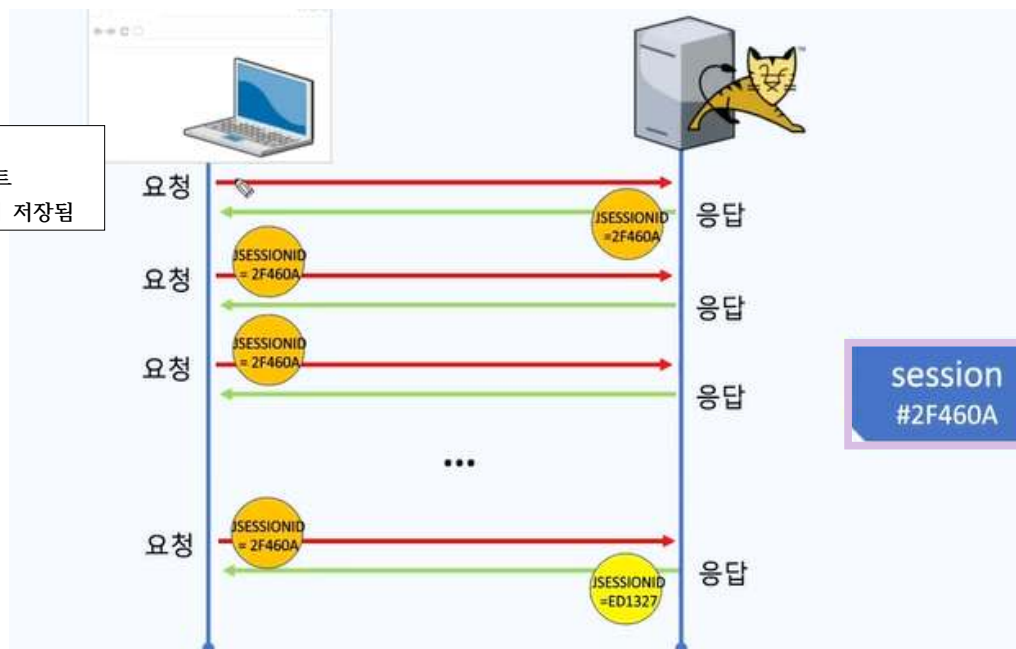


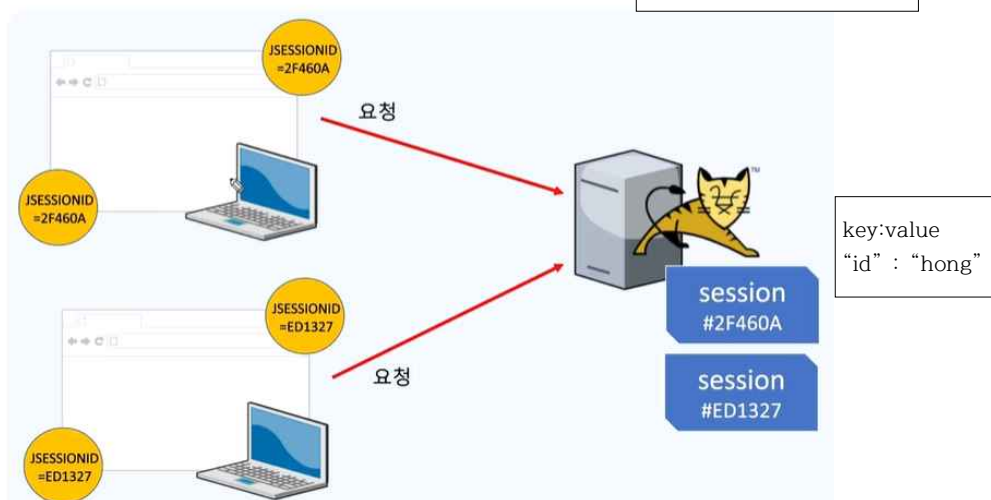
세션(session)

```
POST /ch2/login/login HTTP/1.1
Host: localhost
Connection: keep-alive
Content-Length: 26
Cache-Control: max-age=0
Cookie: id=asdf; JSESSIONID=2F460A779BD0DD7339FB68220B814875
```

쿠키:
클라이언트
컴퓨터에 저장됨



세션객체얻기



세션얼기

```
//로그인 성공했을 때 아이디값을 세션에 저장하기
HttpSession session = request.getSession();
session.setAttribute("id" , "hong" );
```

로그인상태 확인할 때
세션의 id값 확인하기
예시) 주문정보조회할 때 (로그인상태에 따라 제공되는 화면이 달라야 함)

```
//로그인 성공했을 때 아이디값을 세션에 저장하기
HttpSession session = request.getSession();
String id = (String) session.getAttribute("id" );

if( id != null){
    // 현재 로그인이 된 상태임
    response.sendRedirect("/order.do" );
}else{
    // 로그인 상태 아니면 , 로그인
    response.sendRedirect("/login.do");
}
```

로그아웃

```
//로그인 성공했을 때 아이디값을 세션에 저장하기
HttpSession session = request.getSession();
session.invalidate();
```

세션종료

1. 수동 종료

```
HttpSession session = request.getSession();  
session.invalidate(); // 1. 세션을 즉시 종료  
session.setMaxInactiveInterval(30*60); // 2. 예약 종료(30분 후)
```

2. 자동 종료 – web.xml

```
<session-config>  
    <session-timeout>30</session-timeout>  
</session-config>
```

로그아웃을 하지 않아도
지정된 시간이 지나면
세션이 종료될 수 있도록 한다

메서드	설명
String getId()	세션의 ID를 반환
long getLastAccessedTime()	세션 내에서 최근 요청을 받은 시간을 반환
boolean isNew()	새로 생성된 세션인지를 반환. request.getSession()호출 후 사용
void invalidate()	세션 객체를 제거(저장된 객체도 함께)
void setMaxInactiveInterval(int interval)	지정된 시간(초)후에 세션을 종료(예약 종료)
int getMaxInactiveInterval()	예약된 세션 종료 시간을 반환

속성 관련 메서드	설명
void setAttribute(String name, Object value)	지정된 값(value)을 지정된 속성 이름(name)으로 저장
Object getAttribute(String name)	지정된 이름(name)으로 저장된 속성의 값을 반환
void removeAttribute(String name)	지정된 이름(name)의 속성을 삭제