```java
package com.myspring.pro30.board.controller;

import java.io.File;
import java.util.ArrayList;
import java.util.Enumeration;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

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

import org.apache.commons.io.FileUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.EnableAspectJAuto
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
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.bind.annotation.ResponseBody;
import org.springframework.web.multipart.MultipartFile;
import org.springframework.web.multipart.MultipartHttpServletRe
import org.springframework.web.servlet.ModelAndView;

import com.myspring.pro30.board.service.BoardService;
import com.myspring.pro30.board.vo.ArticleVO;
import com.myspring.pro30.board.vo.ImageVO;
import com.myspring.pro30.member.vo.MemberVO;


@Controller("boardController")
public class BoardControllerImpl  implements BoardController{
    private static final String ARTICLE_IMAGE_REPO = "C:\\board
    @Autowired
    private BoardService boardService;
    @Autowired
    private ArticleVO articleVO;

    @Override
    @RequestMapping(value= "/board/listArticles.do", method = {
    public ModelAndView listArticles(HttpServletRequest request
        String viewName = (String)request.getAttribute("viewNam
        List articlesList = boardService.listArticles();
        ModelAndView mav = new ModelAndView(viewName);
        mav.addObject("articlesList", articlesList);
        return mav;

    }

     //한 개 이미지 글쓰기
    @Override
    @RequestMapping(value="/board/addNewArticle.do" ,method = R
    @ResponseBody
    public ResponseEntity addNewArticle(MultipartHttpServletReq
    HttpServletResponse response) throws Exception {
        multipartRequest.setCharacterEncoding("utf-8");
        Map<String,Object> articleMap = new HashMap<String, Obj
        Enumeration enu=multipartRequest.getParameterNames();
        while(enu.hasMoreElements()){
            String name=(String)enu.nextElement();
            String value=multipartRequest.getParameter(name);
            articleMap.put(name,value);
        }

        String imageFileName= upload(multipartRequest);
        HttpSession session = multipartRequest.getSession();
        MemberVO memberVO = (MemberVO) session.getAttribute("me
        String id = memberVO.getId();
        articleMap.put("parentNO", 0);
        articleMap.put("id", id);
        articleMap.put("imageFileName", imageFileName);

        String message;
        ResponseEntity resEnt=null;
        HttpHeaders responseHeaders = new HttpHeaders();
        responseHeaders.add("Content-Type", "text/html; charset
        try {
            int articleNO = boardService.addNewArticle(articleM
            if(imageFileName!=null && imageFileName.length()!=0
                File srcFile = new
                File(ARTICLE_IMAGE_REPO+ "\\" + "temp"+ "\\" +
                File destDir = new File(ARTICLE_IMAGE_REPO+"\\"
                FileUtils.moveFileToDirectory(srcFile, destDir,
            }

            message = "<script>";
            message += " alert('새글을 추가했습니다.');";
            message += " location.href='"+multipartRequest.getC
            message +=" </script>";
            resEnt = new ResponseEntity(message, responseHeader
        }catch(Exception e) {
            File srcFile = new File(ARTICLE_IMAGE_REPO+"\\"+"te
            srcFile.delete();

            message = " <script>";
            message +=" alert('오류가 발생했습니다. 다시 시도해 주세요
            message +=" location.href='"+multipartRequest.getCo
            message +=" </script>";
            resEnt = new ResponseEntity(message, responseHeader
            e.printStackTrace();
        }
        return resEnt;
    }


    //한개의 이미지 보여주기
    @RequestMapping(value="/board/viewArticle.do" ,method = Req
    public ModelAndView viewArticle(@RequestParam("articleNO")
                                    HttpServletRequest request,
        String viewName = (String)request.getAttribute("viewNam
        articleVO=boardService.viewArticle(articleNO);
        ModelAndView mav = new ModelAndView();
        mav.setViewName(viewName);
        mav.addObject("article", articleVO);
        return mav;
    }

    /*
```

```
  //다중 이미지 보여주기
  @RequestMapping(value="/board/viewArticle.do" ,method = Req
  public ModelAndView viewArticle(@RequestParam("articleNO")
            HttpServletRequest request, HttpServletResponse r
      String viewName = (String)request.getAttribute("viewNam
      Map articleMap=boardService.viewArticle(articleNO);
      ModelAndView mav = new ModelAndView();
      mav.setViewName(viewName);
      mav.addObject("articleMap", articleMap);
      return mav;
  }
 */



//한 개 이미지 수정 기능
@RequestMapping(value="/board/modArticle.do" ,method = Reques
@ResponseBody
public ResponseEntity modArticle(MultipartHttpServletRequest
  HttpServletResponse response) throws Exception{
  multipartRequest.setCharacterEncoding("utf-8");
  Map<String,Object> articleMap = new HashMap<String, Object>
  Enumeration enu=multipartRequest.getParameterNames();
  while(enu.hasMoreElements()){
      String name=(String)enu.nextElement();
      String value=multipartRequest.getParameter(name);
      articleMap.put(name,value);
  }

  String imageFileName= upload(multipartRequest);
  articleMap.put("imageFileName", imageFileName);

  String articleNO=(String)articleMap.get("articleNO");
  String message;
  ResponseEntity resEnt=null;
  HttpHeaders responseHeaders = new HttpHeaders();
  responseHeaders.add("Content-Type", "text/html; charset=utf
  try {
     boardService.modArticle(articleMap);
     if(imageFileName!=null && imageFileName.length()!=0) {
       File srcFile = new File(ARTICLE_IMAGE_REPO+"\\"+"temp"
       File destDir = new File(ARTICLE_IMAGE_REPO+"\\"+articl
       FileUtils.moveFileToDirectory(srcFile, destDir, true);

       String originalFileName = (String)articleMap.get("orig
       File oldFile = new File(ARTICLE_IMAGE_REPO+"\\"+articl
       oldFile.delete();
     }
     message = "<script>";
     message += " alert('글을 수정했습니다.');";
     message += " location.href='"+multipartRequest.getContex
     message +=" </script>";
     resEnt = new ResponseEntity(message, responseHeaders, Ht
  }catch(Exception e) {
    File srcFile = new File(ARTICLE_IMAGE_REPO+"\\"+"temp"+"\
    srcFile.delete();
    message = "<script>";
    message += " alert('오류가 발생했습니다.다시 수정해주세요');";
    message += " location.href='"+multipartRequest.getContext
    message +=" </script>";
    resEnt = new ResponseEntity(message, responseHeaders, Htt
  }
  return resEnt;
}

@Override
@RequestMapping(value="/board/removeArticle.do" ,method = Req
@ResponseBody
public ResponseEntity  removeArticle(@RequestParam("articleNC
                            HttpServletRequest request, HttpS
  response.setContentType("text/html; charset=UTF-8");
  String message;
  ResponseEntity resEnt=null;
  HttpHeaders responseHeaders = new HttpHeaders();
  responseHeaders.add("Content-Type", "text/html; charset=utf
  try {
      boardService.removeArticle(articleNO);
      File destDir = new File(ARTICLE_IMAGE_REPO+"\\"+article
      FileUtils.deleteDirectory(destDir);

      message = "<script>";
      message += " alert('글을 삭제했습니다.');";
      message += " location.href='"+request.getContextPath()+
      message +=" </script>";
      resEnt = new ResponseEntity(message, responseHeaders, H

  }catch(Exception e) {
      message = "<script>";
      message += " alert('작업중 오류가 발생했습니다.다시 시도해 주세
      message += " location.href='"+request.getContextPath()+
      message +=" </script>";
      resEnt = new ResponseEntity(message, responseHeaders, H
      e.printStackTrace();
  }
  return resEnt;
}

/*
//다중 이미지 글 추가하기
@Override
@RequestMapping(value="/board/addNewArticle.do" ,method = Req
@ResponseBody
public ResponseEntity  addNewArticle(MultipartHttpServletRequ
  multipartRequest.setCharacterEncoding("utf-8");
  String imageFileName=null;

  Map articleMap = new HashMap();
  Enumeration enu=multipartRequest.getParameterNames();
  while(enu.hasMoreElements()){
      String name=(String)enu.nextElement();
      String value=multipartRequest.getParameter(name);
      articleMap.put(name,value);
  }

  //로그인 시 세션에 저장된 회원 정보에서 글쓴이 아이디를 얻어와서 Map에
  HttpSession session = multipartRequest.getSession();
  MemberVO memberVO = (MemberVO) session.getAttribute("member
  String id = memberVO.getId();
  articleMap.put("id",id);
  articleMap.put("parentNO", 0);

  List<String> fileList =upload(multipartRequest);
  List<ImageVO> imageFileList = new ArrayList<ImageVO>();
```

```
        if(fileList!= null && fileList.size()!=0) {
            for(String fileName : fileList) {
                ImageVO imageVO = new ImageVO();
                imageVO.setImageFileName(fileName);
                imageFileList.add(imageVO);
            }
            articleMap.put("imageFileList", imageFileList);
        }
        String message;
        ResponseEntity resEnt=null;
        HttpHeaders responseHeaders = new HttpHeaders();
        responseHeaders.add("Content-Type", "text/html; charset=utf
        try {
            int articleNO = boardService.addNewArticle(articleMap);
            if(imageFileList!=null && imageFileList.size()!=0) {
                for(ImageVO  imageVO:imageFileList) {
                    imageFileName = imageVO.getImageFileName();
                    File srcFile = new File(ARTICLE_IMAGE_REPO+"\\"
                    File destDir = new File(ARTICLE_IMAGE_REPO+"\\"
                    //destDir.mkdirs();
                    FileUtils.moveFileToDirectory(srcFile, destDir,
                }
            }

            message = "<script>";
            message += " alert('새글을 추가했습니다.');";
            message += " location.href='"+multipartRequest.getConte
            message +=" </script>";
            resEnt = new ResponseEntity(message, responseHeaders, H


        }catch(Exception e) {
            if(imageFileList!=null && imageFileList.size()!=0) {
              for(ImageVO  imageVO:imageFileList) {
                imageFileName = imageVO.getImageFileName();
                File srcFile = new File(ARTICLE_IMAGE_REPO+"\\"+"te
                srcFile.delete();
              }
            }


            message = " <script>";
            message +=" alert('오류가 발생했습니다. 다시 시도해 주세요');'
            message +=" location.href='"+multipartRequest.getContex
            message +=" </script>";
            resEnt = new ResponseEntity(message, responseHeaders, H
            e.printStackTrace();
        }
        return resEnt;
      }

    */



        @RequestMapping(value = "/board/*Form.do", method =  Reques
        private ModelAndView form(HttpServletRequest request, HttpS
            String viewName = (String)request.getAttribute("viewNam
            ModelAndView mav = new ModelAndView();
            mav.setViewName(viewName);
            return mav;
        }

        //한개 이미지 업로드하기
        private String upload(MultipartHttpServletRequest multipart
            String imageFileName= null;
            Iterator<String> fileNames = multipartRequest.getFileNa

            while(fileNames.hasNext()){
                String fileName = fileNames.next();
                MultipartFile mFile = multipartRequest.getFile(file
                imageFileName=mFile.getOriginalFilename();
                File file = new File(ARTICLE_IMAGE_REPO +"\\"+"temp
                if(mFile.getSize()!=0){ //File Null Check
                    if(!file.exists()){ //경로상에 파일이 존재하지 않을 
                        file.getParentFile().mkdirs();  //경로에 해당
                        mFile.transferTo(new File(ARTICLE_IMAGE_REP
                    }
                }

            }
            return imageFileName;
        }

        /*
        //다중 이미지 업로드하기
        private List<String> upload(MultipartHttpServletRequest mul
            List<String> fileList= new ArrayList<String>();
            Iterator<String> fileNames = multipartRequest.getFileNa
            while(fileNames.hasNext()){
                String fileName = fileNames.next();
                MultipartFile mFile = multipartRequest.getFile(file
                String originalFileName=mFile.getOriginalFilename()
                fileList.add(originalFileName);
                File file = new File(ARTICLE_IMAGE_REPO +"\\"+"temp
                if(mFile.getSize()!=0){ //File Null Check
                    if(!file.exists()){ //경로상에 파일이 존재하지 않을
                        file.getParentFile().mkdirs();  //경로에 해당
                        mFile.transferTo(new File(ARTICLE_IMAGE_REP
                    }
                }
            }
            return fileList;
        }
        */
    }

```