

```
1 package com.myspring.pro30.board.controller;
2
3 import java.io.File;
4 import java.util.ArrayList;
5 import java.util.Enumeration;
6 import java.util.HashMap;
7 import java.util.Iterator;
8 import java.util.List;
9 import java.util.Map;
10
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import javax.servlet.http.HttpSession;
14
15 import org.apache.commons.io.FileUtils;
16 import org.springframework.beans.factory.annotation.Autowired;
17 import org.springframework.context.annotation.EnableAspectJAuto
18 import org.springframework.http.HttpHeaders;
19 import org.springframework.http.HttpStatus;
20 import org.springframework.http.ResponseEntity;
21 import org.springframework.stereotype.Controller;
22 import org.springframework.web.bind.annotation.RequestMapping;
23 import org.springframework.web.bind.annotation.RequestMethod;
24 import org.springframework.web.bind.annotation.RequestParam;
25 import org.springframework.web.bind.annotation.ResponseBody;
26 import org.springframework.web.multipart.MultipartFile;
27 import org.springframework.web.multipart.MultipartHttpServletRequest;
28 import org.springframework.web.servlet.ModelAndView;
29
30 import com.myspring.pro30.board.service.BoardService;
31 import com.myspring.pro30.board.vo.ArticleVO;
32 import com.myspring.pro30.board.vo.ImageVO;
33 import com.myspring.pro30.member.vo.MemberVO;
34
35
36 @Controller("boardController")
37 public class BoardControllerImpl implements BoardController{
38     private static final String ARTICLE_IMAGE_REPO = "C:\\board
39     @Autowired
40     private BoardService boardService;
41     @Autowired
```

```
42     private ArticleVO articleVO;
43
44     @Override
45     @RequestMapping(value= "/board/listArticles.do", method = {
46     public ModelAndView listArticles(HttpServletRequest request
47         String viewName = (String)request.getAttribute("viewName");
48         List<ArticleVO> articlesList = boardService.listArticles();
49         ModelAndView mav = new ModelAndView(viewName);
50         mav.addObject("articlesList", articlesList);
51         return mav;
52
53     }
54
55     //한 개 이미지 글쓰기
56     @Override
57     @RequestMapping(value="/board/addNewArticle.do" ,method = RequestMethod.POST)
58     @ResponseBody
59     public ResponseEntity addNewArticle(MultipartHttpServletRequest request,
60     HttpServletResponse response) throws Exception {
61         multipartRequest.setCharacterEncoding("utf-8");
62         Map<String, Object> articleMap = new HashMap<String, Object>();
63         Enumeration<String> enu=multipartRequest.getParameterNames();
64         while(enu.hasMoreElements()){
65             String name=(String)enu.nextElement();
66             String value=multipartRequest.getParameter(name);
67             articleMap.put(name,value);
68         }
69
70         String imageFileName= upload(multipartRequest);
71         HttpSession session = multipartRequest.getSession();
72         MemberVO memberVO = (MemberVO) session.getAttribute("memberVO");
73         String id = memberVO.getId();
74         articleMap.put("parentNO", 0);
75         articleMap.put("id", id);
76         articleMap.put("imageFileName", imageFileName);
77
78         String message;
79         ResponseEntity resEnt=null;
80         HttpHeaders responseHeaders = new HttpHeaders();
81         responseHeaders.add("Content-Type", "text/html; charset=utf-8");
82         try {
```

```
83         int articleNO = boardService.addNewArticle(articleM
84         if(imageFileName!=null && imageFileName.length()!=0
85             File srcFile = new
86             File(ARTICLE_IMAGE_REPO+ "\\\" + \"temp\"+ \"\\\" +
87             File destDir = new File(ARTICLE_IMAGE_REPO+\"\\\"
88             FileUtils.moveFileToDirectory(srcFile, destDir,
89         }
90
91         message = "<script>";
92         message += " alert('새글을 추가했습니다.');"
93         message += " location.href='"+multipartRequest.getCo
94         message += "</script>";
95         resEnt = new ResponseEntity(message, responseHeader
96     }catch(Exception e) {
97         File srcFile = new File(ARTICLE_IMAGE_REPO+"\\\"+\"te
98         srcFile.delete();
99
100         message = " <script>";
101         message += " alert('오류가 발생했습니다. 다시 시도해 주세요
102         message += " location.href='"+multipartRequest.getCo
103         message += "</script>";
104         resEnt = new ResponseEntity(message, responseHeader
105         e.printStackTrace();
106     }
107     return resEnt;
108 }
109
110
111 //한개의 이미지 보여주기
112 @RequestMapping(value="/board/viewArticle.do" ,method = Req
113 public ModelAndView viewArticle(@RequestParam("articleNO")
114                                 HttpServletRequest request,
115     String viewName = (String)request.getAttribute("viewName");
116     articleVO=boardService.viewArticle(articleNO);
117     ModelAndView mav = new ModelAndView();
118     mav.setViewName(viewName);
119     mav.addObject("article", articleVO);
120     return mav;
121 }
122
123 /*
```

```
124     //다중 이미지 보여주기
125     @RequestMapping(value="/board/viewArticle.do" ,method = Req
126     public ModelAndView viewArticle(@RequestParam("articleNO")
127         HttpServletRequest request, HttpServletResponse r
128         String viewName = (String)request.getAttribute("viewName");
129         Map articleMap=boardService.viewArticle(articleNO);
130         ModelAndView mav = new ModelAndView();
131         mav.setViewName(viewName);
132         mav.addObject("articleMap", articleMap);
133         return mav;
134     }
135     */
136
137
138
139     //한 개 이미지 수정 기능
140     @RequestMapping(value="/board/modArticle.do" ,method = Request
141     @ResponseBody
142     public ResponseEntity modArticle(MultipartHttpServletRequest
143         HttpServletResponse response) throws Exception{
144         multipartRequest.setCharacterEncoding("utf-8");
145         Map<String,Object> articleMap = new HashMap<String, Object>
146         Enumeration enu=multipartRequest.getParameterNames();
147         while(enu.hasMoreElements()){
148             String name=(String)enu.nextElement();
149             String value=multipartRequest.getParameter(name);
150             articleMap.put(name,value);
151         }
152
153         String imageFileName= upload(multipartRequest);
154         articleMap.put("imageFileName", imageFileName);
155
156         String articleNO=(String)articleMap.get("articleNO");
157         String message;
158         ResponseEntity resEnt=null;
159         HttpHeaders responseHeaders = new HttpHeaders();
160         responseHeaders.add("Content-Type", "text/html; charset=utf
161         try {
162             boardService.modArticle(articleMap);
163             if(imageFileName!=null && imageFileName.length()!=0) {
164                 File srcFile = new File(ARTICLE_IMAGE_REPO+"\\temp"
```

```
165         File destDir = new File(ARTICLE_IMAGE_REPO+"\\")+article
166         FileUtils.moveFileToDirectory(srcFile, destDir, true);
167
168         String originalFileName = (String)articleMap.get("orig
169         File oldFile = new File(ARTICLE_IMAGE_REPO+"\\")+article
170         oldFile.delete();
171     }
172     message = "<script>";
173     message += " alert('글을 수정했습니다.');"";
174     message += " location.href='"+multipartRequest.getContext
175     message += "</script>";
176     resEnt = new ResponseEntity(message, responseHeaders, Ht
177 }catch(Exception e) {
178     File srcFile = new File(ARTICLE_IMAGE_REPO+"\\")+temp+"\
179     srcFile.delete();
180     message = "<script>";
181     message += " alert('오류가 발생했습니다.다시 수정해주세요');"";
182     message += " location.href='"+multipartRequest.getContext
183     message += "</script>";
184     resEnt = new ResponseEntity(message, responseHeaders, Htt
185 }
186 return resEnt;
187 }
188
189 @Override
190 @RequestMapping(value="/board/removeArticle.do" ,method = Req
191 @ResponseBody
192 public ResponseEntity removeArticle(@RequestParam("articleNO
193         HttpServletRequest request, HttpS
194     response.setContentType("text/html; charset=UTF-8");
195     String message;
196     ResponseEntity resEnt=null;
197     HttpHeaders responseHeaders = new HttpHeaders();
198     responseHeaders.add("Content-Type", "text/html; charset=utf
199     try {
200         boardService.removeArticle(articleNO);
201         File destDir = new File(ARTICLE_IMAGE_REPO+"\\")+article
202         FileUtils.deleteDirectory(destDir);
203
204         message = "<script>";
205         message += " alert('글을 삭제했습니다.');"";
```

```
206         message += " location.href='"+request.getContextPath()+
207         message += "</script>";
208         resEnt = new ResponseEntity(message, responseHeaders, H
209
210     }catch(Exception e) {
211         message = "<script>";
212         message += " alert('작업중 오류가 발생했습니다.다시 시도해 주세
213         message += " location.href='"+request.getContextPath()+
214         message += "</script>";
215         resEnt = new ResponseEntity(message, responseHeaders, H
216         e.printStackTrace();
217     }
218     return resEnt;
219 }
220
221 /*
222 //다중 이미지 글 추가하기
223 @Override
224 @RequestMapping(value="/board/addNewArticle.do" ,method = Req
225 @ResponseBody
226 public ResponseEntity addNewArticle(MultipartHttpServletRequest
227     multipartRequest.setCharacterEncoding("utf-8");
228     String imageFileName=null;
229
230     Map articleMap = new HashMap();
231     Enumeration enu=multipartRequest.getParameterNames();
232     while(enu.hasMoreElements()){
233         String name=(String)enu.nextElement();
234         String value=multipartRequest.getParameter(name);
235         articleMap.put(name,value);
236     }
237
238     //로그인 시 세션에 저장된 회원 정보에서 글쓴이 아이디를 얻어와서 Map에
239     HttpSession session = multipartRequest.getSession();
240     MemberVO memberVO = (MemberVO) session.getAttribute("member
241     String id = memberVO.getId();
242     articleMap.put("id",id);
243     articleMap.put("parentNO", 0);
244
245     List<String> fileList =upload(multipartRequest);
246     List<ImageVO> imageFileList = new ArrayList<ImageVO>();
```

```
247     if(fileList!= null && fileList.size()!=0) {
248         for(String fileName : fileList) {
249             ImageVO imageVO = new ImageVO();
250             imageVO.setImageFileName(fileName);
251             imageFileList.add(imageVO);
252         }
253         articleMap.put("imageFileList", imageFileList);
254     }
255     String message;
256     ResponseEntity resEnt=null;
257     HttpHeaders responseHeaders = new HttpHeaders();
258     responseHeaders.add("Content-Type", "text/html; charset=utf
259 try {
260     int articleNO = boardService.addNewArticle(articleMap);
261     if(imageFileList!=null && imageFileList.size()!=0) {
262         for(ImageVO imageVO:imageFileList) {
263             imageFileName = imageVO.getImageFileName();
264             File srcFile = new File(ARTICLE_IMAGE_REPO+"\\\"
265             File destDir = new File(ARTICLE_IMAGE_REPO+"\\\"
266             //destDir.mkdirs();
267             FileUtils.moveFileToDirectory(srcFile, destDir,
268         }
269     }
270
271     message = "<script>";
272     message += " alert('새글을 추가했습니다.');"
273     message += " location.href='"+multipartRequest.getConte
274     message += "</script>";
275     resEnt = new ResponseEntity(message, responseHeaders, H
276
277
278 }catch(Exception e) {
279     if(imageFileList!=null && imageFileList.size()!=0) {
280         for(ImageVO imageVO:imageFileList) {
281             imageFileName = imageVO.getImageFileName();
282             File srcFile = new File(ARTICLE_IMAGE_REPO+"\\\"+"te
283             srcFile.delete();
284         }
285     }
286
287
```



```
288         message = " <script>";
289         message += " alert('오류가 발생했습니다. 다시 시도해 주세요');";
290         message += " location.href='"+multipartRequest.getContext
291         message += " </script>";
292         resEnt = new ResponseEntity(message, responseHeaders, H
293         e.printStackTrace();
294     }
295     return resEnt;
296 }
297
298 */
299
300
301
302 @RequestMapping(value = "/board/*Form.do", method = Reques
303 private ModelAndView form(HttpServletRequest request, HttpS
304     String viewName = (String)request.getAttribute("viewNam
305     ModelAndView mav = new ModelAndView();
306     mav.setViewName(viewName);
307     return mav;
308 }
309
310 //한개 이미지 업로드하기
311 private String upload(MultipartHttpServletRequest multipart
312     String imageFileName= null;
313     Iterator<String> fileNames = multipartRequest.getFileNa
314
315     while(fileNames.hasNext()){
316         String fileName = fileNames.next();
317         MultipartFile mFile = multipartRequest.getFile(file
318         imageFileName=mFile.getOriginalFilename();
319         File file = new File(ARTICLE_IMAGE_REPO + "\\\"+temp
320         if(mFile.getSize()!=0){ //File Null Check
321             if(!file.exists()){ //경로상에 파일이 존재하지 않을
322                 file.getParentFile().mkdirs(); //경로에 해당
323                 mFile.transferTo(new File(ARTICLE_IMAGE_REP
324             }
325         }
326
327     }
328     return imageFileName;
```



```
329     }
330
331     /*
332     //다중 이미지 업로드하기
333     private List<String> upload(MultipartHttpServletRequest mul
334         List<String> fileList= new ArrayList<String>();
335         Iterator<String> fileNames = multipartRequest.getFileNa
336         while(fileNames.hasNext()){
337             String fileName = fileNames.next();
338             MultipartFile mFile = multipartRequest.getFile(file
339             String originalFileName=mFile.getOriginalFilename()
340             fileList.add(originalFileName);
341             File file = new File(ARTICLE_IMAGE_REPO + "\\\"+\"temp
342             if(mFile.getSize()!=0){ //File Null Check
343                 if(!file.exists()){ //경로상에 파일이 존재하지 않을
344                     file.getParentFile().mkdirs(); //경로에 해당
345                     mFile.transferTo(new File(ARTICLE_IMAGE_REP
346                 }
347             }
348         }
349         return fileList;
350     }
351     */
352 }
353
```