보를 얻습니다.
8. getValue()	쿠키의 값을 얻습니다.
9. getVersion()	쿠키 버전을 얻습니다.

세션이란 무엇인가?



- 세션도 쿠키와 마찬가지로 서버와의 관계를 유지하기 위한 수단입니다.
- 단, 쿠키와 달리 클라이언트의 특정 위치에 저장되는 것이 아니라, **서버상에 객체 형태로** 존재합니다.
- 서버당 하나의 세션 객체를 가질 수 있습니다. (브라우저 별로 서로 다른 세션을 사용)
- 세션 객체는 브라우저 창을 종료하면 소멸합니다.
- 따라서 세션은 서버에서만 접근이 가능하여 쿠키보다 보안이 좋고, 저장할 수 있는 데이터에 한계가 없습니다.
- 세션은 클라이언트의 요청이 발생하면 자동생성되어 고유한 ID값을 클라이언트에게 넘겨주며, 이것은 쿠키에 저장됩니다.
- JSP에서는 session이라는 내장 객체를 지원하여 세션의 속성을 설정할 수 있습니다.

메서드	기능
1. setAttribute()	세션에 데이터를 저장합니다.
2. getAttribute()	세션에 저장되어 있는 데이터를 얻습니다.
3. getAttributeNames()	세션에 저장되어 있는 모든 데이터의 세션 이름(key)을 얻습니다.
4. getId()	자동생성된 세션의 유니크한 아이디를 얻습니다.
5. getCreationTime()	세션이 생성된 시간을 구합니다.
6. getLastAccessedTime()	웹 브라우저가 가장 마지막에 세션에 접근한 시간을 구합니다.
7. setMaxInactiveInterval()	세션의 유효시간을 설정합니다. 초 단위로 기록합니다.
8. getMaxInactiveInterval()	세션의 유효시간을 얻습니다. 가장 최근 요청시점을 기준으로 카운트됩니다.
9. removeAttribute()	특정 세션을 삭제합니다.
10. invalidate()	모든 세션을 삭제합니다.

쿠키 VS 세션

- 쿠키 대신에 세션을 사용하는 가장 큰 이유는 **세션이 쿠키보다 보안에서 앞서기 때문**입니다.
 - 쿠키의 이름이나 데이터는 네트워크를 통해 전달되기 때문에 HTTP 프로토콜을 사용하는 경우 중간에서 누군가가 쿠키의 값을 읽어올 수 있습니다.
 - 그러나 세션은 오직 서버에만 저장되기 때문에 중요한 데이터를 저장하기에 좋습니다.
 - 세션을 사용하는 또 다른 이유는 웹 브라우저가 쿠키를 지원하지 않거나 강제로 사용자가 쿠키를 차단한 경우에도 사용할 수 있다는 점입니다.
- 세션은 여러 서버에서 **공유할 수 없는** 단점이 있습니다. 그러나 쿠키는 도메인을 이용해 쿠키를 여러 도메인에서 **공유할 수 있기 때문에** Naver, Daum과 같은 포털사이트들은 쿠키에 로그인 방식을 저장하는 것을 선호합니다.
- ex) www.naver.com과 mail.naver.com, blog.naver.com의 서버는 각각 다름.

Application 내장 객체

- 특정 웹 어플리케이션에 포함된 모든 JSP 페이지는 하나의 application 기본 객체를 공유합니다.
- application 객체는 웹 어플리케이션 전반에 걸쳐서 사용되는 정보를 담고 있습니다.
- 생명 주기
 - request 객체는 요청 영역마다 생성되고 소멸됩니다.
 - session 객체는 브라우저별로 생성되고 소멸됩니다.
 - application 객체는 서버가 실행될 때 생성되고, 서버가 꺼질 때 소멸됩니다.