

5. 쿠키와 세션

5.1. HTTP의 특징

웹서비스 환경에서 클라이언트와 서버는 HTTP(Hyper Text Transfer Protocol) 방식으로 통신을 합니다.

HTTP 통신은 Connectionless(비연결성), Stateless(무상태) 이라는 2가지 특성이 있습니다.

5.1.1 Connectionless(비연결성)

비연결성은 클라이언트가 서버에 요청을 하고 서버가 클라이언트에게 응답을 보내면 접속이 끊어지는 특성입니다. 즉 웹 브라우저에서 링크를 클릭(요청)할 때마다 접속되어 서버에서 정보를 전송(응답)한 후 끊기고 하는 상황이 계속 진행됩니다.

5.1.2 Stateless(무상태)

무상태는 통신이 끝나면 상태 정보를 유지하지 않는 특성입니다. 연결을 끊는 순간 클라이언트와 서버의 통신이 끝나며 상태 정보는 유지하지 않는 상황이 발생합니다.

5.2 Session(세션)

세션의 일반적인 의미는 클라이언트의 요청과 서버의 응답 사이에 연결이 유지되는 기간을 뜻합니다. 예를 들어 로그인 후 로그아웃 때까지 연결이 지속되는 기간을 의미합니다.

HTTP의 비연결성, 무상태 특성에 따라 인터넷 쇼핑몰에서 검색중에 장바구니에 담은 물품 기록 등이 사라지게 됩니다. 또한 로그인 정보를 잃어버리기 때문에 다시 로그인해야 합니다. 이러한 문제를 해결하기 위해서는 클라이언트와 서버 간의 상태와 정보 등을 유지하는 인증 방법이 필요한데 이것이 바로 쿠키와 세션입니다.

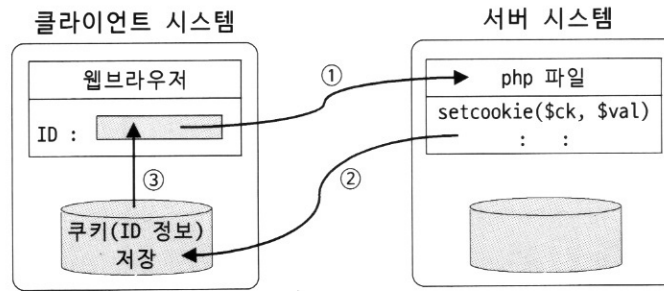
5.3 쿠키(Cookie)

쿠키란 서버에서 웹 클라이언트에 전송하는 작은 데이터 조각을 의미합니다. 이 데이터 조각은 텍스트 파일 형태이며 사이트 개발자가 생성하여 클라이언트 컴퓨터에 저장하게 되는데, 이를 통해 서버는 클라이언트의 연결 상태를 확인합니다.

쿠키 파일은 4KB 크기이며 최대 300개까지 저장할 수 있으며, 이름, 값, 만료날짜, 경로 등의 정보가 기록됩니다.

다음 그림은 ID를 입력했을 경우에 쿠키의 동작 상황을 도식화 예입니다.

- ① : 웹브라우저의 폼에서 ID 정보를 입력하면 php 파일에 전달합니다.
- ② : php 파일에서 ID 정보를 쿠키로 생성하여 클라이언트 시스템(하드디스크)에 저장합니다.
- ③ : 웹브라우저에 다시 그 사이트에 접속하면 클라이언트 시스템에 저장된 ID 정보를 사용합니다.



5.3.1 쿠키의 역할

쿠키에는 로그인 정보나 장바구니 정보 등이 저장됩니다. 서버에 직접 저장하지 않고 클라이언트 컴퓨터에 정보를 저장해서 다음에 접속했을 때 사이트를 좀더 빠르게 사용할 수 있게 하는 역할을 수행합니다. 또한 클라이언트의 정보로 인증을 수행하기 때문에 서버의 역할을 줄이고, 용량도 아낄 수 있고 서버가 해킹을 당해도 클라이언트는 안전하다는 장점이 있습니다.

5.3.2 쿠키의 생성

쿠키의 생성과 전송은 setcookie() 함수를 사용합니다. 쿠키를 생성할 때 이름과 값뿐만 아니라 여러 가지 옵션을 지정할 수 있는데 대표적으로 쿠키의 지속시간 등이 있습니다.

```
bool setcookie($name, $value = "", $expire, $path, $domain, $secure, $http)
```

다음 표는 쿠키 생성 옵션에 대한 설명입니다.

파라미터	설명
name	쿠키의 이름이며 키 역할
value	쿠키의 값. 클라이언트 컴퓨터에 저장. \$COOKIE['name']으로 값 추출
expire	쿠키의 소멸 시간. time()+초단위 형태로 소멸기간 지정
path	쿠키가 적용될 서버에서의 경로. / 경로는 서버 전체에 적용
domain	쿠키가 적용될 도메인 주소의 범위.
secure	HTTPS 방식으로 접속해야 할 경우 true 설정
http	HTTP 방식으로만 접속해야 할 경우 true 설정

5.3.3 쿠키 내용 출력

생성된 쿠키의 값은 슈퍼 전역변수인 \$_COOKIE[]를 이용하여 출력합니다.

```
$_COOKIE['쿠키이름']
```

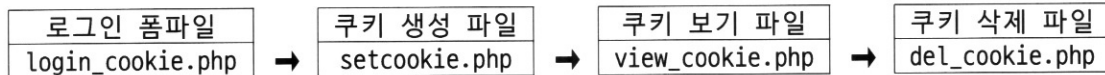
5.3.4 쿠키 삭제

쿠키의 삭제는 쿠키 생성과 마찬가지로 setcookie() 함수를 사용합니다. 삭제할 쿠키 이름만 지정하면 됩니다.

```
bool setcookie('쿠키이름'); 또는 setcookie('쿠키이름', "", time()-1);
```

5.3.5 쿠키 활용 예제

다음은 로그인 폼에서 사용자의 입력 값으로 쿠키를 생성하고 생성된 쿠키를 확인하는 예제입니다.



login_cookie.php		결과
1	<form method=post action=setcookie.php>	<div> <div>wonjuman</div> <div>제출</div> </div>
2	<input type=text name=userid>	
3	<input type=submit>	
4	</form>	

- 1번 라인 : 제출 버튼 클릭시 실행할 파일을 setcookie.php 지정
- 2번 라인 : 입력된 값을 userid 변수에 저장하여 setcookie.php 파일에 전달

setcookie.php		결과
1	\$cookie_name = "login";	<div> 쿠키 보기 </div>
2	\$cookie_value = \$_POST["userid"];	
3	setcookie(\$cookie_name, \$cookie_value, time()+3600);	
4	쿠키 보기	

- 1번 라인 : 쿠키 이름을 login으로 지정하여 \$cookie_name 변수에 저장
- 2번 라인 : 폼 파일의 userid 변수에서 전달된 값을 \$cookie_value 변수에 저장
- 3번 라인 : setcookie() 함수로 3,600초(1시간) 지속되는 쿠키 생성

view_cookie.php		결과
1	<div> <div><body></div> <div>로그인 ID : <?php echo \$_COOKIE['login']; ?></div> <div></body></div> </div>	로그인 ID : wonjuman

- 1번 라인 : 슈퍼 전역변수인 \$_COOKIE[]를 이용하여 login 쿠키 이름의 값을 출력.

del_cookie.php	
1	<div> <div><?php</div> <div>setcookie('login'); //setcookie('login', '', time()-1);</div> <div>?></div> </div>

- 1번 라인 : setcookie() 함수에 쿠키 이름을 지정하여 삭제. time()-1 시간 설정으로 삭제 가능

5.3.6 쿠키 파일 저장경로

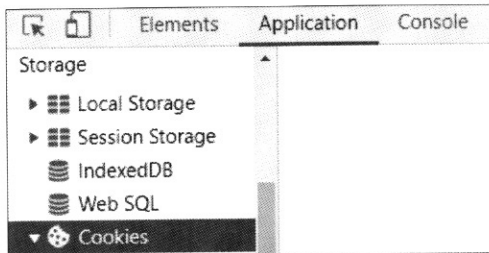
쿠키 파일은 클라이언트 컴퓨터에 저장되는데, 저장 경로는 웹브라우저마다 다릅니다. 크롬의 경우에는 다음과 같습니다.

C:\Users\user_name\AppData\Local\Google\Chrome\User Data\Default\Sessions

위 경로에 있는『Session_XXXXXX』파일이 쿠키 파일입니다. 쿠키 파일은 인코딩되어 있기 때문에 아스키 편집기로는 일부만 확인됩니다.

5.3.7 웹브라우저에서 쿠키 관리

생성된 쿠키 내용은 웹브라우저에서 확인할 수 있습니다. 웹브라우저의 개발자도구(F12)를 실행하면 다음과 같이 Application 탭에 Cookies 항목이 있습니다.



위 쿠키 생성 예제를 이용하면 다음과 같이 쿠키 생성과 관련된 내용들이 표시됩니다.

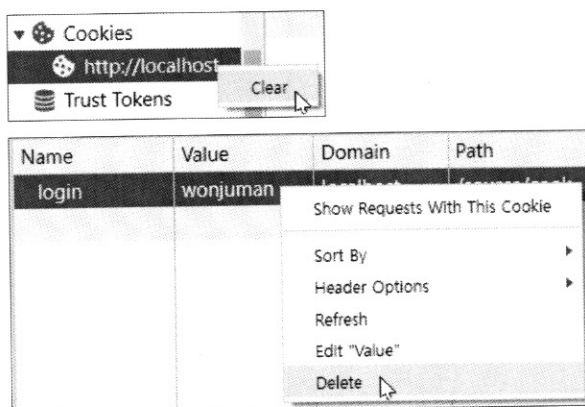
Name	Value	Domain	Path	Expires / Max-Age
login	wonjuman	localhost	/source/cook	2022-10-01T07:44:

<div>Web SQL</div> <div>▼ Cookies</div> <div>http://localhost</div> <div>Trust Tokens</div> <div>Interest Groups</div>	<div>Cache</div> <div>Cache Storage</div> <div>Back/forward cache</div>
--	---

Cookie Value ☒ Show URL decoded
 wonjuman

만약 여러 사람이 사용하는 컴퓨터에 쿠키가 저장되면 사용자의 정보가 노출될 수 있습니다. 이런 상황을 방지하기 위해 앞에서 설명한 쿠키 파일을 삭제해도 되지만, 웹브라우저에서 간단히 삭제할 수 있습니다.

쿠키 사이트 목록에서 마우스 오른쪽 버튼의 Clear 기능을 선택하거나, 쿠키 목록에서 마우스 오른쪽 버튼의 Delete 기능을 선택하면 삭제됩니다.



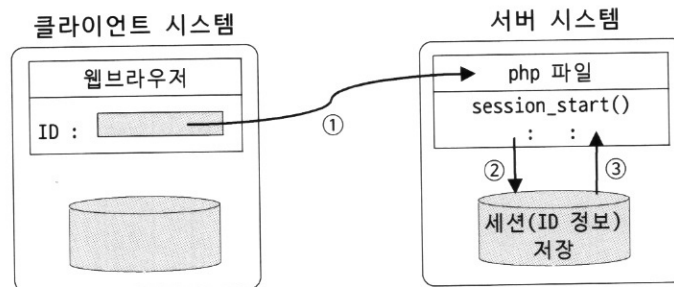
5.4 세션(Session)

세션은 클라이언트와 서버간의 로그인 유지에 관련된 정보를 서버에 저장하는 방식입니다. 세션 관련 정보들이 서버에 저장된다고는 하지만 어떤 클라이언트인지를 식별하기 위해서는 결국 쿠키를 사용합니다.

서버에 저장되는 세션 정보는 세션 상태, 클라이언트 상태, 세션 데이터 등이며, 고유한 세션 ID 값인 세션 쿠키를 클라이언트에 저장하여 서버가 클라이언트를 식별할 수 있도록 합니다. 쿠키와 달리 용량에 제한이 없습니다.

다음 그림은 ID를 입력했을 경우에 세션의 동작 상황을 도식화 예입니다.

- ① : 웹브라우저의 폼에서 ID 정보를 입력하면 php 파일에 전달합니다.
- ② : php 파일에서 ID 정보를 세션으로 생성하여 서버 시스템(하드디스크)에 저장합니다.
- ③ : 웹브라우저에 다시 그 사이트에 접속하면 서버 시스템에 저장된 ID 정보를 사용합니다.



5.4.1 세션의 역할

세션은 쿠키의 역할과 비슷하지만, 접속 관련 정보가 서버에 저장되기 때문에 서버의 자원(CPU, 메모리)을 이용하여 인증을 수행하게 되어 쿠키에 비해 다소 느릴 수 있으며, 서버가 해킹되면 클라이언트 정보도 유출될 수 있습니다.

서버 해킹 문제만 아니라면 세션이 쿠키보다 보안에 더 유리합니다. 클라이언트에 저장된 쿠키 파일은 조작될 가능성이 있어 보안에 취약하지만, 세션은 서버에서 처리하기 때문에 비교적 보안성이 좋습니다.

5.4.2 세션의 생성

세션은 `session_start()` 함수를 이용하여 생성하며, 슈퍼 전역변수인 `$_SESSION[]`을 이용하여 세션 이름과 값을 설정합니다.

```
session_start();
$_SESSION['세션이름'] = '세션값';
```

5.4.3 세션 내용 출력

생성된 세션의 값은 슈퍼 전역변수인 `$_SESSION[]`를 이용하여 출력합니다. 주의할 점은 `session_start()` 함수를 먼저 수행해야 합니다.

```
session_start();
$_SESSION['세션이름'];
```

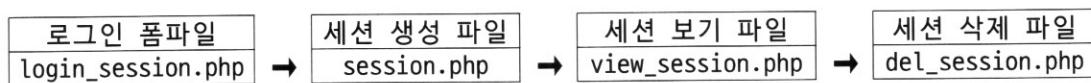
5.4.4 세션 삭제

생성된 세션은 unset() 함수를 이용하여 삭제합니다. 주의할 점은 session_start() 함수를 먼저 수행해야 합니다. 등록된 모든 세션을 삭제하려면 session_unset() 함수를 사용합니다.

```
session_start();
unset($_SESSION['세션이름']); // 특정 세션 삭제
session_unset(); // 모든 세션 삭제
```

5.4.5 세션 활용 예제

다음은 로그인 폼에서 사용자의 입력값으로 쿠키를 생성하고 생성된 쿠키를 확인하는 예제입니다.



login_session.php		결과
1	<form method=post action=session.php>	<div> <div>wonjuman</div> <div>제출</div> </div>
2	<input type=text name=userid>	
3	<input type=submit>	
4	</form>	

- 1번 라인 : 제출 버튼 클릭시 실행할 파일을 session.php 지정
- 2번 라인 : 입력된 값을 userid 변수에 저장하여 session.php 파일에 전달

session.php		결과
1	session_start();	<u>세션 보기</u>
2	\$_SESSION['login'] = \$_POST['userid'];	
3	세션 보기	

- 1번 라인 : 세션 이름을 login으로 지정하여 \$cookie_name 변수에 저장
- 2번 라인 : 폼 파일의 userid 변수에서 전달된 값을 login 세션 이름에 저장

view_session.php		
1	session_start();	
2	echo "로그인 ID :", \$_SESSION['login'];	

- 1번 라인 : 세션 시작 수행
- 2번 라인 : 슈퍼 전역변수인 \$_SESSION[]를 이용하여 login 세션 이름의 값을 출력.

del_session.php		
1	session_start();	
2	unset(\$_SESSION['login']);	

- 1번 라인 : 세션 시작 수행

- 2번 라인 : unset() 함수에 세션 이름을 지정하여 삭제.

세션 삭제후 세션 보기를 시도하면 다음과 같이 출력되어 세션이 삭제된 것이 확인됩니다.

Warning: Undefined array key "login" in

5.4.6 세션 파일 저장경로

세션 파일은 서버 컴퓨터에 저장되는데, 저장 경로는 웹서버마다 다를 수 있습니다. 아파치 웹서의 경우에는 다음과 같습니다.

H:\xampp\tmp

위 경로에 있는 『sess_57vo921t7xxxx』 파일이 세션 파일입니다. 세션 파일은 서버 관리자가 아스키 편집기로 확인할 수 있습니다. 서버 컴퓨터에 저장된 세션 파일의 내용은 다음의 형태입니다.

login|s:8:"wonjuman";

5.4.7 웹브라우저에서 세션 관리

쿠키와 달리 사용자가 입력한 세션 정보는 웹브라우저에서 확인할 수 없습니다. 대신 세션 이름은 PHPSESSID로 표시되며 값은 57vo921t7xxxx 형태로 나타납니다(세션 파일명 sess_가 제외된 형태).

웹브라우저의 개발자도구(F12)를 이용하면 다음과 같이 표시됩니다.

▼ Cookies		Name	Value	Domain	Path	Expires / Max-Age
http://localhost		PHPSESSID	57vo921t7em1...	localhost	/	Session
Trust Tokens						
Interest Groups						
Cache		Cookie Value <input checked="" type="checkbox"/> Show URL decoded				
Cache Storage		57vo921t7em19o4jj6uf848c7g				

사용자 입력 정보가 표지되지는 않지만 클라이언트의 웹브라우저에서도 세션 정보를 삭제할 수 있습니다.

Name	Value	Domain	Path
PHPSESSID	nh12i3ol4ccai...		
Show Requests With This Cookie			
Sort By ▶			
Header Options ▶			
Refresh			
Edit "Value"			
Delete			
Cookie Value <input checked="" type="checkbox"/> Show UR			
nh12i3ol4ccai1rdrbni483o99			