

```
1 package com.myspring.pro30.member.controller;
2
3 import java.util.List;
4
5 import javax.servlet.http.HttpServletRequest;
6 import javax.servlet.http.HttpServletResponse;
7 import javax.servlet.http.HttpSession;
8
9 import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.context.annotation.EnableAspectJAutoProxy;
11 import org.springframework.stereotype.Controller;
12 import org.springframework.web.bind.annotation.ModelAttribute;
13 import org.springframework.web.bind.annotation.RequestMapping;
14 import org.springframework.web.bind.annotation.RequestMethod;
15 import org.springframework.web.bind.annotation.RequestParam;
16 import org.springframework.web.servlet.ModelAndView;
17 import org.springframework.web.servlet.mvc.support.RedirectAttr
18
19 import com.myspring.pro30.member.service.MemberService;
20 import com.myspring.pro30.member.vo.MemberVO;
21
22
23
24
25 @Controller("memberController")
26 //@EnableAspectJAutoProxy
27 public class MemberControllerImpl implements MemberController
28     @Autowired
29     private MemberService memberService;
30     @Autowired
31     private MemberVO memberVO ;
32
33     @RequestMapping(value = { "/", "/main.do"}, method = Request
34     private ModelAndView main(HttpServletRequest request, HttpS
35         String viewName = (String)request.getAttribute("viewName");
36         ModelAndView mav = new ModelAndView();
37         mav.setViewName(viewName);
38         return mav;
39     }
40
41     @Override
```

```
42     @RequestMapping(value="/member/listMembers.do" ,method = Re
43     public ModelAndView listMembers(HttpServletRequest request,
44         request.setCharacterEncoding("utf-8");
45         response.setContentType("html/text; charset=utf-8");
46         String viewName = (String)request.getAttribute("viewName");
47         List membersList = memberService.listMembers();
48         ModelAndView mav = new ModelAndView(viewName);
49         mav.addObject("membersList", membersList);
50         return mav;
51     }
52
53     @Override
54     @RequestMapping(value="/member/addMember.do" ,method = Requ
55     public ModelAndView addMember(@ModelAttribute("member") Mem
56         HttpServletRequest request, HttpS
57         request.setCharacterEncoding("utf-8");
58         response.setContentType("html/text; charset=utf-8");
59         int result = 0;
60         result = memberService.addMember(member);
61         ModelAndView mav = new ModelAndView("redirect:/member/l
62         return mav;
63     }
64
65     @Override
66     @RequestMapping(value="/member/removeMember.do" ,method = R
67     public ModelAndView removeMember(@RequestParam("id") String
68         HttpServletRequest request, HttpServletRequestR
69         request.setCharacterEncoding("utf-8");
70         memberService.removeMember(id);
71         ModelAndView mav = new ModelAndView("redirect:/member/l
72         return mav;
73     }
74     /*
75     @RequestMapping(value = { "/member/loginForm.do", "/member/
76     @RequestMapping(value = "/member/*Form.do", method = Reque
77     public ModelAndView form(HttpServletRequest request, HttpSe
78         String viewName = getViewName(request);
79         ModelAndView mav = new ModelAndView();
80         mav.setViewName(viewName);
81         return mav;
82     }
```

```
83      */
84
85      @Override
86      @RequestMapping(value = "/member/login.do", method = Request
87      public ModelAndView login(@ModelAttribute("member") MemberV
88                              RedirectAttributes rAttr,
89                              HttpServletRequest request, Http
90      ModelAndView mav = new ModelAndView();
91      memberVO = memberService.login(member);
92      if(memberVO != null) {
93          HttpSession session = request.getSession();
94          session.setAttribute("member", memberVO);
95          session.setAttribute("isLogOn", true);
96          //mav.setViewName("redirect:/member/listMembers.do");
97          String action = (String)session.getAttribute("action");
98          session.removeAttribute("action");
99          if(action != null) {
100              mav.setViewName("redirect:"+action);
101          }else {
102              mav.setViewName("redirect:/member/listMembers.do");
103          }
104
105      }else {
106          rAttr.addAttribute("result", "loginFailed");
107          mav.setViewName("redirect:/member/loginForm.do");
108      }
109      return mav;
110  }
111
112      @Override
113      @RequestMapping(value = "/member/logout.do", method = Requ
114      public ModelAndView logout(HttpServletRequest request, Http
115          HttpSession session = request.getSession();
116          session.removeAttribute("member");
117          session.removeAttribute("isLogOn");
118          ModelAndView mav = new ModelAndView();
119          mav.setViewName("redirect:/member/listMembers.do");
120          return mav;
121      }
122
123      @RequestMapping(value = "/member/*Form.do", method = Reque
```

```
124     private ModelAndView form(@RequestParam(value= "result", re
125                               @RequestParam(value= "action", re
126                               HttpServletRequest request,
127                               HttpServletResponse response) th
128     String viewName = (String)request.getAttribute("viewName");
129     HttpSession session = request.getSession();
130     session.setAttribute("action", action);
131     ModelAndView mav = new ModelAndView();
132     mav.addObject("result",result);
133     mav.setViewName(viewName);
134     return mav;
135 }
136
137
138 private String getViewName(HttpServletRequest request) throws
139     String contextPath = request.getContextPath();
140     String uri = (String) request.getAttribute("javax.servlet
141     if (uri == null || uri.trim().equals("")) {
142         uri = request.getRequestURI();
143     }
144
145     int begin = 0;
146     if (!((contextPath == null) || ("".equals(contextPath)))
147         begin = contextPath.length();
148     }
149
150     int end;
151     if (uri.indexOf(";") != -1) {
152         end = uri.indexOf(";");
153     } else if (uri.indexOf("?") != -1) {
154         end = uri.indexOf("?");
155     } else {
156         end = uri.length();
157     }
158
159     String viewName = uri.substring(begin, end);
160     if (viewName.indexOf(".") != -1) {
161         viewName = viewName.substring(0, viewName.lastIndexOf(
162     }
163     if (viewName.lastIndexOf("/") != -1) {
164         viewName = viewName.substring(viewName.lastIndexOf(
```

```
165         }  
166         return viewName;  
167     }  
168  
169  
170 }  
171
```