

스프링 프레임워크 ver.3

세션(Session)과 쿠키(Cookie)

Connectionless Protocol

요청(request), 응답(Response) 후 바로 연결을 해제한다. 서버의 자원을 효율적으로 사용하기 위함. 문제점이 로그인 상태 유지, 장바구니 기능 등을 계속 인증해야 한다.

세션 - 서버, 쿠키 - 클라이언트에서 연결 정보를 관리

세션(Session)

- 세션 선언, 생성

HttpServletRequest

파라미터로 HttpServletRequest를 받은 후 getSession()으로 세션 얻음.

HttpSession

파라미터로 HttpSession을 받아 세션 사용.

```
@Autowired
```

```
private HttpSession session;
```

- 세션 삭제 등

```
session.invalidate();
```

세션 메소드	기능
<code>getId()</code>	세션 ID를 반환한다.
<code>setAttribute()</code>	세션 객체에 속성을 저장한다.
<code>getAttribute()</code>	세션 객체에 저장된 속성을 반환한다.
<code>removeAttribute()</code>	세션 객체에 저장된 속성을 제거한다.
<code>setMaxInactiveInterval()</code>	세션 객체의 유지시간을 설정한다.
<code>getMaxInactiveInterval()</code>	세션 객체의 유지시간을 반환한다.
<code>invalidate()</code>	세션 객체의 모든 정보를 삭제한다.

쿠키(Cookie)

HttpServletResponse에 쿠키를 담는다.

```

@RequestMapping("/main")
public String mallMain(Mall mall, HttpServletResponse response) {
    Cookie genderCookie = new Cookie("gender", mall.getGender());
    if(mall.isCookieDel()) {
        genderCookie.setMaxAge(0);
        mall.setGender(null);
    } else {
        genderCookie.setMaxAge(60*60*24*30);
    }
    response.addCookie(genderCookie);

    return "/mall/main";
}

```

쿠키 생성

쿠키의 setMaxAge 쿠키의 유지 시간을 정해준 후에

response에 쿠키를 실어준다.

세션은 서버에서 저장되어 request 객체에서 session을 생성해 받아서 저장하고

쿠키는 클라이언트에 저장되어 response 객체에 실어서 클라이언트에 저장하게 한다.

-
- 쿠키 활용

```
@RequestMapping("/index")
public String mallIndex(Mall mall,
    @CookieValue(value="gender", required=false) Cookie genderCookie,
    HttpServletRequest request) {

    if(genderCookie != null)
        mall.setGender(genderCookie.getValue());

    return "/mall/index";
}
```

쿠키를 받아오는 것

```
@CookieValue(value="저장값(키)" required=false)Cookie genderCookie
```

required 속성은 default =true 이기 때문에 쿠키가 없다면 exception이 발생하기 때문에 false 설정을 습관화 해준다.
그 후에 getValue()로 가져온다.

쿠키는 로컬에 로그가 남는다.

리다이렉트(Redirect), 인터셉터(Interceptor)

컨트롤러에서 뷰를 분기하는 방법 = redirect

컨트롤러 실행 전, 후 특정작업 = interceptor

리다이렉트

```
return "redirect:이동할 분기";
```

re(다시) + 지시하다(direct) 다시 지시하는 것

클라이언트 요청을 서버가 바꿔는 작업등이 있으면 redirect를 사용해서 중복해서 작업이 실행되지 않게 한다. 등등 특정조건에 만족하지 않으면 다른 분기로 이동시킴

인터셉터(Interceptor)



```
public class MemberLoginInterceptor extends HandlerInterceptorAdapter {  
    @Override  
    public boolean preHandle(HttpServletRequest request,  
        HttpServletResponse response, Object handler) throws Exception {
```

```
        HttpSession session = request.getSession(false);  
        if(session != null) {  
            Object obj = session.getAttribute("member");  
            if(obj != null)  
                return true;  
        }  
        response.sendRedirect(request.getContextPath() + "/");  
        return false;  
    }
```

인터셉트 구현 후에 사용할 때

인터셉터 적용법 적용할 컨트롤러 설정

-Servlet-Context.xml 에서 인터셉트 설정

```
<interceptors>
    <interceptor>
        <mapping path="/member/modifyForm"/>
        <mapping path="/member/removeForm"/>
        <beans:bean class="com.bs.lec21.member.MemberLoginInterceptor"/>
    </interceptor>
</interceptors>
```

설정할 인터셉터 빈 생성후 mapping 하여 적용될 Mapping path 를 적용해주면

pre, post, afterCompletion 등을 그 매핑된 메소드가 실행될때 사용된다.

```
<mapping path="/member/**"/>
<exclude-mapping path="/member/joinForm"/>
<exclude-mapping path="/member/join"/>
<exclude-mapping path="/member/loginForm"/>
<exclude-mapping path="/member/login"/>
<exclude-mapping path="/member/logout"/>
<exclude-mapping path="/member/modify"/>
<exclude-mapping path="/member/remove"/> [
```

전체를 지정해주고 제외하는
방식으로도 사용 할 수 있다.

데이터 베이스

```
C:\> C:\WINDOWS\system32\cmd.exe - sqlplus
Microsoft Windows [Version 10.0.17134.165]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\>Users\ho_msi>sqlplus

SQL*Plus: Release 11.2.0.2.0 Production on 화 7월 17 10:14:50 2018
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Enter user-name: system
Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
SQL> create user scott identified by tiger;
User created.
SQL> grant connect, resource to scott;
Grant succeeded.
```

계정 생성

계정에 권한 주기

삭제 => drop user scott cascade;

기본 CRUD

CREATE - INSERT (생성)

READ - SELECT (조회)

UPDATE - UPDATE (갱신)

DELETE - DELETE(삭제)

JDBC 사용할 때 드라이버 로딩, 연결, 연결 끊기 동일 부분이 계속해서 진행된다.

=> JDBC TEMPLATE 사용

— Datasource 를 통해 연결에 관련된 정보를 관리하고

template을 통해 sql, pstmt 를 사용해서 쿼리문을 수행한다.

CP(Connection pool)

웹 컨테이너(WAS)가 실행되면서 DB와 미리 connection(연결)을 해놓은 객체들을 pool에 저장해두었다가.

클라이언트 요청이 오면 connection을 빌려주고, 처리가 끝나면 다시 connection을 반납받아 pool에 저장하는 방식을 말합니다.

-spring 설정파일에서 datasource를 생성하고 주입받아 사용 (java파일에서 어노테이션으로 빈을 생성 할 수 있다.

질문

1. 컨트롤러에서 여러 서비스를 사용하는게 좋은가?

서비스에서 여러 맵퍼(DAO)를 사용하는게 좋은가?

ex) MarketController 에 BankService, BasketService 사용

or

MarketService 생성 MarketService에 BankMapper, BasketMapper 사용

MarketController 에 MarketService 사용

2. 회원이 로그인된 정보등을 실무에서도 Session을 사용하는지

3. 외부 간편 로그인 API 사용시 데이터를 (전화번호 이메일 주소 등)을 따로 서버 DB에
저장하나? 간편로그인시 - 간편 회원가입 설계 문제

-원티드, 리멤버

세션 4.