

CommendAction.java

```
package Pet.action;

import javax.annotation.Resource;

import org.springframework.stereotype.Controller;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;

import Pet.dao.CommentDao;
import Pet.entity.CommentPO;
import Pet.entity.PageBean;
import Pet.service.CommentService;
@Controller
public class CommentAction extends ActionSupport {
    private static final long serialVersionUID = 1L;

    public CommentPO getCommentPO() {
        return commentPO;
    }

    public void setCommentPO(CommentPO commentPO) {
        this.commentPO = commentPO;
    }

    private CommentPO commentPO;

    @Resource
    private CommentService commentService;
    @Resource
    private CommentDao commentDao;
    private String checkComment;

    private int currentPage = 1;

    public void setCurrentPage(int currentPage) {
        this.currentPage = currentPage;
    }

    public String findComment() {
        //System.out.println(commentService);
        PageBean<CommentPO> pageBean = commentService.findComment(currentPage);
        ActionContext.getContext().getValueStack().push(pageBean);
    }
}
```

```

        return "findCommentSuccess";
    }

    public String deleteComment() {
        System.out.println(commentService);
        System.out.println(commentP0);
        //commentDao.deleteComment(commentP0);
        commentService.deleteComment(commentP0);
        System.out.println("删除成功");
        return "deleteCommentSuccess";
    }

    public String deleteMoreComment() {
        String[] checkCommentIDs = this.getCheckComment().split(",");
        //commentDao.deleteMoreComment(checkCommentIDs);
        commentService.deleteMoreComment(checkCommentIDs);
        System.out.println("删除成功");
        return "deleteMoreCommentSuccess";
    }

    public String getCheckComment() {
        return checkComment;
    }

    public void setCheckComment(String checkComment) {
        this.checkComment = checkComment;
    }
}

```

CommentAction2.java

```
package Pet.action;

import javax.annotation.Resource;

import org.springframework.stereotype.Controller;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;

import Pet.dao.CommentDao2;
import Pet.entity.CommentP02;
import Pet.entity.PageBean;
import Pet.service.CommentService2;
@Controller
public class CommentAction2 extends ActionSupport {
    private static final long serialVersionUID = 1L;

    public CommentP02 getCommentP02() {
        return commentP02;
    }

    public void setCommentP02(CommentP02 commentP02) {
        this.commentP02 = commentP02;
    }

    private CommentP02 commentP02;

    @Resource
    private CommentService2 commentService2;
    @Resource
    private CommentDao2 commentDao2;
    private String checkComment;

    private int currentPage = 1;

    public void setCurrentPage(int currentPage) {
        this.currentPage = currentPage;
    }

    public String findComment() {
        System.out.println(commentService2);
        PageBean<CommentP02> pageBean =
commentService2.findComment(currentPage);
```

```

        ActionContext.getContext().getValueStack().push(pageBean);
        return "findCommentSuccess";
    }

    public String deleteComment() {
        System.out.println(commentService2);
        System.out.println(commentP02);
        commentService2.deleteComment(commentP02);
        System.out.println("删除成功");
        return "deleteCommentSuccess";
    }

    public String deleteMoreComment() {
        String[] checkCommentIDs = this.getCheckComment().split(",");

        commentService2.deleteMoreComment(checkCommentIDs);
        System.out.println("删除成功");
        return "deleteMoreCommentSuccess";
    }

    public String getCheckComment() {
        return checkComment;
    }

    public void setCheckComment(String checkComment) {
        this.checkComment = checkComment;
    }
}

```

CountInfoAction.java

```
package Pet.action;

import javax.annotation.Resource;

import org.springframework.stereotype.Controller;

import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.ModelDriven;
import Pet.entity.CountInfoPO;
import Pet.service.CountInfoService;

/**
 * Created by GUOFENG on 2017/4/20.
 */
@Controller
public class CountInfoAction extends ActionSupport {
    /**
     *
     */
    private static final long serialVersionUID = 1L;

    private CountInfoPO countInfoPO;
    @Resource
    private CountInfoService countInfoService;

    public CountInfoPO getCountInfoPO() {
        return countInfoPO;
    }

    public void setCountInfoPO(CountInfoPO countInfoPO) {
        this.countInfoPO = countInfoPO;
    }

    public String countInfo() {
        System.out.println(countInfoService);
        countInfoPO = countInfoService.countInfo();
        ActionContext.getContext().getSession().put("countinfo", countInfoPO);
        return "countInfoSuccess";
    }
}
```

DownloadStreamAction.java

```
package Pet.action;
import java.io.InputStream;
import java.net.URLEncoder;
import java.util.Date;

import javax.annotation.Resource;

import org.apache.struts2.ServletActionContext;

import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.ModelDriven;

import Pet.entity.PageBean;
import Pet.entity.Vedio1;
import Pet.service.DownloadService;
import Pet.service.MemberService;
public class DownloadStreamAction extends ActionSupport implements
ModelDriven<Vedio1>{
    private static final long serialVersionUID = -2747191035343710583L;
    private String fileName;
    private String vedioId;
    private String vedioTitle;
    private String vedioContent;
    private String vedioKey;
    private String vedioAuthor;
    private String vedioDate;
    private int id;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    //private String member_id;
    // public String getMember_id() {
    //     return member_id;
    // }
    // public void setMember_id(String member_id) {
    //     this.member_id = member_id;
    // }
    public String getVedioDate() {
```

```

        return vedioDate;
    }
    public void setVedioDate(String vedioDate) {
        this.vedioDate = vedioDate;
    }
    public String getVedioTitle() {
        return vedioTitle;
    }
    public void setVedioTitle(String vedioTitle) {
        this.vedioTitle = vedioTitle;
    }
    public String getVedioContent() {
        return vedioContent;
    }
    public void setVedioContent(String vedioContent) {
        this.vedioContent = vedioContent;
    }
    public String getVedioKey() {
        return vedioKey;
    }
    public void setVedioKey(String vedioKey) {
        this.vedioKey = vedioKey;
    }
    public String getVedioAuthor() {
        return vedioAuthor;
    }
    public void setVedioAuthor(String vedioAuthor) {
        this.vedioAuthor = vedioAuthor;
    }
    Vedio1 vel = new Vedio1();
    @Resource
    private MemberService memberService;
    @Resource
    private DownloadService downloadService;
    public String getVediolid() {
        return vediolid;
    }
    public void setVediolid(String vediolid) {
        this.vediolid = vediolid;
    }
    public void setFileName(String fileName) throws Exception {

        if(ServletActionContext.getRequest().getMethod().equals("GET")){

```

```

        byte[] bytes = fileName.getBytes("ISO8859-1");
        fileName=new String(bytes,"utf-8");

    }
    this.fileName = fileName;
}

public String getFileName() throws Exception {
    return URLEncoder.encode(fileName, "utf-8");
}

public InputStream getVedioStream() throws Exception {
    InputStream inputStream =
ServletActionContext.getServletContext().getResourceAsStream("/vedio/"+fileName
);

    System.out.println(inputStream);
    vel.setId(Integer.parseInt(vediolid));
    vel.setAddress(fileName);
    Date time = new Date();
    vel.setVedioDate(time);
    vel.setVedioAuthor(vedioAuthor);
    memberService.addVediol(vel);
    return inputStream;
}

@Override
public Vediol getModel() {
    // TODO Auto-generated method stub
    return vel;
}

private String checkVediol;

private int currentPage = 1;

public void setCurrentPage(int currentPage) {
    this.currentPage = currentPage;
}

public String findVediolByCondition() {
    if (vel.getVedioTitle() == null) {
        vel.setVedioTitle("");
    }
    ActionContext.getContext().getSession().put("find_key",
vel.getVedioTitle());
    PageBean<Vediol> pageBean =
downloadService.findDownloadByCondition(
        currentPage,

```



```

        vel.getVedioTitle()
    );
    ActionContext.getContext().getValueStack().push(pageBean);
    return "findDownloadByConditionSuccess";
}

public String deleteVedio1() {
    downloadService.deleteDownload(vel);
    return "deleteDownloadSuccess";
}

public String deleteMoreVedio1() {

    String[] checkVedio1IDs = this.getCheckVedio1().split(",");
    downloadService.deleteMoreDownload(checkVedio1IDs);
    return "deleteMoreDownloadSuccess";
}

public String getCheckVedio1() {
    return checkVedio1;
}

public void setCheckVedio1(String checkVedio1) {
    this.checkVedio1= checkVedio1;
}
}

```

GetFileUploadProgress.java

```
package Pet.action;

import java.io.File;
import java.util.Map;

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

import org.apache.struts2.interceptor.ServletRequestAware;
import org.apache.struts2.interceptor.ServletResponseAware;
import org.springframework.stereotype.Controller;

import Pet.entity.UploadStatus;
import net.sf.json.JSONObject;

@Controller
public class GetFileUploadProgress implements
    ServletRequestAware, ServletResponseAware {

    private File resource;
    private String resourceFileName;
    private String resourceContentType;
    private HttpServletRequest request;
    private HttpServletResponse response;
    public File getResource() {
        return resource;
    }
    public void setResource(File resource) {
        this.resource = resource;
    }
    public String getResourceFileName() {
        return resourceFileName;
    }
    public void setResourceFileName(String resourceFileName) {
        this.resourceFileName = resourceFileName;
    }
    public String getResourceContentType() {
        return resourceContentType;
    }
    public void setResourceContentType(String resourceContentType) {
```

```

        this.resourceContentType = resourceContentType;
    }
    /**
     * 获取文件上传的进度
     */
    public void getFileUploadProgress() throws Exception{

        HttpSession session=request.getSession();

        UploadStatus          status          =(UploadStatus)
session.getAttribute("uploadStatus");

        if (status == null) {
            System.out.println("uploadStatus is null");
            return;
        }

        long startTime = status.getStartTime(); //上传开始时间
        long currentTime = System.currentTimeMillis(); //现在时间
        long time = (currentTime - startTime) / 1000 + 1; //已传输的时间 单位:s

        //传输速度单位:byte/s
        double velocity = ((double) status.getBytesRead()) / (double) time;
        //估计总时间
        double totalTime = status.getContentLength();
        //估计剩余时间
        double timeLeft = totalTime - time;
        //已经完成的百分比
        int percent = (int) (100 * (double) status.getBytesRead() / (double)
status.getContentLength());
        //已经完成数单位:m
        double length = ((double) status.getBytesRead()) / 1024 / 1024;
        //总长度 单位:m
        double totalLength = ((double) status.getContentLength()) / 1024 / 1024;

        System.out.println("bytesRead:"+status.getBytesRead());
        System.out.println("ContentLength:"+status.getContentLength());

        //      System.out.println("percent:"+percent);
        //      System.out.println("length:"+length);
        //      System.out.println("totalLength:"+totalLength);
        //      System.out.println("velocity:"+velocity);
        //      System.out.println("time:"+time);
        //      System.out.println("totalTime:"+totalTime);
    }

```

```

//      System.out.println("timeLeft:"+timeLeft);
//      System.out.println("fileNumber:"+status.getItems());

JSONObject jsonObject = new JSONObject();

jsonObject.put("percent", String.valueOf(percent));
jsonObject.put("length", String.valueOf(length));
jsonObject.put("totalLength", String.valueOf(totalLength));
jsonObject.put("velocity", String.valueOf(velocity));
jsonObject.put("time", String.valueOf(time));
jsonObject.put("totalTime", String.valueOf(totalTime));
jsonObject.put("timeLeft", String.valueOf(timeLeft));
jsonObject.put("fileNumber", String.valueOf(status.getItems()));

response.setContentType("text/html;charset=utf-8");
response.setCharacterEncoding("utf-8");
response.getWriter().write(jsonObject.toString());
}

@Override
public void setServletResponse(HttpServletResponse response) {
    // TODO Auto-generated method stub
    this.response=response;
}

@Override
public void setServletRequest(HttpServletRequest request) {
    // TODO Auto-generated method stub
    this.request=request;
}

}

```

IndexAction.java

```
package Pet.action;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;

import com.opensymphony.xwork2.ActionSupport;

import Pet.service.IndexService;
import net.sf.json.JSONArray;

/**
 *
 */
@Controller
public class IndexAction extends ActionSupport {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    @Autowired
    private IndexService indexService;
    private JSONArray root;

    private int hotCurrentPage = 1;
    private int societyCurrentPage = 1;
    private int techCurrentPage = 1;
    private int inteCurrentPage = 1;
    private int miliCurrentPage = 1;

    private int thisNewsId;

    private int thisVedioId;

    public int getThisVedioId() {
        return thisVedioId;
    }

    public void setThisVedioId(int thisVedioId) {
        this.thisVedioId = thisVedioId;
    }

    public void setIndexService(IndexService indexService) {
```

```

        this.indexService = indexService;
    }

    public JSONArray getRoot() {
        return root;
    }

    public void setRoot(JSONArray root) {
        this.root = root;
    }
/**
 *find News
 * @return
 */
    public String findThisNews() {
        root = JSONArray.fromObject(indexService.findNewsOfThis(thisNewsId));
        return SUCCESS;
    }
/**
 * find Vedio
 * @return
 */
    public String findThisVedio() {

        root = JSONArray.fromObject(indexService.findVedioOfThis(thisVedioId));
        return SUCCESS;
    }
    private void restrat() {
        hotCurrentPage = 0;
        societyCurrentPage = 0;
        techCurrentPage = 0;
        inteCurrentPage = 0;
        miliCurrentPage = 0;
    }

    public String findHeadNews() {
        restrat();
        root = JSONArray.fromObject(indexService.findNewsOfHead());
        return SUCCESS;
    }

    public String findHotNews() {
        int pageCount = 5;
        int pageSize = indexService.PageSizeOfHot(pageCount);

```

```

        if (hotCurrentPage < pageSize) {
            hotCurrentPage++;
        } else {
            hotCurrentPage = 1;
        }
        root = JSONArray.fromObject(indexService.findNewsOfHot(hotCurrentPage,
pageCount));
        return SUCCESS;
    }

    public String findSocietyNews() {
        int pageCount = 5;
        int pageSize = indexService.PageSizeOfSociety(pageCount);
        if (societyCurrentPage < pageSize) {
            societyCurrentPage++;
        } else {
            societyCurrentPage = 1;
        }
        root
=
JSONArray.fromObject(indexService.findNewsOfSociety(societyCurrentPage,
pageCount));
        return SUCCESS;
    }

    public String findTechNews() {
        int pageCount = 5;
        int pageSize = indexService.PageSizeOfTech(pageCount);
        if (techCurrentPage < pageSize) {
            techCurrentPage++;
        } else {
            techCurrentPage = 1;
        }
        root = JSONArray.fromObject(indexService.findNewsOfTech(techCurrentPage,
pageCount));
        return SUCCESS;
    }

    public String findInteNews() {
        int pageCount = 5;
        int pageSize = indexService.PageSizeOfInte(pageCount);
        if (inteCurrentPage < pageSize) {
            inteCurrentPage++;
        } else {
            inteCurrentPage = 1;
        }
    }

```

```

        root = JSONArray.fromObject(indexService.findNewsOfInte(inteCurrentPage,
pageCount));
        return SUCCESS;
    }
    public String findMiliNews() {
        int pageCount = 5;
        int pageSize = indexService.PageSizeOfMili(pageCount);
        if (miliCurrentPage < pageSize) {
            miliCurrentPage++;
        } else {
            miliCurrentPage = 1;
        }
        root = JSONArray.fromObject(indexService.findNewsOfMili(miliCurrentPage,
pageCount));
        return SUCCESS;
    }
    public void setHotCurrentPage(int hotCurrentPage) {
        this.hotCurrentPage = hotCurrentPage;
    }

    public void setSocietyCurrentPage(int societyCurrentPage) {
        this.societyCurrentPage = societyCurrentPage;
    }

    public void setTechCurrentPage(int techCurrentPage) {
        this.techCurrentPage = techCurrentPage;
    }

    public void setInteCurrentPage(int inteCurrentPage) {
        this.inteCurrentPage = inteCurrentPage;
    }

    public void setMiliCurrentPage(int miliCurrentPage) {
        this.miliCurrentPage = miliCurrentPage;
    }

    public int getThisNewsId() {
        return thisNewsId;
    }

    public void setThisNewsId(int thisNewsId) {
        this.thisNewsId = thisNewsId;
    }
}

```


LoginRegAction.java

```
package Pet.action;

import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

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

import org.apache.struts2.interceptor.ServletRequestAware;
import org.apache.struts2.interceptor.ServletResponseAware;
import org.springframework.stereotype.Controller;

import com.opensymphony.xwork2.ActionSupport;

import Pet.entity.Manager;
import Pet.entity.Member;
import Pet.service.LoginRegService;
import net.sf.json.JSONArray;

/**
 *
 */
@Controller
public class LoginRegAction extends ActionSupport implements
    ServletRequestAware, ServletResponseAware {

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private Member member;

    public Member getMember() {
        return member;
    }

    public void setMember(Member member) {
        this.member = member;
    }
}
```

```

public LoginRegService getLoginRegService() {
    return loginRegService;
}

private LoginRegService loginRegService;
private JSONArray root;
private HttpServletRequest request;
private HttpServletResponse response;

public void setLoginRegService(LoginRegService loginRegService) {
    this.loginRegService = loginRegService;
}

/**
 * 注册
 */
public String register() {
    if (loginRegService.register(member)) {
        root = JSONArray.fromObject(1);
    } else {
        root = JSONArray.fromObject(0);
    }
    return SUCCESS;
}

public String login() {
    //Member member1 = new Member();
    member.setId(Integer.parseInt(request.getParameter("userid")));
    member.setPwd(request.getParameter("password"));
    Member loginMember = loginRegService.login(member);
    if (loginMember == null) {
        root = JSONArray.fromObject(0);
        return SUCCESS;
    } else {
        root = JSONArray.fromObject(loginMember);
        String jsonstr = root.toString();
        try {
            //URLEncoder.encode(jsonstr, "utf-8");
            Cookie cookie = new Cookie("news_user", URLEncoder.encode(jsonstr,
"utf-8"));

            response.addCookie(cookie);
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}

```

```

        }
        return SUCCESS;
    }
}

public JSONArray getRoot() {
    return root;
}

public void setRoot(JSONArray root) {
    this.root = root;
}

@Override
public void setServletRequest(HttpServletRequest request) {
    // TODO Auto-generated method stub
    this.request = request;
}

@Override
public void setServletResponse(HttpServletResponse response) {
    this.response= response;
}
}

```

ManagerAction.java

```
package Pet.action;

import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;

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

import org.apache.struts2.ServletActionContext;
import org.apache.struts2.interceptor.ServletRequestAware;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ResponseBody;

import com.opensymphony.xwork2.ActionSupport;

import Pet.entity.Danmu;
import Pet.entity.Manager;
import Pet.entity.Member;
import Pet.entity.News;
import Pet.entity.Vedio;
import Pet.entity.Vedio1;
import Pet.service.Constant;
import Pet.service.ManagerService;
import Pet.service.MemberService;
import Pet.service.VedioService;
import Pet.tool.Page;
import net.sf.json.JSONArray;
import net.sf.json.JSONObject;

@Controller
public class ManagerAction extends ActionSupport implements
ServletRequestAware, Constant {

    /**
     *
     */
    private static final long serialVersionUID = 1L;
```

```

private int page;
private Page page1;
public int getPage() {
    return page;
}

public void setPage(int page) {
    this.page = page;
}

public Page getPage1() {
    return page1;
}

public void setPage1(Page page1) {
    this.page1 = page1;
}
private HttpServletRequest request;

@Autowired
private ManagerService managerService;
@Autowired
private MemberService memberService;
@Autowired
private VedioService vedioService;
private Manager manager;

private String error;
private String oldPassword;
private String hql;

public String getHql() {
    return hql;
}

public void setHql(String hql) {

```

```

        this.hql = hql;
    }

    public Manager getManager() {
        return manager;
    }

    public void setManager(Manager manager) {
        this.manager = manager;
    }

    public String getError() {
        return error;
    }

    public void setError(String error) {
        this.error = error;
    }

    public String login() throws Exception {
        HttpSession session=request.getSession();
        Manager
currentmanager=managerService.findManagerByIdAndPassword(manager);
        if(currentmanager!=null){
            session.setAttribute("admininfo", currentmanager);
            return "loginSuccess";
        }
        else{
            error="用后名或者密码错误! ";
            return "login1";
        }
    }

    @Override
    public void setServletRequest(HttpServletRequest request) {
        // TODO Auto-generated method stub

```

```

        this.request=request;
    }
    public void find() {
        Date date = new Date();
        SimpleDateFormat date1 =new SimpleDateFormat("yy-MM-dd");
        int page2 = Integer.parseInt(request.getParameter("page2"));
        HttpSession session=request.getSession();

        hql = "from News";
        List<News> list=managerService.find(hql,page2);
        HttpServletResponse hsr = ServletActionContext.getResponse();
        JSONArray json = new JSONArray();
        for(News a : list){
            JSONObject jo = new JSONObject();

            jo.put("newsDate", date1.format(date));
            jo.put("newsId", a.getNewsId());
            jo.put("newsContent", a.getNewsContent());
            jo.put("newsTitle", a.getNewsTitle());
            json.add(jo);
        }
        try {
            //System.out.println(json.toString());
            hsr.setCharacterEncoding("UTF-8");
            hsr.getWriter().write(json.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}

/**
 * news fenye
 * @return
 */
@ResponseBody
public String getPageList() {
    this.page1 = managerService.queryForPage(Constant.EVERYPAGE, page);
    HttpSession session=request.getSession();
    session.setAttribute("news", this.page1);
    System.out.println("成功");
    return SUCCESS;
}

/**

```

```

    * news search
    * @return
    */
public String search() {
    String T = request.getParameter("s");
    this.page1 = managerService.queryForPage1(Constant.EVERYPAGE, page, T);
    List<News> list = this.page1.getList();
    return "newList";
}

@ResponseBody
public String getPageList1() {
    this.page1 = managerService.queryForPage2(Constant.EVERYPAGE, page);
    HttpSession session=request.getSession();
    session.setAttribute("vedio", this.page1);
    System.out.println("成功");
    return "SUCCESS1";
}
/**
 * vedio search
 * @return
 */
public String search1() {
    String T = request.getParameter("s");
    this.page1 = managerService.queryForPage3(Constant.EVERYPAGE, page, T);
    List<Vedio> list = this.page1.getList();
    return "vedioList";
}

@ResponseBody
public String getPageList2() {
    this.page1 = managerService.queryForPage4(Constant.EVERYPAGE, page);
    HttpSession session=request.getSession();
    session.setAttribute("vedio1", this.page1);
    System.out.println("成功");
    return "SUCCESS2";
}

public String search2() {
    String T = request.getParameter("s");
    this.page1 = managerService.queryForPage5(Constant.EVERYPAGE, page, T);
    List<Vedio1> list = this.page1.getList();
    return "vedio1List";
}

```



```

    }
    public String updatePassword() {
        if (manager.getId() == 0) {
            return "updatePwdSuccess";
        }
        oldPassword=request.getParameter("oldPassword");
        managerService.updatePassword(manager, oldPassword);
        return "updatePwdSuccess";
    }

    public String getOldPassword() {
        return oldPassword;
    }

    public void setOldPassword(String oldPassword) {
        this.oldPassword = oldPassword;
    }
    private Danmu danmu;
    public Danmu getDanmu() {
        return danmu;
    }

    public void setDanmu(Danmu danmu) {
        this.danmu = danmu;
    }

    private int id;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }

    public void findDanmu() throws IOException {
        System.out.println(managerService);
        int id=Integer.parseInt(request.getParameter("id"));
        List<Danmu> danmuList = managerService.findDanmus(id);
        HttpServletResponse response = ServletActionContext.getResponse();
        if(danmuList.size() != 0 && danmuList != null) {
            JSONArray json = new JSONArray();
            for(Danmu a : danmuList) {
                JSONObject jo = new JSONObject();

```

```

        jo.put("text", a.getText());
        jo.put("color", a.getColor());
        jo.put("size", a.getSize());
        jo.put("position", a.getPosition());
        jo.put("time", a.getTime());
        json.add(jo);
    }
    try {
        //System.out.println(json.toString());
        response.setCharacterEncoding("UTF-8");
        response.getWriter().write(json.toString());
        System.out.println(json.toString());
    } catch (IOException e) {
        e.printStackTrace();
    }
}

}

public void addDanmu() {
    System.out.println("方法进入成功");
    System.out.println(request.getParameter("json"));
    JSONObject json=JSONObject.fromObject(request.getParameter("json"));
    // JSONArray jsonParam = JSONArray.fromObject(object)
    // for(int i=0;i<jsonParam.size();i++)
    // {
    //     JSONObject obj=(JSONObject) jsonParam.get(i);
    //     String text=(String) json.get("text");

    String color=(String) json.get("color");

    //int size = (int) json.get("size");
    int size = Integer.parseInt(json.get("size").toString());
    //System.out.println("1111111");
    int position = Integer.parseInt(json.get("position").toString());
    //int position = Integer.parseInt((String) json.get("position"));
    int time = Integer.parseInt(json.get("time").toString());
    //int time = Integer.parseInt((String) json.get("time"));
    int id = Integer.parseInt(request.getParameter("id1"));//弹幕发送人
    int id2 = Integer.parseInt(request.getParameter("id2"));//视频编号
    System.out.println(id);
    Danmu danmu1 = new Danmu();
    danmu1.setText(text);
    danmu1.setColor(color);
    danmu1.setSize(size);
}

```

```

        danmul.setPosition(position);
        danmul.setTime(time);
        Member member = memberService.find(id);
        Vedio vedio = vedioService.find(id2);
        member.setId(id);
        vedio.setId(id2);
        danmul.setMember(member);
        danmul.setVedio(vedio);

        managerService.addDanmu(danmul);
//    }

}

public void findl() {
    Date date = new Date();
    SimpleDateFormat datel =new SimpleDateFormat("yy-MM-dd");
    int page2 = Integer.parseInt(request.getParameter("page2"));
    HttpSession session=request.getSession();
    hql = "from Vedio";
    List<Vedio> list=managerService.findl(hql,page2);
    HttpServletResponse hsr = ServletActionContext.getResponse();
    JSONArray json = new JSONArray();
    for(Vedio a : list){
        JSONObject jo = new JSONObject();

        jo.put("vedioDate", datel.format(date));
        jo.put("id", a.getId());
        jo.put("vedioContent", a.getVedioContent());
        jo.put("vedioTitle", a.getVedioTitle());
        json.add(jo);
    }
    try {
        //System.out.println(json.toString());
        hsr.setCharacterEncoding("UTF-8");
        hsr.getWriter().write(json.toString());
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```

MemberAction.java

```
package Pet.action;

import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.net.URLEncoder;
import java.util.Date;
import java.util.Properties;

import javax.annotation.Resource;
import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.commons.io.FileUtils;
import org.apache.struts2.interceptor.ServletRequestAware;
import org.apache.struts2.interceptor.ServletResponseAware;
import org.springframework.stereotype.Controller;

import com.opensymphony.xwork2.ActionSupport;

import Pet.dao.BaseDao;
import Pet.entity.Member;
import Pet.entity.Vedio;
import Pet.service.MemberService;
import Pet.utils.Constant;
import net.sf.json.JSONArray;

@Controller
public class MemberAction extends ActionSupport implements
    ServletRequestAware, ServletResponseAware{

    private static final long serialVersionUID = 1L;
```

```

private HttpServletRequest request;

@Resource
private MemberService memberService;

private Member member;

private Vedio1 vedio1;
public Vedio1 getVedio1() {
    return vedio1;
}

public void setVedio1(Vedio1 vedio1) {
    this.vedio1 = vedio1;
}

public JSONArray getRoot() {
    return root;
}

public void setRoot(JSONArray root) {
    this.root = root;
}
private JSONArray root;

private String error;
public String getOldPassword() {
    return oldPassword;
}

public void setOldPassword(String oldPassword) {
    this.oldPassword = oldPassword;
}
private String oldPassword;
private HttpServletResponse response;
@Resource
private BaseDao<Member> baseDao;
public Member getMember() {
    return member;
}

public void setMember(Member member) {
    this.member = member;
}

```

```

    }

    public String getError() {
        return error;
    }

    public void setError(String error) {
        this.error = error;
    }

    private File file;
    private String fileFileName;
    public String login() throws Exception{
        HttpSession session=request.getSession();
        Member currentMember=memberService.findMemberByNameAndPassword(member);
        if(currentMember!=null&&currentMember.getState()!=0){
            session.setAttribute("currentMember", currentMember);
            root = JSONArray.fromObject(currentMember);
            String jsonstr = root.toString();
            Cookie cookie = new Cookie("user",URLEncoder.encode(jsonstr, "utf-
8"));
            response.addCookie(cookie);
            return SUCCESS;
        }
        else if(currentMember!=null&&currentMember.getState()==0){

            session.setAttribute("message",currentMember.getEmail());
            error=" ";
            root = JSONArray.fromObject(0);
            String jsonstr = root.toString();
            Cookie cookie = new Cookie("user",URLEncoder.encode(jsonstr, "utf-
8"));
            response.addCookie(cookie);
            return ERROR;
        }
        else{
            error="";
            session.setAttribute("message",error);
            root = JSONArray.fromObject(0);
            String jsonstr = root.toString();
            Cookie cookie = new Cookie("user",URLEncoder.encode(jsonstr, "utf-
8"));
            response.addCookie(cookie);
            return ERROR;
        }
    }

```

```

    }
}

public File getFile() {
    return file;
}

public void setFile(File file) {
    this.file = file;
}

public String getFileFileName() {
    return fileFileName;
}

public void setFileFileName(String fileFileName) {
    this.fileFileName = fileFileName;
}

public String regist() throws Exception{
    String account = String.valueOf(member.getAccount());
    String url = "";
    url += "?account="+account+"&pwd="+member.getPwd();
    System.out.println(url);
    memberService.Insert(member);
    SendMail(member, url);
    return INPUT;
}

@Override
public void setServletRequest(HttpServletRequest request) {
    // TODO Auto-generated method stub
    this.request=request;
}

public void SendMail(Member member, String url) {
    InputStream is =
this.getClass().getResourceAsStream("/mailInfo.properties");
    Properties prop = new Properties();
    try {
        prop.load(is);
    } catch (IOException e1) {
        e1.printStackTrace();
    }
}

```

```

        String msgText = "请点击下面链接进行激活\n" +
"http://localhost:8080/Pet/Member_process.action"+url;
        String smtpHost = prop.get("smtpHost").toString();
        String from = prop.get("mailName").toString();
        String pwd = prop.get("pwd").toString();
        String to = member.getEmail();

        Properties props = new Properties();

        props.put("mail.smtp.host", smtpHost);
        props.put("mail.smtp.auth", "true");

        Session session = Session.getDefaultInstance(props, null);
        MimeMessage message = new MimeMessage(session);
        try {
            message.setFrom(new InternetAddress(from));
            InternetAddress[] address = {new InternetAddress(to)};
            message.setRecipients(Message.RecipientType.TO, address);
            message.setSubject("");
            message.setSentDate(new Date());
            message.setText(msgText);

            message.saveChanges(); // implicit with send()

            Transport transport = session.getTransport("smtp");
            transport.connect(smtpHost, from, pwd);
            transport.sendMessage(message, message.getAllRecipients());
            transport.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public String process() throws Exception {
        Member member = new Member();
        HttpSession session = request.getSession();
        int account = Integer.parseInt(request.getParameter("account"));
        String pwd = request.getParameter("pwd");
        member.setAccount(account);
        member.setPwd(pwd);
        Member currentMember =
memberService.findMemberByNameAndPassword(member);
        member.setState(1);
        member.setId(currentMember.getId());
        member.setEmail(currentMember.getEmail());
    }

```



```

        memberService.update(member);
        session.setAttribute("message", currentMember);
        return LOGIN;
    }

    public String addVedio1() {
        if (file != null) {
            try {
                File tempFile = new File(Constant.systemPath, fileFileName);
                FileUtils.copyFile(file, tempFile);
                vedio1.setAddress(fileFileName);
            } catch (IOException e) {
                return INPUT;
            }
        }
        memberService.addVedio1(vedio1);
        return "addNewsSuccess";
    }

    @Override
    public void setServletResponse(HttpServletResponse response) {

        this.response=response;
    }

    public String updatePassword() {
        oldPassword=request.getParameter("oldPassword");
        if(memberService.updatePassword(member, oldPassword))
            return "updatePwdSuccess";
        return "updatePwdSuccess";
    }
}

```

NewsAction.java

```
package Pet.action;

import java.io.File;
import java.io.IOException;

import javax.annotation.Resource;

import org.apache.commons.io.FileUtils;
import org.springframework.stereotype.Controller;
import org.springframework.transaction.annotation.Transactional;

import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.ModelDriven;

import Pet.dao.NewsDao;
import Pet.entity.News;
import Pet.entity.NewsTypeP0;
import Pet.entity.PageBean;
import Pet.service.NewsService;
import Pet.utils.Constant;

/**
 * Created by GUOFENG on 2017/4/16.
 */
@Controller
public class NewsAction extends ActionSupport implements ModelDriven<News>{

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Resource
    private NewsService newsService;
    private News news = new News();
    private File file;
    private String fileFileName;
    private String checkNews;
```

```

private int currentPage = 1;

public void setCurrentPage(int currentPage) {
    this.currentPage = currentPage;
}

public String addNews() {
    if (file != null) {
        try {
            File tempFile = new File(Constant.systemPath, fileFileName);
            FileUtils.copyFile(file, tempFile);
            news.setNewsImg(fileFileName);
        } catch (IOException e) {
            return INPUT;
        }
    }
    newsService.addNews(news);
    return "addNewsSuccess";
}

public String findNewsByCondition() {
    if (news.getNewsTypeP0() == null) {
        NewsTypeP0 newsTypeP0 = new NewsTypeP0();
        newsTypeP0.setTypeId(0);
        news.setNewsTypeP0(newsTypeP0);
    }
    if (news.getNewsKey() == null) {
        news.setNewsKey("");
    }
    ActionContext.getContext().getSession().put("find_type",
news.getNewsTypeP0().getTypeId());
    ActionContext.getContext().getSession().put("find_key",
news.getNewsKey().trim());
    PageBean<News> pageBean = newsService.findNewsByCondition(
        currentPage,
        news.getNewsTypeP0().getTypeId(),
        news.getNewsKey()
    );
    ActionContext.getContext().getValueStack().push(pageBean);
    return "findNewsByConditionSuccess";
}

```

```

    public String deleteNews() {
        System.out.println("删除成功");
        newsService.deleteNews(news);
        return "deleteNewsSuccess";
    }

    public String deleteMoreNews() {
        String[] checkNewsIDs = this.getCheckNews().split(",");
        newsService.deleteMoreNews(checkNewsIDs);
        System.out.println("删除成功");
        return "deleteMoreNewsSuccess";
    }

    public String getCheckNews() {
        return checkNews;
    }

    public void setCheckNews(String checkNews) {
        this.checkNews = checkNews;
    }

    public File getFile() {
        return file;
    }

    public void setFile(File file) {
        this.file = file;
    }

    public String getFileFileName() {
        return fileFileName;
    }

    public void setFileFileName(String fileFileName) {
        this.fileFileName = fileFileName;
    }

    @Override
    public News getModel() {
        // TODO Auto-generated method stub
        return news;
    }
}

```

ResourceAction.java

```
package Pet.action;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts2.interceptor.ServletRequestAware;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;

import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.ModelDriven;

import Pet.entity.Member;
import Pet.entity.PageBean;
import Pet.entity.Resource;
import Pet.service.ResourceService;

@Controller
public class ResourceAction extends ActionSupport implements
ModelDriven<Resource>, ServletRequestAware{

    private static final long serialVersionUID = 1L;

    @Autowired
    private ResourceService resourceService;
    Resource resource=new Resource() ;

    private String checkResource;

    private int currentPage = 1;

    private String idea;

    private String introduce;

    //在 struts.xml 取值就必须要有 get 方法
    public String getIdea() {
        return idea;
    }
}
```

```

//在 struts.xml 中设值就必须要有 set 方法
public void setIdea(String idea) {
    this.idea = idea;
}
public void setIdeal(String ieda) {
    int ideal = Integer.parseInt(ieda);
    this.ideal = ideal;
}
private int ideal=1;
public int getIdeal() {
    return ideal;
}
public String getIntroduce() {
    return introduce;
}
public void setIntroduce(String introduce) {
    this.introduce = introduce;
}
private HttpServletRequest request;

public void setCurrentPage(int currentPage) {
    this.currentPage = currentPage;
}
public String findResourceByCondition() {
    System.out.println(ideal+"###");
    System.out.println(idea);
    System.out.println(ideal+"###");
    int idea2 = Integer.parseInt(idea);
    if (resource.getName() == null) {
        resource.setName("");
    }
    //int id1=Integer.parseInt(request.getParameter("idea"));
    //ideal = Integer.parseInt(idea);
    PageBean<Resource> pageBean = resourceService.findResourceByCondition(
        currentPage,
        idea2
    );
    ActionContext.getContext().getValueStack().push(pageBean);
    return "findResourceByConditionSuccess";
}
public String findResourceByCondition1() {

    if (resource.getIntroduce() == null) {
        resource.setIntroduce("");
    }
}

```

```

    }

    PageBean<Resource> pageBean = resourceService.findResourceByCondition1(
        currentPage,
        resource.getIntroduce()
    );
    ActionContext.getContext().getValueStack().push(pageBean);
    return "findResourceByConditionSuccess";
}

public String deleteResource() {
    resourceService.deleteResource(resource);
    return "deleteResourceSuccess";
}

public String deleteMoreResource() {

    String[] checkResourceIDs = this.getCheckResource().split(",");
    resourceService.deleteMoreResource(checkResourceIDs);
    System.out.println("删除成功");
    return "deleteMoreResourceSuccess";
}

public String getCheckResource() {
    return checkResource;
}

public void setCheckResource(String checkResource) {
    this.checkResource = checkResource;
}

@Override
public Resource getModel() {
    // TODO Auto-generated method stub
    return resource;
}

@Override
public void setServletRequest(HttpServletRequest request) {
    // TODO Auto-generated method stub
    this.request=request;
}
}

```