```
package com.myspring.pro27.member.controller;

import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.EnableAspectJAutc
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.support.RedirectAttr

import com.myspring.pro27.member.service.MemberService;
import com.myspring.pro27.member.vo.MemberVO;



@Controller("memberController")
//@EnableAspectJAutoProxy
public class MemberControllerImpl   implements MemberController
    private static final Logger logger = LoggerFactory.getLogge
    @Autowired
    private MemberService memberService;
    @Autowired
    private MemberVO memberVO ;

    @Override
    @RequestMapping(value="/member/listMembers.do" ,method = Re
    public ModelAndView listMembers(HttpServletRequest request,
        String viewName = getViewName(request);
//      String viewName = (String)request.getAttribute("viewNam
        //System.out.println("viewName: " +viewName);
        logger.info("viewName: "+ viewName);
```

```java
        logger.debug("viewName: "+ viewName);
        List membersList = memberService.listMembers();
        ModelAndView mav = new ModelAndView(viewName);
        mav.addObject("membersList", membersList);
        return mav;
    }

    @Override
    @RequestMapping(value="/member/addMember.do" ,method = Requ
    public ModelAndView addMember(@ModelAttribute("member") Mem
                              HttpServletRequest request, HttpS
        request.setCharacterEncoding("utf-8");
        int result = 0;
        result = memberService.addMember(member);
        ModelAndView mav = new ModelAndView("redirect:/member/l
        return mav;
    }

    @Override
    @RequestMapping(value="/member/removeMember.do" ,method = R
    public ModelAndView removeMember(@RequestParam("id") String
                       HttpServletRequest request, HttpServletR
        request.setCharacterEncoding("utf-8");
        memberService.removeMember(id);
        ModelAndView mav = new ModelAndView("redirect:/member/l
        return mav;
    }

    @RequestMapping(value = { "/member/loginForm.do", "/member/
//  @RequestMapping(value = "/member/*Form.do", method =  Reque
    public ModelAndView form(HttpServletRequest request, HttpSe
        String viewName = getViewName(request);
        ModelAndView mav = new ModelAndView();
        mav.setViewName(viewName);
        return mav;
    }

    @Override
    @RequestMapping(value = "/member/login.do", method = Reques
    public ModelAndView login(@ModelAttribute("member") MemberV
                              RedirectAttributes rAttr,
                               HttpServletRequest request, Http
    ModelAndView mav = new ModelAndView();
    memberVO = memberService.login(member);
    if(memberVO != null) {
            HttpSession session = request.getSession();
            session.setAttribute("member", memberVO);
            session.setAttribute("isLogOn", true);
            mav.setViewName("redirect:/member/listMembers.do");
    }else {
            rAttr.addAttribute("result","loginFailed");
            mav.setViewName("redirect:/member/loginForm.do");
    }
    return mav;
    }

    @Override
    @RequestMapping(value = "/member/logout.do", method =  Requ
    public ModelAndView logout(HttpServletRequest request, Http
        HttpSession session = request.getSession();
        session.removeAttribute("member");
        session.removeAttribute("isLogOn");
        ModelAndView mav = new ModelAndView();
        mav.setViewName("redirect:/member/listMembers.do");
        return mav;
    }

    @RequestMapping(value = "/member/*Form.do", method =  Reque
    private ModelAndView form(@RequestParam(value= "result", re
                               HttpServletRequest request,
                               HttpServletResponse response) th
        //String viewName = getViewName(request);
        String viewName = (String)request.getAttribute("viewNam
        ModelAndView mav = new ModelAndView();
        mav.addObject("result",result);
        mav.setViewName(viewName);
        return mav;
    }


    private String getViewName(HttpServletRequest request) thro
        String contextPath = request.getContextPath();
        String uri = (String) request.getAttribute("javax.servl
        if (uri == null || uri.trim().equals("")) {
            uri = request.getRequestURI();
        }

        int begin = 0;
        if (!((contextPath == null) || ("".equals(contextPath))
            begin = contextPath.length();
        }

        int end;
        if (uri.indexOf(";") != -1) {
            end = uri.indexOf(";");
        } else if (uri.indexOf("?") != -1) {
            end = uri.indexOf("?");
        } else {
            end = uri.length();
        }

        String viewName = uri.substring(begin, end);
        if (viewName.indexOf(".") != -1) {
            viewName = viewName.substring(0, viewName.lastIndex
        }
        if (viewName.lastIndexOf("/") != -1) {
            viewName = viewName.substring(viewName.lastIndexOf(
        }
        return viewName;
    }


}

```