

TOC

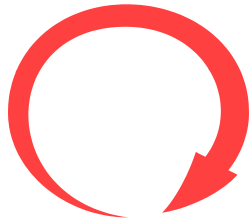
- 쿠키 사용
 - 쿠키 구성
 - 생성
 - 값 사용
 - 도메인/경로/유효시간
 - 쿠키와 헤더
- 세션 사용
 - 세션: session 기본 객체
 - 세션 종료 / 유효 시간

상태 유지 개요

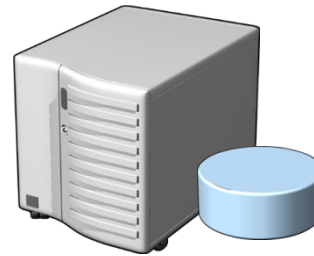
- Http 프로토콜은 모든 요청을 독립적인 개별 요청으로 처리
- 이로 인해 이전 요청과 관련된 정보가 유지되지 않습니다.
- 다수의 상태유지 객체를 통해 상태 정보 유지를 지원



Client-Side State Management



상태 정보



Server-Side State Management

- 상태 유지 객체 종류

| 객체명 | 저장위치 | 유지범위 |
|-------------|--------------------------|-------------|
| cookie | Browser / Client Machine | 동일 브라우저/컴퓨터 |
| session | Server | 동일 브라우저 |
| application | Server | 동일 웹 애플리케이션 |

상태 유지 방법

- ClientSide State Management

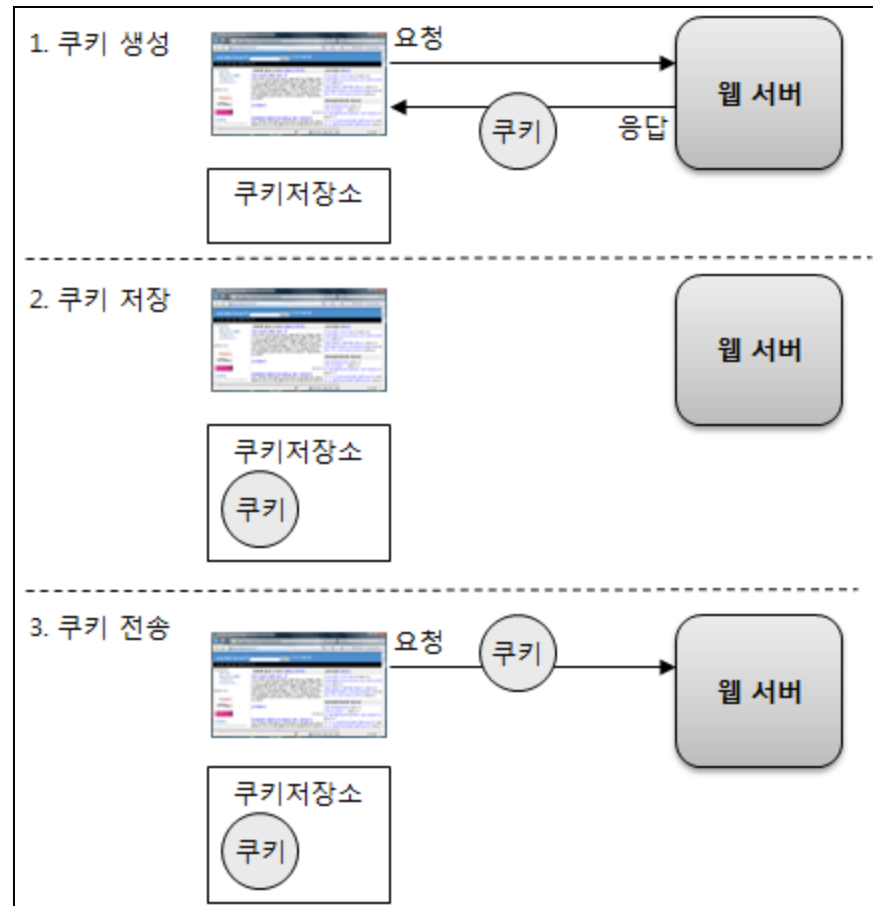


- ServerSide State Management



쿠키(Cookie)

- '쿠키(cookie)'는 웹 브라우저가 보관하고 있는 데이터로서 웹 서버에 요청을 보낼 때 함께 전송
- 동작 방식



쿠키의 구성

- 구성 요소

- 이름 - 각각의 쿠키를 구별하는 데 사용되는 이름
- 값 - 쿠키의 이름과 관련된 값
- 유효시간 - 쿠키의 유지 시간
- 도메인 - 쿠키를 전송할 도메인
- 경로 - 쿠키를 전송할 요청 경로

- 쿠키 이름의 제약

- 쿠키의 이름은 아스키 코드의 알파벳과 숫자만을 포함할 수 있다.
- 콤마(,), 세미콜론(;), 공백(' ') 등의 문자는 포함할 수 없다.
- '\$'로 시작할 수 없다.

JSP에서 쿠키 생성 / 읽기

- Cookie 클래스를 이용해서 쿠키 생성

```
<%  
    Cookie cookie = new Cookie("cookieName", "cookieValue");  
    response.addCookie(cookie);  
%>
```

- 클라이언트가 보낸 쿠키 읽기

```
Cookie[] cookies = request.getCookies();
```

- 읽기 관련 주요 메서드

| 메서드 | 리턴타입 | 설명 |
|------------|--------|--------------|
| getName() | String | 쿠키의 이름을 구한다. |
| getValue() | String | 쿠키의 값을 구한다. |

쿠키 값의 인코딩/디코딩 처리

- 쿠키는 값으로 한글과 같은 문자를 가질 수 없음
 - 쿠키의 값을 인코딩해서 지정할 필요 있음
- 쿠키 값의 처리
 - 값 설정시 : `URLEncoder.encode("값", "euc-kr")`
 - 예, `new Cookie("name", URLEncoder.encode("값", "euc-kr"));`
 - 값 조회시 : `URLDecoder.decode("값", "euc-kr")`
 - `Cookie cookie = ...;`
`String value = URLDecoder.decode(cookie.getValue(), "euc-kr");`

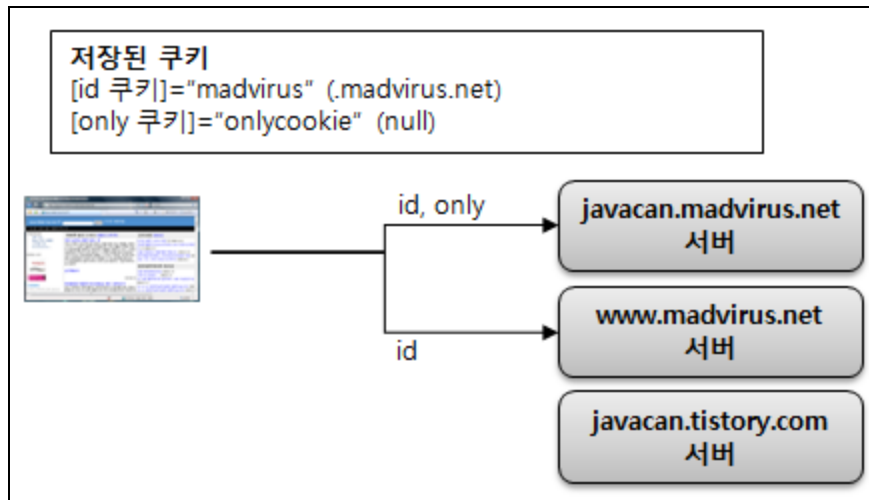
쿠키 값 변경

- 기존에 존재하는 지 확인 후, 쿠키 값 새로 설정

```
Cookie[] cookies = request.getCookies();
if (cookies != null && cookies.length > 0) {
    for (int i = 0 ; i < cookies.length ; i++) {
        if (cookies[i].getName().equals("name")) {
            Cookie cookie = new Cookie(name, value);
            response.addCookie(cookie);
        }
    }
}
```


쿠키의 도메인과 경로

- 도메인 지정시, 해당 도메인에 쿠키 전달
 - Cookie.setDomain()으로 쿠키 설정
 - 도메인 형식
 - .madvirus.net - 점으로 시작하는 경우 관련 도메인에 모두 쿠키를 전송한다.
 - www.madvirus.net - 특정 도메인에 대해서만 쿠키를 전송한다.
- 웹 브라우저는 도메인이 벗어난 쿠키는 저장하지 않음
- 쿠키 도메인에 따라 쿠키가 전달



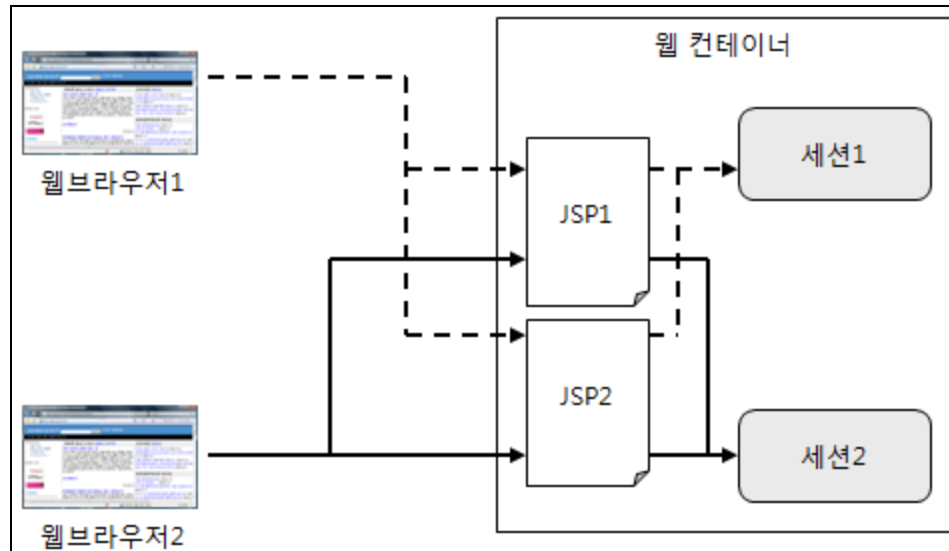
only 쿠키를 javacan.madvirus.net 서버에서 생성했다고 한 경우

쿠키의 경로 / 유효 시간

- 경로 설정시 해당 경로를 기준으로 쿠키 전달
 - 경로 미 설정시, 요청 URL의 경로에 대해서만 쿠키 전달
 - 경로 설정시, 설정한 경로 및 그 하위 경로에 대해서 쿠키 전달
 - `Cookie.setPath()`로 경로 설정
- 유효 시간
 - 유효 시간 미 지정시, 웹 브라우저 닫을 때 쿠키도 함께 삭제
 - `Cookie.setMaxAge()`로 쿠키 유효 시간 설정
 - 유효 시간이 지나지 않을 경우 웹 브라우저를 닫더라도 쿠키가 삭제되지 않고, 이후 웹 브라우저를 열었을 때 해당 쿠키 전송됨
 - 유효 시간 : 초 단위로 설정

세션(session)이란

- 웹 컨테이너에서 클라이언트의 정보를 보관할 때 사용
- 오직 서버에서만 생성
- 클라이언트마다 세션이 생성



세션과 session 기본 객체

- page 디렉티브의 session 속성 값을 true로 지정
 - 세션이 존재하지 않을 경우 세션이 생성되고, 세션이 존재할 경우 이미 생성된 세션을 사용
- session 기본 객체를 이용해서 세션에 접근
 - session의 기본 값은 true이므로 false로 하지 않는 이상 항상 세션 사용

```
<%@ page contentType = ... %>
<%@ page session = "true" %>
<%
    ...
    session.setAttribute("userInfo", userInfo);
    ...
%>
```

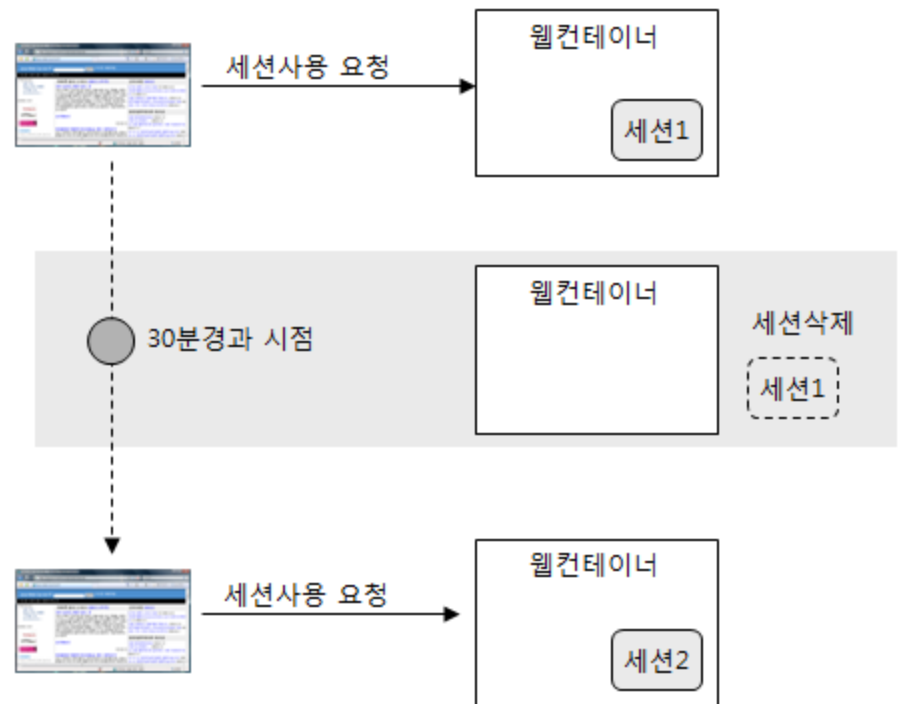
- 속성 이용해서 클라이언트 관련 정보 저장

세션 종료 / 타임 아웃

- session.invalidate() 을 이용해서 세션 종료
 - 세션이 종료되면 기존에 생성된 세션이 삭제
 - 이후 접근 시 새로운 세션 생성 됨
- 마지막 세션 사용 이후 유효 시간이 지나면 자동 종료
 - web.xml 파일에서 지정

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

- 또는 session 기본 객체의
setMaxInactiveInterval()
메서드 이용해서 지정
 - 초 단위 지정



session 기본 객체에 저장될 값 타입

- 필요한 값을 개별 속성으로 저장하는 것 보다는
 - 유지 보수 불편함 (신규 속성 추가시 변경할 코드 증가 등)
- 속성을 하나로 묶은 자바 클래스를 작성해서 저장하는 것이 좋음
 - 유지 보수 편리함