

spring 6강 – 회원가입

양 명 속

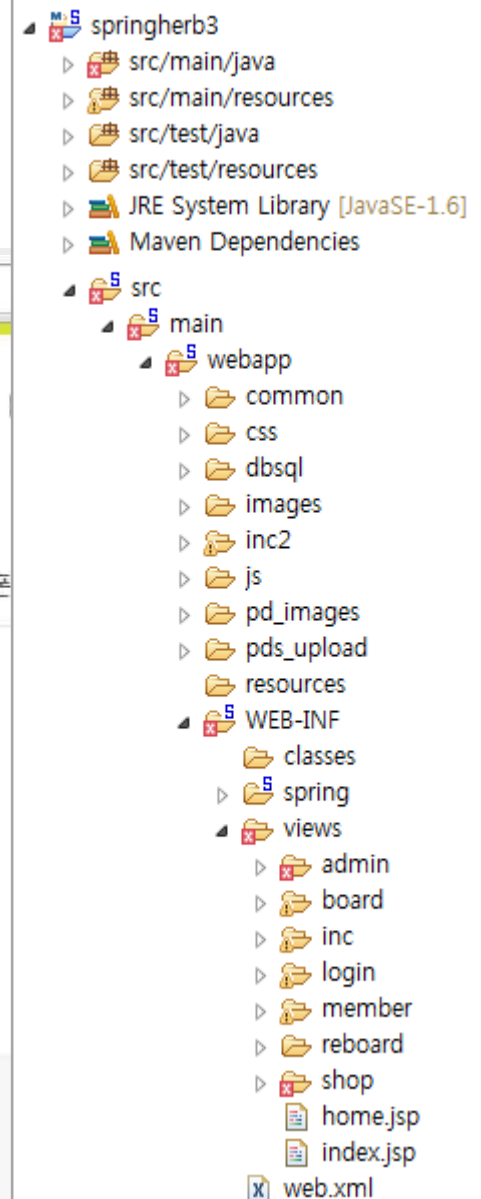
[now4ever7@gmail.com]



목차

- 회원가입
- HandlerInterceptor를 통한 요청 가로채기

템플릿 페이지



```

<%@ page language="java" contentType="text/html; charset=utf-8"    pageEncoding="utf-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html><html lang="ko"><head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>herbmall</title>
<link rel="stylesheet" type="text/css" href="${pageContext.request.contextPath}/css/mainstyle.css"/>
<link rel="stylesheet" type="text/css" href="${pageContext.request.contextPath}/css/clear.css" />
<link rel="stylesheet" type="text/css" href="${pageContext.request.contextPath}/css/layout.css" />
<link rel="stylesheet" type="text/css" href="${pageContext.request.contextPath}/css/mystyle.css" />
<script type="text/javascript" src="<c:url value='/jquery/jquery-1.11.2.min.js'/>"></script>
<!-- [if lt IE 9]>
    <script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>
<body>
    <!-- 상단 띠 이미지 영역 -->
    <div id="topImg" class="top_img"
    style="background:url(${pageContext.request.contextPath}/images/bg_top1.PNG) repeat-x">&nbsp;</div>
    <div id="wrap">
        <!-- header -->
        <header id="header">
            <h1><a href="<c:url value='/index.do' />">
            </a></h1>
            <nav id="headerRight">
                <ul class="views">
                    <c:if test="${empty sessionScope.userid}">
                        <li><a href="<c:url value='/login/login.do' />">로그인</a> | </li>
                        <li><a href="<c:url value='/member/agreement.do' />">회원가입</a> | </li>

                    </c:if>
                    <!-- 로그인 된 경우 -->
                    <c:if test="${!empty sessionScope.userid}">

```

```

        <li><span style="color:gray;font-size:1.0em">
            (${sessionScope.userName} 님) </span></li>
        <li><a href="<c:url value='/login/logout.do' />">로그아웃</a> | </li>
        <li><a href="<c:url value='/member/memberEdit.do' />">회원정보변경</a> |</li>
        <li><a href="<c:url value='/member/memberOut.do' />">회원탈퇴</a> |</li>
    </c:if>
    <li><a href="<c:url value='/shop/cart/cartList.do' />">장바구니</a> |</li>
    <li><a href="<c:url value='/shop/order/orderList.do' />">주문내역</a> |</li>
    <li><a href="<c:url value='/shop/mypage.do' />">마이페이지</a> |</li>
    <li><a href="<c:url value='/board/boardList.do' />">고객센터</a></li>
</ul>
</nav>
</header>
<nav id="top_navi">
    <div id='cssmenu'>
        <ul>
            <li><a href="#"><span>베스트</span></a></li>
            <li><a href="#"><span>자유게시판</span></a></li>
            <li><a href="#"><span>이용후기</span></a></li>
            <li><a href="#"><span>Q&A</span></a></li>
            <li><a href="#"><span>FAQ</span></a></li>
            <li class='last'><a href="#"><span>이벤트&쿠폰</span></a></li>
        </ul>
    </div>
</nav>
<!-- 라인 이미지 영역-->
<div id="topLine" class="top_Line" style="background:url(${pageContext.request.contextPath}/images/line6.jpg)
repeat-x;font-size:7px">&nbsp; </div>
<!-- container -->
<div id="container">
    <nav>
        <dl id="leftNavi"><!-- category list -->
            <dt>허브</dt>
            <dd><a href="default.jsp">허브차</a></dd>
            <dd><a href="default.jsp">아로마 오일</a></dd>
            <dd><a href="default.jsp">향초</a></dd>
            <dd><a href="default.jsp">허브비누</a></dd>
        </dl></nav>
    <section id="contents">

```



authority table – 권한 테이블

--권한 테이블

```
drop table authority cascade constraint;
```

```
create table authority
```

```
(
```

```
    authCode      varchar2(30) primary key , --권한코드
```

```
    authName      varchar2(100) not null , --권한명
```

```
    authDesc      varchar2(200) null , --권한설명
```

```
    authLevel     number(3), --레벨 순위 --1(ADMIN), 2(ASSISTANT), 3(STAFF)
```

```
    regdate       date default sysdate
```

```
);
```

```
INSERT INTO authority VALUES ('ADMIN','관리자', 'manager',1, SYSDATE);
```

```
INSERT INTO authority VALUES ('ASSISTANT','부관리자', 'assistant manager',2, SYSDATE);
```

```
INSERT INTO authority VALUES ('STAFF','스태프', 'staff',3, SYSDATE);
```

```

create table member2
(
    no          number          primary key,
    userid      varchar2(20)    unique not null,
    name        varchar2(20)    not null,
    pwd         varchar2(15)    not null,
    email1      varchar2(30)    null,
    email2      varchar2(20)    null,
    hp1         varchar2(3)     null,
    hp2         varchar2(4)     null,
    hp3         varchar2(4)     null,
    zipcode     varchar2(7)     null,
    address     varchar2(200)    null,
    addressDetail varchar2(50)  null,
    regdate     date            default sysdate,
    mileage     number          default 0,
    outdate     date            null
);

```

```

create sequence member2_seq
increment by 1
start with 1
nocache;

```



MemberVO

```
public class MemberVO {  
    //멤버변수  
    private int no;  
    private String userid;  
    private String name;  
    private String pwd;  
  
    private String email1;  
    private String email2;  
  
    private String hp1;  
    private String hp2;  
    private String hp3;  
  
    private String zipcode;  
    private String address;  
    private String addressDetail;  
    private Timestamp regdate;  
    private int mileage;  
    private Timestamp outdate; //탈퇴일  
}
```




IndexController

```
package com.hy.herb.controller;
```

```
@Controller
```

```
public class IndexController {
```

```
    private static final Logger logger = LoggerFactory.getLogger(IndexController.class);
```

```
    @RequestMapping("/index.do")
```

```
    public String index(){
```

```
        //index.jsp 메인 페이지 보여주기
```

```
        logger.debug("IndexController : index()호출");
```

```
        return "/index";
```

```
    }
```

```
}
```

회원가입 페이지-agreement.jsp

Herb mall

[관리자로그인](#) | [로그인](#) | [회원가입](#) | [장바구니](#) | [주문내역](#) | [마이페이지](#) | [고객](#)

[베스트](#)

[자유게시판](#)

[이용후기](#)

[Q&A](#)

[FAQ](#)

[이벤트&쿠폰](#)

허브

회원 약관

허브차

아로마오일

허브향초

인터넷 쇼핑몰 『테스트쇼핑몰』회원 약관

제1조(목적)

이 약관은 (주)테스트닷컴 회사(전자상거래 사업자)가 운영하는 테스트쇼핑몰(이하 "몰"이라 한다)에서 제공하는 인터넷 「PC통신, 무선 등을 이용하는 전자상거래에 대해서도 그 성질에 반하지 않는 한 이 약관을 준용합니다」

제2조(정의)

- ① "몰" 이란 (주)테스트닷컴 회사가 재화 또는 용역(이하 "재화등"이라 함)을 이용자에게 제공하기 위하여 컴퓨터등 정보통신설비를 이용하여 재화 또는 용역을 거래하는 것만을 그 한으로 한정하지 아니하며, 전자상거래에 관한 법률에 따른 전자상거래의 개념에 포함된 전기통신매매업에 포함되는 것까지 포함한다.
- ② "이용자"란 "몰"에 접속하여 이 약관에 따라 "몰"이 제공하는 서비스를 받는 회원 및 비회원을 말한다.
- ③ '회원'이라 함은 "몰"에 개인정보를 제공하여 회원등록을 한 자로서, "몰"의 정보를 지속적으로 제공받으며, "몰"이 지정한 약관과 조건 등에 동의한 자를 말한다.
- ④ '비회원'이라 함은 회원에 가입하지 않고 "몰"이 제공하는 서비스를 이용하는 자를 말한다.

☐ 약관에 동의합니다.

확인

취소



agreement.jsp

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```
<%@ include file="../inc/top.jsp" %>
```

```
<script type="text/javascript">
```

```
    function send(frm){
```

```
        if(!frm.chkAgree.checked){
```

```
            alert("약관에 동의하셔야 합니다!!");
```

```
            frm.chkAgree.focus();
```

```
            return false;
```

```
        }
```

```
    return true;
```

```
}
```

```
</script>
```

```
<h2>회원 약관</h2><br>
```

```
<iframe src='<c:url value="/inc2/provision.html"/>' width="700px" height="300px"></iframe>
```

```
<br><br>
```

```
$(document).ready(function(){
    $("#frmAgree").submit(function(){
        if(!$("#chkAgree").is(":checked")){
            alert("약관에 동의하셔야 합니다.");
            $("#chkAgree").focus();
            return false;
        }
    });
});
```



agreement.jsp

```
<div style="width:700px">
  <form id="frmAgree" name="frmAgree" method="post"
    action='<c:url value="/member/register.do"/>' onsubmit="return send(this)">
    <div style="text-align:right">
      <input type="checkbox" name="chkAgree">
      <label for="chkAgree">약관에 동의합니다.</label>
    </div>
    <div style="text-align:center">
      <input type="submit" value="확인" >
      <input type="reset" value="취소">
    </div>
  </form>
</div>

<%@ include file="../inc/bottom.jsp" %>
```



/inc2/provision.html

```
<!DOCTYPE html>
<html><head><meta charset="utf-8">
<title>회원약관</title>
<style type="text/css">
  pre{
    font-family: 맑은 고딕, 한컴돋움, 돋움;
    font-size: 0.8em;
    color:#555555;
  }
</style>
</head>
<body>
<pre>
인터넷 쇼핑몰 『테스트쇼핑몰』 회원 약관
```

제1조(목적)

이 약관은 (주)테스트닷컴 회사(전자상거래 사업자)가 운영하는 테스트쇼핑몰(이하 "몰"이라 한다)에서 제공하는 인터넷 관련 서비스(이하 "서비스"라 한다)를 이용함에 있어 사이버 몰과 이용자의 권리·의무 및 책임사항을 규정함을 목적으로 합니다.

```
@Controller
@RequestMapping(value="/member")
public class MemberController {
    @Autowired MemberService memberService;

    private static final Logger logger = LoggerFactory.getLogger(MemberController.class);

    @RequestMapping(value="/agreement.do")
    public String agreementGet(){
        //회원 약관 페이지 보여주기
        logger.debug("회원 약관 화면 보여주기");
        return "/member/agreement";
    }

    @RequestMapping(value="/register.do")
    public String agreementPost(){
        //회원 가입 페이지 보여주기
        logger.debug("회원 가입 페이지 보여주기");

        return "/member/register";
    }

    @RequestMapping("/memberWrite.do")
    public String insertMember(@ModelAttribute MemberVO vo, String email3){
        //회원 가입 처리
        //1. 파라미터
        logger.debug("파라미터 : vo= {}", vo);

        String hp1 = vo.getHp1();
        String hp2 = vo.getHp2();
        String hp3 = vo.getHp3();
    }
}
```

```

if((hp2==null || hp2.isEmpty() || hp3==null || hp3.isEmpty())){
    hp1="";
    hp2="";
    hp3="";
}
vo.setHp1(hp1);
vo.setHp2(hp2);
vo.setHp3(hp3);

```

MemberController

```

String email1 = vo.getEmail1();
String email2 = vo.getEmail2();
if(email1==null || email1.isEmpty()){
    email1= "";
    email2= "";
}else{
    if(email2.equals("etc")){
        email2 = email3;
    }
}
vo.setEmail1(email1);
vo.setEmail2(email2);

```

```

//2. db작업 - insert
int n= memberService.insertMember(vo);
logger.debug("회원가입 결과, n={}", n);

```

```

//3. 결과 저장, 리턴
return "redirect:/index.do";

```

```

}

```

[베스트](#)[자유게시판](#)[이용후기](#)[Q&A](#)[FAQ](#)[이벤트&쿠폰](#)[허브](#)[허브차](#)[아로마오일](#)[허브향초](#)

회원 가입

성명

회원ID

중복확인

비밀번호

비밀번호 확인

주소

우편번호 찾기

핸드폰

010 ▼

-

-

이메일 주소

@

naver.com ▼

회원가입


```

<div class="divForm">
<form name="frm1" method="post" action='<c:url value="/member/memberWrite.do"/>'
    onsubmit="return send(this)">
<fieldset>
    <legend>회원 가입</legend>
    <p>
        <label for="name">성명</label><input type="text" name="name" id="name">
    </p>
    ....

    <input type="hidden" name="checkIdYn">
</form>
<%@ include file="../inc/bottom.jsp"%>

```

```

function useridCheck(){
    //아이디 중복 확인 창 띄우기
    var userid = frm1.userid.value;
    window.open(
        "<c:url value='/member/checkId.do?userid="+ userid+"/>", "idWin",
        "width=400, height=200, scrollbars=yes, resizable=yes");
}
function getZipcode(){
    //우편번호 찾기 창 띄우기
    //window.open("url","name","option");
    window.open("${pageContext.request.contextPath}/member/zipcode.do","win",
        "left=0, top=0, width=500, height=500, scrollbars=yes, resizable=yes");
}

```

Member.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//ibatis.apache.org//DTD Mapper 3.0//EN" "http://ibatis.apache.org/dtd/ibatis-3-
mapper.dtd">
<mapper namespace="config.mybatis.mapper.oracle.Member">
    <select id="zipcodeList" parameterType="string" resultType="zipcodeBean">
        select * from zipcode
        where dong like '%' || #{dong} || '%'
    </select>

    <select id="memberCheckId" parameterType="string" resultType="int">
        select count(*) from member2
        where userid=#{userid}
    </select>

    <insert id="memberInsert" parameterType="memberBean">
        <selectKey keyProperty="no" resultType="int" order="BEFORE">
            select member2_seq.nextval from dual
        </selectKey>
        insert into member2(no, userid, name, pwd, email1, email2,
            hp1, hp2, hp3, zipcode, address, addressDetail, authCode)
        values(#{no}, #{userid}, #{name}, #{pwd}, #{email1}, #{email2},
            #{hp1}, #{hp2}, #{hp3}, #{zipcode}, #{address}, #{addressDetail}, #{authCode})
    </insert>

    <select id="memberGetPwd" parameterType="string" resultType="string">
        select pwd from member2
        where userid=#{userid} and outdate is null
    </select>
```

```
<update id="memberDelete" parameterType="string">
    update member2
    set outdate = sysdate
    where userid=#{userid} and outdate is null
</update>
```

```
<select id="memberDetail" parameterType="string" resultType="memberBean">
    select * from member2
    where userid=#{userid} and outdate is null
</select>
```

```
<update id="updateMember" parameterType="memberBean">
    update member2
    set email1=#{email1}, email2=#{email2}, hp1=#{hp1}, hp2=#{hp2}, hp3=#{hp3},
        zipcode=#{zipcode}, address=#{address}, addressDetail=#{addressDetail}
    where userid=#{userid} and outdate is null
</update>
```

```
</mapper>
```

@Repository

```
public class MemberDAOMybatis extends SqlSessionDaoSupport implements MemberDAO{
    private static final Logger logger = LoggerFactory.getLogger(MemberDAOIbatis.class);
    private String namespace="config.mybatis.mapper.oracle.Member";
    //메서드
    public int checkUserId(String userid) {
        //아이디 중복확인에서 사용
        //해당 id가 존재하는지 체크
        int cnt = (Integer) getSqlSession().selectOne(namespace+ ".memberCheckId", userid);
        return cnt;
    }

    public int insertMember(MemberBean bean) {
        int n = (Integer) getSqlSession().insert(namespace+ ".memberInsert", bean);
        return n;
    }

    public String checkIdPwd(String userid) {
        String pwd = (String) getSqlSession().selectOne(namespace+ ".memberGetPwd", userid);
        return pwd;
    }

    public int delMember(String userid){
        int n=getSqlSession().update(namespace+ ".memberDelete", userid);
        return n;
    }
}
```

memberDAOMybatis

```
public MemberBean selectMember(String userid){
    MemberBean bean
        = (MemberBean) getSqlSession().selectOne(namespace+ ".memberDetail", userid);
    return bean;
}

public int updateMember(MemberBean bean){
    int n      = getSqlSession().update(namespace+ ".updateMember", bean);
    return n;
}

} //class
```

memberDAOMybatis

@Service

@Transactional

MemberServiceImpl

```
public class MemberServiceImpl implements MemberService{
    @Resource(name="memberDAOMybatis") MemberDAO memberDao;
    private static final Logger logger = LoggerFactory.getLogger(MemberServiceImpl.class);

    public boolean checkUserid(String userid) {
        boolean result=false;
        int cnt = memberDao.checkUserid(userid);
        if(cnt>0){ //해당 아이디가 존재하는 경우
            result = true;
        }
        return result;
    }
    public int insertMember(MemberBean bean) {
        return memberDao.insertMember(bean);
    }
    public int checkIdPwd(String userid, String pwd) {
        int result=0;
        String dbPwd = memberDao.checkIdPwd(userid);
        if(dbPwd==null || dbPwd.isEmpty()){ //해당 아이디가 존재하지 않는 경우
            result = ID_NONE;
        }else{ //아이디는 존재하는 경우
            if(pwd.equals(dbPwd)){ //비밀번호도 일치
                result = LOGIN_OK;
            }else{ //비밀번호 불일치
                result = PWD_DISAGREE;
            }
        }
    }
    return result;
}
```



MemberServiceImpl

```
public int deleteMember(String userid){
    int n = memberDao.delMember(userid);
    return n;
}

public MemberBean selectMember(String userid){
    return memberDao.selectMember(userid);
}

public int updateMember(MemberBean bean){
    return memberDao.updateMember(bean);
}
}
```



MemberService

```
public interface MemberService {  
    //로그인 관련 상수 선언  
    public static final int LOGIN_OK=1;  
    public static final int ID_NONE=2;  
    public static final int PWD_DISAGREE=3;  
  
    public boolean checkUserid(String userid);  
    public int insertMember(MemberBean bean);  
    public int checkIdPwd(String userid, String pwd);  
  
    public int deleteMember(String userid);  
  
    public MemberBean selectMember(String userid);  
    public int updateMember(MemberBean bean);  
}
```


우편번호 찾기

http://localhost:9090/springherb3/member/zipcode.do - Internet Exp...

http://localhost:9090/springherb3/member/zipcode.do

우편번호 검색

찾고 싶으신 주소의 동(읍/면)을 입력하세요

지역명 :

우편번호	주소
151-705	서울 관악구 조원동 KT구로지사
152-879	서울 구로구 구로3동 820~861
152-880	서울 구로구 구로3동 1118~1252
152-880	서울 구로구 구로3동 1260
152-053	서울 구로구 구로3동
152-764	서울 구로구 구로4동 두산아파트
152-787	서울 구로구 구로4동 럭키아파트 (1~2동)
152-839	서울 구로구 구로4동 73~74
152-841	서울 구로구 구로4동 91~99
152-845	서울 구로구 구로4동 114~131
152-846	서울 구로구 구로4동 132~144
152-873	서울 구로구 구로4동 313
152-850	서울 구로구 구로4동 314~328
152-850	서울 구로구 구로4동 330~338

zipcode.jsp

```
<script type="text/javascript">
    function search(frm){
        if(!frm.dong.value){
            alert("주소(동)를 입력하세요");
            frm.dong.focus();
            return false;
        }
        return true;
    }
    function setZipcode(zipcode, address){    //120-456
        opener.frm1.zipcode.value = zipcode;
        opener.frm1.address.value = address;
        self.close();
    }
</script>    </head>
<body onload="frmZipcode.dong.focus()">
<div class="div1">
<h2>우편번호 검색</h2>
<p>찾고 싶으신 주소의 동(읍/면)을 입력하세요</p>
<form name="frmZipcode" method="post"
    action='<c:url value="/member/zipcode.do" />' onsubmit="return search(this)">
    <label for="dong">지역명 : </label>
    <input type="text" name="dong" id="dong" style="ime-mode:active" value="{param.dong }">
    <input type="submit" value="찾기">
</form>
<c:if test="{empty zipcodeList }">
    <p>해당하는 주소가 없습니다</p>
</c:if><br>
```

```
$(opener.document).find("#zipcode").val(zipcode);
$(opener.document).find("#address").val(address);
```

```

<c:if test="${!empty zipcodeList }">
  <table width="460" class="box2"
summary="우편번호 검색 결과에 관한 표로써, 우편번호, 주소에 대한 정보를 제공합니다.">
  <caption>우편번호 검색</caption>
  <colgroup>
    <col style="width:20%" /><col style="width:*" />
  </colgroup>
  <thead>
    <tr><th scope="col">우편번호</th><th scope="col">주소</th> </tr>
  </thead>
  <tbody>
    <c:forEach var="bean" items="${zipcodeList}">
      <c:set var="address" value="${bean.sido } ${bean.gugun } ${bean.dong }" />
      <!-- 반복문 시작 -->
      <tr>
        <td>${bean.zipcode}</td>
        <td>
          <a href="#" onclick="setZipcode('${bean.zipcode}', '${address}')">
            ${address} ${bean.bunji}
          </a>
        </td>
      </tr>
      <!-- 반복문 끝 -->
    </c:forEach>
  </tbody>
</table>
</c:if>
</div> </body> </html>

```

```
@Controller
public class ZipcodeController {
    @Autowired ZipcodeService zipcodeService;
```

```
    private static final Logger logger = LoggerFactory.getLogger(ZipcodeController.class);
```

```
    @RequestMapping("/member/zipcode.do")
```

```
    public String getZipcode(@RequestParam(required=false) String dong, Model model){
```

```
        //우편번호 조회
```

```
        //1. 파라미터
```

```
        logger.debug("파라미터 dong={}", dong);
```

```
        //2. db작업 - select
```

```
        List<ZipcodeBean> alist=null;
```

```
        if(dong!=null && !dong.isEmpty()){
```

```
            alist = zipcodeService.listZipcode(dong);
```

```
            logger.debug("우편번호 조회 결과: alist.size()={},", alist.size());
```

```
        }
```

```
        //3. 결과 저장, 리턴
```

```
        model.addAttribute("zipcodeList", alist);
```

```
        return "/member/zipcode";
```

```
    }
```

```
}
```

아이디 중복 확인

http://localhost:9090/springherb3/member/checkId.do

아이디 중복 검사

이미 등록된 아이디입니다. 다른 아이디를 입력하세요

http://localhost:9090/springherb3/member/checkId.do

아이디 중복 검사

사용가능한 아이디입니다. [사용하기]버튼을 눌러주세요

```

<script>
function send(frm){
    if(!frm.userid.value){
        alert("아이디를 입력하세요");
        frm.userid.focus();
        return false;
    }
    return true;
}
function setUserid(){
    opener.frm1.userid.value= frmId.userid.value;
    opener.frm1.chkId.value="Y"; //중복확인을 한 경우
    self.close();
    $("#btUse").click(function(){
        $(opener.document).find("#userid").val('${param.userid}');
        $(opener.document).find("#chkId").val('Y');
        $(opener.document).find("#userid").attr('readonly','readonly');
        self.close();
    });
}
</script> </head>
<body><div class="div1">
<h2>아이디 중복 검사</h2>
<form name="frmId" method="post"
    action='<c:url value="/member/checkId.do"/>' onsubmit="return send(this)">
    <label for="userid">아이디</label>
    <input type="text" name="userid" id="userid" value="${param.userid}">
    <input type="submit" value="아이디 확인" >
    <c:if test="${existId==false && param.userid!=null
        && !empty param.userid}">
        <input type="button" value="사용하기" onclick="setUserid()">
        <p>사용가능한 아이디입니다.    [사용하기]버튼을 눌러주세요</p>
    </c:if>
    <c:if test="${existId==true }">
        <p>이미 등록된 아이디입니다. 다른 아이디를 입력하세요</p>
    </c:if>
</form></div></body></html>

```

```
@RequestMapping("/checkId.do")
public String checkId(String userid, Model model){
    //아이디 중복 확인
    //1. 파라미터
    logger.debug("파라미터 : userid={}", userid);

    //2. db작업 - select
    boolean result =false;
    if(userid!=null && !userid.isEmpty()){
        result = memberService.checkUserid(userid);
        logger.debug("아이디 중복 확인 결과, result={}", result);
    }

    //3. 결과 저장, 리턴
    model.addAttribute("existId", result);
    return "/member/checkId";
}
```

로그인 화면

관리자로그인 | 로그인 | 회원가입 | 장바

Herb mall

베스트 자유게시판 이용후기 Q&A FAQ 이벤트&쿠폰

로그인

허브

허브차

아로마오일

허브향초

아이디

hong

비밀번호

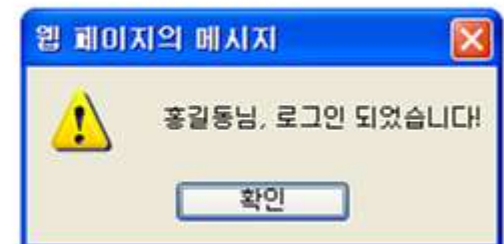
로그인

☒ 아이디 저장하기

회원가입

ID찾기

PWD찾기




```

<article class="simpleForm">
<form name="frmLogin" method="post"
    action='<c:url value="/login/login.do" />' onsubmit="return send()">
    <!-- 로그인후 이동할 url 관련 hidden 필드 -->
    <input type="hidden" name="loginUri" value="{param.loginUri }">
        <fieldset>
            <legend>로그인</legend>
            <div>
                <label for="userid" class="label">아이디</label>
                <input type="text" name="userid" id="userid"
                    value="{cookie.ck_userid.value }">
            </div>
            <div>
                <label for="pwd" class="label">비밀번호</label>
                <input type="password" name="pwd" id="pwd">
            </div>
            <div class="align_center">
                <input type="submit" id="lg_submit" value="로그인">
                <input type="checkbox" name="idSave" id="idSave"
                    <c:if test="{!empty cookie.ck_userid }">
                        checked
                    </c:if> >
                <label for="idSave">아이디 저장하기</label>
            </div>
            <div class="align_center">
                <input type="button" value="회원가입" >
                <input type="button" value="ID찾기" >
                <input type="button" value="PWD찾기" >
            </div>
        </fieldset>
    </form> </article>

```

```
@Controller
@RequestMapping("/login")
public class LoginController {
    @Autowired MemberService memberService;
    private static final Logger logger = LoggerFactory.getLogger(LoginController.class);

    @RequestMapping(value="/login.do", method=RequestMethod.GET)
    public String loginGet(){
        //로그인 화면 보여주기
        logger.debug("로그인 화면 보여주기 위한 핸들러");

        return "/login/login";
    }

    @RequestMapping(value="/login.do", method=RequestMethod.POST)
    public String loginPost(String userid, String pwd, String idSave, HttpServletRequest request,
        HttpServletResponse response, Model model){
        //로그인 처리
        //1. 파라미터
        logger.debug("파라미터 userid={}, pwd={}", userid , pwd);
        logger.debug("idSave={}", idSave);

        //2. db작업 - select
        int result=memberService.checkIdPwd(userid, pwd);
        logger.debug("아이디 비밀번호 체크 결과, result={},result);

        //3. 결과 저장, 리턴
        String msg="", url="";
        if(result==MemberService.LOGIN_OK){
            //세션에 아이디 저장
            HttpSession session = request.getSession();
            session.setAttribute("userid", userid);
```

LoginController

```
//쿠키에 아이디 저장
Cookie ck = new Cookie("ck_userid", userid);
ck.setPath("/");
if(idSave!=null){ //ID 저장하기를 체크한 경우
    ck.setMaxAge(1000*24*60*60); //1000일
    response.addCookie(ck);
}else{ //체크하지 않은 경우
    ck.setMaxAge(0); //쿠키 제거
    response.addCookie(ck);
}
msg=userid + "으로 로그인되었습니다";
url="../index.do";
}else if(result==MemberService.PWD_DISAGREE){
    msg="비밀번호가 일치하지 않습니다";
    url="login.do";
}else if(result==MemberService.ID_NONE){
    msg="해당 아이디가 존재하지 않습니다";
    url="login.do";
}else{
    msg="로그인 조회 실패";
    url="login.do";
}
model.addAttribute("msg", msg);
model.addAttribute("url", url);
return "/inc/message";
}
```

- `cookie.setPath("경로")`
- 경로 미 설정 : 요청 URL의 경로에 대해서만 쿠키 전달
- 경로 설정 : 설정한 경로 및 그 하위 경로에 대해서 쿠키 전달

```
@RequestMapping("/logout.do")
public String logout(HttpSession session){
    //로그아웃 처리
    session.invalidate();

    return "redirect:/index.do";
}
```

컨트롤러 메서드의 리턴 타입 : **String =>** 뷰 이름을 리턴함

회원정보 수정

Herb mall

(홍길동 님) 로그아웃 | 회원정보변경 | 회원탈퇴 | 장바구니 | 주문내역

베스트

자유게시판

이용후기

Q&A

FAQ

이벤트&쿠폰

회원 정보 수정

허브

허브차

아로마오일

허브향초

성명 홍길동

회원ID hong

비밀번호

비밀번호 확인

주소

135-911

우편번호 찾기

서울 강남구 역삼1동 그레이스타워

300

핸드폰

016 ▼

- 200

- 2000

이메일 주소

ho2

@

직접입력 ▼

hotmail.com

수정

회원 탈퇴

(홍길동 님) [로그아웃](#) | [회원정보변경](#) | [회원탈퇴](#) | [장바구니](#) | [주](#)

Herb mall

[베스트](#)

[자유게시판](#)

[이용후기](#)

[Q&A](#)

[FAQ](#)

[이벤트&쿠폰](#)

허브

회원 탈퇴

회원탈퇴하시겠습니까?

허브차

비밀번호

아로마오일

회원탈퇴

취소

허브향초

허브몰 대표:김허브 사업자정보 : 123-45-67890 통신판매업 신고업번호 : 서울 제1호-허브몰 [사업자정보확인]

주소 : 서울시 강남구 역삼동 123-4 ABC타워 3층 101호 TEL : 02-1234-5678 fax : 02-1234-5678

개인정보관리책임자 : 이허브 (admin@herbmall.com)

Copyright © 2015 HerbMall All Rights Reserved.

```
@RequestMapping(value="/memberEdit.do", method=RequestMethod.GET)
```

```
public String updateGet(HttpSession session, Model model){
```

```
    //수정 화면 보여주기 - userid에 해당하는 회원 정보 보여준다
```

```
    //1. 파라미터
```

```
    String userid = (String) session.getAttribute("userid");
```

```
    logger.debug("파라미터 userid={}", userid);
```

```
    //로그인 안된 경우 - 로그인 페이지로 이동시킨다
```

```
    if(userid==null || userid.isEmpty()){
```

```
        model.addAttribute("msg", "먼저 로그인하세요");
```

```
        model.addAttribute("url", "../login/login.do");
```

```
        return "/inc/message";
```

```
    }
```

```
    //2. db작업 - select
```

```
    MemberVO vo= memberService.selectMember(userid);
```

```
    logger.debug("회원 정보 조회 결과, vo={}",vo);
```

```
    //3. 결과 저장, 리턴
```

```
    model.addAttribute("vo", vo);
```

```
    return "/member/memberEdit";
```

```
}
```

```
@RequestMapping(value="/memberEdit.do", method=RequestMethod.POST)
```

```
public String updatePost(MemberVO vo, String email3, HttpSession session, Model model){
```

```
    //수정 처리
```

```
    //1. 파라미터
```

```
    String userid = (String) session.getAttribute("userid");
```

```
    logger.debug("파라미터 userid={}, vo={}", userid, vo);
```

```
    vo.setUserid(userid);
```

MemberController

LoginInterceptor에서
처리하도록 변경

```

//hp, email 널인 경우 처리
String hp1 = vo.getHp1();
String hp2 = vo.getHp2();
String hp3 = vo.getHp3();
if(hp2==null || hp2.isEmpty() || hp3==null || hp3.isEmpty()){
    hp1="";
    hp2="";
    hp3="";
}
vo.setHp1(hp1);
vo.setHp2(hp2);
vo.setHp3(hp3);

String email1 = vo.getEmail1();
String email2 = vo.getEmail2();
if(email1==null || email1.isEmpty()){
    email1= "";
    email2= "";
}else{
    if(email2.equals("etc")){
        email2 = email3;
    }
}
vo.setEmail1(email1);
vo.setEmail2(email2);

```

```
//2. db작업 - update
```

```
//비밀번호가 일치하는 경우에만 수정 처리
```

```
String msg="", url="memberEdit.do";
```

```
int result = memberService.checkIdPwd(userid, vo.getPwd());
```

```
if(result == MemberService.LOGIN_OK){
```

```
    int n = memberService.updateMember(vo);
```

```
    if(n>0){
```

```
        msg="회원 정보 수정 성공!!";
```

```
    }else{
```

```
        msg="회원 정보 수정 실패!!";
```

```
    }
```

```
    logger.debug("회원정보 수정 결과, n={}",n);
```

```
}else if(result == MemberService.PWD_DISAGREE){
```

```
    msg="비밀번호가 일치하지 않습니다";
```

```
}else{
```

```
    msg="비밀번호 체크 실패";
```

```
}
```

```
//3. 결과 저장, 리턴
```

```
model.addAttribute("msg", msg);
```

```
model.addAttribute("url", url);
```

```
return "/inc/message";
```

```
}
```



```

@RequestMapping(value="/memberOut.do", method=RequestMethod.GET)
public String outGet(){
    //회원탈퇴 페이지 보여주기
    logger.debug("회원탈퇴 페이지 보여주기");
    return "/member/memberOut";
}

@RequestMapping(value="/memberOut.do", method=RequestMethod.POST)
public String outPost(String pwd, HttpSession session, Model model){
    //회원탈퇴 처리
    //1. 파라미터
    String userid = (String) session.getAttribute("userid");
    logger.debug("파라미터 : userid={}, pwd={}", userid, pwd);

    //2. db작업 - delete
    int result= memberService.checkIdPwd(userid, pwd);
    logger.debug("비밀번호 체크 결과, result={}", result);
    String msg="",url="";
    if(result==MemberService.LOGIN_OK){
        int n = memberService.deleteMember(userid);
        if(n>0){
            msg="회원탈퇴처리 되었습니다.";
            url="../index.do";
            session.invalidate();
            Cookie ck = new Cookie("ck_userid", userid);
            ck.setMaxAge(0); //쿠키제거
            ck.setPath("/");
            response.addCookie(ck);
        }else{
            msg="회원 탈퇴 실패";
            url="memberOut.do";
        }
    }
}

```

```
}else if(result == MemberService.PWD_DISAGREE){  
    msg="비밀번호가 일치하지 않습니다";  
    url="memberOut.do";  
}  
else{  
    msg="비밀번호 체크 실패!";  
    url="memberOut.do";  
}
```

//3. 결과 저장, 리턴

```
model.addAttribute("msg", msg);  
model.addAttribute("url", url);
```

```
return "/inc/message";
```

```
}
```

```

<div class="divForm">
<form name="frm1" method="post" action="<c:url value="/member/memberEdit.do"/>"
    onsubmit="return send(this)">
<fieldset>
    <legend>회원 정보 수정</legend>
    <p><span class="sp1">성명</span> ${memberBean.name}</p>
    <p><span class="sp1">회원ID</span> ${sessionScope.userid}</p>
    <p>
        <label for="pwd">비밀번호</label> <input type="Password" name="pwd" id="pwd">
    </p>
    <p>
        <label for="pwd2">비밀번호 확인</label> <input type="Password" name="pwd2" id="pwd2">
    </p>
    <p>
        <label for="zipcode">주소</label>
        <input type="text" name="zipcode" id="zipcode" ReadOnly size="10"
            title="우편번호" value="${memberBean.zipcode} ">
        <input type="Button" value="우편번호 찾기"
            onclick="getZipcode()" title="새창열림"><br />
        <span class="sp1">&nbsp;</span> <input type="text" name="address" ReadOnly size="50"
            title="주소" value="${memberBean.address} "><br />
        <span class="sp1">&nbsp;</span> <input type="text" name="addressDetail" size="50"
            title="상세주소" value="${memberBean.addressDetail} ">
    </p>
    <p>
        <label for="hp1">핸드폰</label> &nbsp;<br>
        <select name="hp1" id="hp1" title="휴대폰 앞자리">
            <option value="010"
                <c:if test="${memberBean.hp1=='010'}">selected</c:if> >010</option>
            <option value="011"
                <c:if test="${memberBean.hp1=='011'}">selected</c:if> >011</option>
        </select>
    </p>

```

```

        <option value="016"
            <c:if test="\${memberBean.hp1=='016'}">selected</c:if> >016</option>
        <option value="017"
            <c:if test="\${memberBean.hp1=='017'}">selected</c:if> >017</option>
        <option value="018"
            <c:if test="\${memberBean.hp1=='018'}">selected</c:if> >018</option>
        <option value="019"
            <c:if test="\${memberBean.hp1=='019'}">selected</c:if> >019</option>
    </select> -
    <input type="text" name="hp2" size="10" maxlength="4" title="휴대폰 가운데자리"
        value="\${memberBean.hp2}"/>-
    <input type="text" name="hp3" size="10" maxlength="4" title="휴대폰 뒷자리"
        value="\${memberBean.hp3}"/>
</p>
<p>
    <c:set var="etcYn" value="" />
    <c:choose>
        <c:when test="\${memberBean.email2=='naver.com' ||
            memberBean.email2=='hanmail.net' ||
            memberBean.email2=='nate.com' ||
            memberBean.email2=='gmail.com' ||
            empty memberBean.email2}"/>
            <c:set var="etcYn" value="N" />
        </c:when>
        <c:otherwise>
            <c:set var="etcYn" value="Y" />
        </c:otherwise>
    </c:choose>
    <label for="email1">이메일 주소</label>
    <input type="text" name="email1" size="15" id="email1" title="이메일주소 앞자리"
        value="\${memberBean.email1}"/>@

```

```

<select name="email2" onchange="emailCheck()" title="이메일주소 뒷자리">
  <option value="naver.com"
    <c:if test="\${memberBean.email2=='naver.com'}">selected</c:if>    >naver.com</option>
  <option value="hanmail.net"
    <c:if test="\${memberBean.email2=='hanmail.net'}">selected</c:if>    >hanmail.net</option>
  <option value="nate.com"
    <c:if test="\${memberBean.email2=='nate.com'}">selected</c:if>    >nate.com</option>
  <option value="gmail.com"
    <c:if test="\${memberBean.email2=='gmail.com'}">selected</c:if>    >gmail.com</option>
  <option value="etc"
    <c:if test="\${etcYn=='Y'}">selected</c:if>    >직접입력</option>
</select>
<c:if test="\${etcYn=='Y'}">
  <input type="text" name="email3" size="20" title="직접입력인 경우 이메일주소 뒷자리"
    style="visibility:visible" value="\${memberBean.email2}">
</c:if>
<c:if test="\${etcYn!='Y'}">
  <input type="text" name="email3" size="20" title="직접입력인 경우 이메일주소 뒷자리"
    style="visibility:hidden">
</c:if>
</p>
<p class="center">
  <input type="submit" value="수정">
</p>
</fieldset>
</form>
</div>

```

```
<form name="frm1" method="post" action='<c:url value="/member/memberOut.do"/>'
onsubmit="return send(this)">
  <fieldset>
    <legend>회원 탈퇴</legend>
    <p>회원탈퇴하시겠습니까?</p>
    <p><label for="pwd">비밀번호</label>
      <input type="password" name="pwd" id="pwd"></p>
    <p class="center">
      <input type="submit" value="회원탈퇴">
      <input type="reset" value="취소">
    </p>
  </fieldset>
</form>
```



HandlerInterceptor를 통한 요청 가로채기



HandlerInterceptor를 통한 요청 가로채기

- 스프링이 기본적으로 제공하는 HandlerMapping은 HandlerInterceptor를 이용해서 컨트롤러가 요청을 처리하기 전과 처리한 후에 알맞은 기능을 수행할 수 있도록 하고 있음
- 조건에 따라 컨트롤러에 요청을 전달하지 않고 싶거나, 컨트롤러가 요청을 처리한 후에 ModelAndView 객체를 조작하고 싶은 경우에 HandlerInterceptor를 사용하면 됨



HandlerInterceptor를 통한 요청 가로채기

■ HandlerInterceptor 인터페이스의 메서드

```
public boolean preHandle(HttpServletRequest request, HttpServletResponse response,  
    Object handler)  
public void postHandle(HttpServletRequest request, HttpServletResponse response,  
    Object handler, ModelAndView modelAndView)  
public void afterCompletion(HttpServletRequest request, HttpServletResponse response,  
    Object handler, Exception ex)
```

■ preHandle() 메서드

- 클라이언트의 요청을 컨트롤러에 전달하기 전에 호출됨
- 세번째 파라미터인 handler 에는 컨트롤러 객체가 전달됨
- 한 개 이상의 HandlerInterceptor 가 체인을 형성하게 되는데, preHandle() 메서드가 **false**를 리턴하면 체인의 다음 HandlerInterceptor 또는 컨트롤러를 실행하지 않고 요청 처리를 종료하게 됨

■ postHandle() 메서드

- 컨트롤러가 요청을 처리한 뒤에 호출됨
- preHandle() 메서드의 실행 순서와 반대로 실행됨
- 컨트롤러 실행 도중에 예외가 발생할 경우 postHandle() 메서드는 실행되지 않음



HandlerInterceptor를 통한 요청 가로채기

- afterCompletion() 메서드
 - 클라이언트의 요청을 처리한 뒤, 즉 뷰를 통해서 클라이언트에 응답을 전송한 뒤에 실행됨
 - 컨트롤러가 처리하는 도중이나 뷰를 생성하는 과정에 예외가 발생하더라도 afterCompletion() 메서드는 실행됨
 - preHandle() 메서드의 실행 순서와 반대로 실행됨
- HandlerInterceptor 인터페이스를 직접 구현하여 클래스를 작성하는 경우에는 사용하지 않는 메서드도 구현해주어야 함
- HandlerInterceptorAdapter 클래스를 상속받으면 필요한 메서드만 오버라이딩해서 구현하면 됨



예

@Component

```
public class CommonHandlerInterceptor extends HandlerInterceptorAdapter {  
    @Autowired CategoryService categoryService;  
  
    private static final Logger logger= LoggerFactory.getLogger(CommonHandlerInterceptor.class);  
  
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response,  
        Object arg2) throws Exception {  
        // 컨트롤러 수행이전에 먼저 수행되는 메서드  
        //클라이언트의 요청을 컨트롤러에 전달하기 전에 호출됨  
        //모든 페이지에서 필요한 카테고리 조회하기 - 왼쪽 메뉴  
        //1. 파라미터  
        String requestURI = request.getRequestURI();  
        logger.debug("preHandle() 메서드 호출, requestURI={}", requestURI);  
    }  
}
```



예

```
if (requestURI.indexOf("/admin") != -1) { //체크 예외페이지  
    return true;  
}
```

```
//2. db작업 - select
```

```
List<CategoryBean> alist
```

```
    = (ArrayList<CategoryBean>) categoryService.listCategoryAll();
```

```
logger.debug("카테고리 목록 조회 성공, alist.size()={}, alist.size());
```

```
//3. 결과 저장
```

```
request.setAttribute("cgList", alist);
```

```
return true;    //true => 다음 컨트롤러가 수행됨
```

```
}
```

```
}
```



HandlerMapping에 HandlerInterceptor 설정하기

- HandlerInterceptor 를 구현한 뒤에는 HandlerMapping의 interceptors 프로퍼티를 사용해서 HandlerInterceptor를 등록해주면 됨

```
<mvc:interceptors>
    <!-- <bean class="com.hy.herb.controller.CommonHandlerInterceptor" /> -->
    <ref bean ="commonHandlerInterceptor" />

    <mvc:interceptor>
        <mvc:mapping path="/shop/cart/*" />
        <mvc:mapping path="/shop/order/*" />
        <mvc:mapping path="/admin/*" />
        <mvc:mapping path="/member/memberEdit.do" />
        <mvc:mapping path="/member/memberOut.do" />
        <!-- <mvc:exclude-mapping path="/info/**"/> -->
        <!-- <bean class="com.hy.herb.controller.LoginInterceptor" /> -->
        <ref bean ="loginInterceptor" />
    </mvc:interceptor>
</mvc:interceptors>
```



HandlerInterceptor의 실행 순서

- HandlerMapping 에는 한 개 이상의 HandlerInterceptor 를 등록할 수 있는데, 이 경우 다음과 같은 순서로 HandlerInterceptor가 실행됨
 - [1] 컨트롤러 실행 전 : 등록된 순서대로 preHandle() 메서드 실행
 - [2] 컨트롤러 실행 후 : 등록된 순서의 반대로 postHandle() 메서드 실행
 - [3] 처리 완료 후 (뷰 생성 후) : 등록된 순서의 반대로 afterCompletion() 메서드 실행



HandlerInterceptor의 실행 순서

- 예) 다음과 같이 3개의 HandlerInterceptor를 등록했다면.

```
<mvc:interceptors>  
    <ref bean ="interceptor1" />  
    <ref bean ="interceptor2" />  
    <ref bean ="interceptor3" />  
</mvc:interceptors>
```



HandlerInterceptor의 실행 순서

- 클라이언트의 요청을 처리할 때 실행되는 메서드 순서
 - [1] preHandle() 메서드 실행
 - (a) interceptor1.preHandle()
 - (b) interceptor2.preHandle()
 - (c) interceptor3.preHandle()
 - [2] 컨트롤러의 handleRequest() 메서드 실행
 - [3] postHandle() 메서드 실행
 - (a) interceptor3.postHandle()
 - (b) interceptor2.postHandle()
 - (c) interceptor1.postHandle()
 - [4] 뷰 객체의 render() 메서드 실행해서 응답 결과 생성
 - [5] afterCompletion() 메서드 실행
 - (a) interceptor3.afterCompletion()
 - (b) interceptor2.afterCompletion()
 - (c) interceptor1.afterCompletion()



servlet-context.xml

```
<interceptors>
```

```
  <beans:ref bean ="commonHandlerInterceptor" />
```

```
  <!-- <beans:bean class="com.hy.herb.controller.CommonHandlerInterceptor" /> -->
```

```
  <interceptor>
```

```
    <mapping path="/shop/cart/*" />
```

```
    <mapping path="/shop/order/*" />
```

```
    <mapping path="/admin/*" />
```

```
    <mapping path="/member/memberEdit.do" />
```

```
    <mapping path="/member/memberOut.do" />
```

```
    <!-- <exclude-mapping path="/info/**"/>    -->
```

```
    <beans:ref bean ="loginInterceptor" />
```

```
    <!--<beans:bean class="com.hy.herb.controller.LoginInterceptor" /> -->
```

```
  </interceptor>
```

```
</interceptors>
```



servlet-context.xml

```
<mvc:interceptors>
  <!-- <bean class="com.hy.herb.controller.CommonHandlerInterceptor" /> -->
  <ref bean ="commonHandlerInterceptor" />

  <mvc:interceptor>
    <mvc:mapping path="/shop/cart/*" />
    <mvc:mapping path="/shop/order/*" />
    <mvc:mapping path="/admin/*" />
    <mvc:mapping path="/member/memberEdit.do" />
    <mvc:mapping path="/member/memberOut.do" />
    <!-- <mvc:exclude-mapping path="/info/**"/> -->
    <!-- <bean class="com.hy.herb.controller.LoginInterceptor" /> -->
    <ref bean ="loginInterceptor" />
  </mvc:interceptor>
</mvc:interceptors>
```



LoginInterceptor

```
package com.hy.herb.controller;
```

```
@Component
```

```
public class LoginInterceptor extends HandlerInterceptorAdapter{
```

```
    private static final Logger logger = LoggerFactory.getLogger(LoginInterceptor.class);
```

```
    @Override
```

```
    public void afterCompletion(HttpServletRequest request,  
                               HttpServletResponse response, Object handler, Exception ex)  
        throws Exception {
```

```
        logger.debug("afterCompletion() 호출");
```

```
        super.afterCompletion(request, response, handler, ex);
```

```
    }
```

```
    @Override
```

```
    public void postHandle(HttpServletRequest request,  
                           HttpServletResponse response, Object handler,  
                           ModelAndView modelAndView) throws Exception {
```

```
        logger.debug("postHandle() 호출");
```

```
        super.postHandle(request, response, handler, modelAndView);
```

```
    }
```

@Override

```

public boolean preHandle(HttpServletRequest request,
    HttpServletResponse response, Object handler) throws Exception {
    String userid = (String) request.getSession().getAttribute("userid");
    logger.debug("preHandle()호출, 세션 userid={}",userid);

    //로그인 안된 경우 처리
    if(userid==null || userid.isEmpty()){
        request.setAttribute("msg", "먼저 로그인하세요");
        request.setAttribute("url", "/login/login.do");

        RequestDispatcher dispatcher
        = request.getRequestDispatcher("/WEB-INF/views/common/message.jsp");
        dispatcher.forward(request, response);

        return false;
    }else{
        return true; //true => 다음 컨트롤러가 수행됨
    }
}
}
class

```

```

public boolean preHandle(HttpServletRequest request, HttpServletResponse response,
                        Object arg2) throws Exception {
    String reqUri=request.getRequestURI();
    //=> /springherb/admin/login/adminLogin.do

    logger.info("preHandle() 호출, reqUri={}", reqUri);
    //컨트롤러가 수행되기 전에 먼저 호출되는 메서드

    //제외할 페이지
    if(reqUri.indexOf("/admin/login/adminLogin.do")!=-1){
        return true;
    }

    HttpSession session = request.getSession();
    String userid =(String)session.getAttribute("userid");

    if(userid==null || userid.isEmpty()){
        String url="";
        if(reqUri.indexOf("/admin")==-1){
            //사용자 페이지인 경우 -사용자 로그인
            url="/login/login.do";
        }else{
            //관리자 페이지인 경우-관리자 로그인
            url="/admin/login/adminLogin.do";
        }
    }
}

```

```
response.setContentType("text/html;charset=utf-8");  
PrintWriter out = response.getWriter();  
out.println("<script>alert('먼저 로그인하세요');");  
out.println("location.href='"+ request.getContextPath()  
        +url+ "';</script>");
```

```
return false; //다음 컨트롤러를 수행하지 않음
```

```
}else{
```

```
return true; //다음 컨트롤러를 정상적으로 수행
```

```
}
```

```
}
```



servlet-context.xml

```
<interceptors>
  <interceptor>
    <mapping path="/shop/cart/*" />
    <mapping path="/shop/order/*" />
    <mapping path="/admin/*" />
    <mapping path="/admin/*/*" />
    <mapping path="/member/memberEdit.do" />
    <mapping path="/member/memberOut.do" />
    <!-- <exclude-mapping path="/info/**"/>    -->
    <beans:ref bean ="loginInterceptor" />
    <!--<beans:bean class="com.hy.herb.controller.LoginInterceptor" /> -->
  </interceptor>
</interceptors>
```