Spring mvc+jsp；

1，基本：---我的springmvc+freemarker的使用方式；

Spring mvc：---springmvc的使用方式；

Web.xml配置，

spring-servlet.xml配置，

~

*Controller：*

@Controller

**public** **class** AboutUsController {

@RequestMapping(value = "/aboutUs", method = RequestMethod.*GET*)

//ModelMap model传递参数到页面，

**public** String aboutUs(ModelMap model) {

model.addAttribute("str", "陈adc123"); //页面参数输出

**return** "aboutUs";

}

}

页面，

aboutUs.ftl页面，

<h1>say hello ${str}</h1>

---------------------

项目文件夹，controller类的命名规范，否则找不到对应信息；

当然，应该有对应的需求文档(页面，controller，业务逻辑的对应记录)

//////////////////////////////////////////////////////////////////////////

2，具体：---公司的springmvc+jsp的使用方式；

页面加载模式：

1），动作.do---controller类---对应的方法，加载必要的初始参数，然后转到页面；

2），页面,js,ajax，动作.do---controller类---对应的方法，查询列表数据，返回页面；---页面显示到上一页面的div中；

3），普通的form表单提交数据：---不使用ajax；| 不能返回数据提示，返回数据等；

应该可以通过，（后台绑定参数，前台jqurey的ready方法，显示提示信息或加载数据；）

（或，跳转到相应页面的页面加载方法中，然后由该方法返回页面，绑定数据；）

Controller类：

可以不要类级别的参数验证，

如：@RequestMapping(value = "/transfer")

.do直接查找方法上的映射；

2.1动作直接到controller：

transfer/toTransferList.do---TransferController.java，toTransferList()方法，加载必要的参数，然后中转到页面---transfer/transferList；

@Controller

@RequestMapping(value = "/transfer")

**public** **class** TransferController **extends** BaseController{

//加载必要的参数，中转页面；

@RequestMapping(value = "toTransferList")

**public** ModelAndView toTransferList(HttpServletRequest request) {

ModelAndView mv = **new** ModelAndView("transfer/transferList");

**return** mv;

}

2.2页面： | 页面，js-ajax中，action方法，查询数据，然后加载到页面；

<div id=*"dataList"*> </div>

Js，

$(document).ready(**function**(){

loadTransferList(1);

});

**function** loadTransferList(pageNum){

$.ajax({

url : '${path}/transfer/queryTransferList.do',

data : {pageNum:pageNum,pageSize:10,'borrowType':BorrowType, 'limitTime':LimitTime,'repayType':RepayType,'orderBy':OrderBy,'orderType':orderType},

type : 'post',

dataType : 'text',

success : **function**(data){

$("#dataList").html(data);

},

error : **function**(data) {

alert("网络连接异常，请刷新页面或稍后重试！");

}

});

}

@RequestMapping(value = "queryTransferList")

**public** ModelAndView queryTransferList(@RequestParam Integer pageNum, @ModelAttribute SearchTransferBean seach) {

ModelAndView mv = **new** ModelAndView("transfer/transferGridByPage");

Page p = **null**;

**try** {

p = transferService.findtransferList(seach, **new** Page(pageNum, BusinessConstants.***DEFAULT\_PAGE\_SIZE***));

} **catch** (Exception e) {

logger.error("进入债权转让出错");

e.printStackTrace();

}

mv.addObject("page", p);

**return** mv;

}

~

transferGridByPage.jsp，显示到transferList.jsp页面的div中；

页面数据显示：

<c:forEach items=*"*${page.result}*"* var=*"transfer"* varStatus= *"idx"*>

<td align=*"center"* width=*"80px;"*><i>${transfer.borrowOrder}</i></td>

</c:forEach>

----------------------------------

其它：

带id的动作，

transfer/toTransferSubscribe/${transfer.id}.do

@RequestMapping(value = "toTransferSubscribe/{id}")

**public** ModelAndView toTransferSubscribe(HttpServletRequest request, HttpServletResponse response, @PathVariable Integer id) {

跳转：一个.do跳转到另一个.do；---url改变；

// 栏目，行业新闻---跳转到公共处理方法中；

@RequestMapping(value = "hangye")

**public** ModelAndView channelTo() {

Integer id = 35;

ModelAndView mv = **new** ModelAndView("redirect:channel/"+id+".do?pageNum=1");

**return** mv;

}

// 栏目列表 | 查询需要的数据:栏目信息，栏目下的文章；

@RequestMapping(value = "channel/{id}")

**public** ModelAndView tochannel(@PathVariable Integer id,@RequestParam Integer pageNum) {

ModelAndView mv = **new** ModelAndView("cms/informationChannel");

~url不变的写法：

</li><li><a href=*"*${path}*/hangye.html"*>行业新闻</a>

·

Cotroller，去掉路径；

@Controller

**public** **class** CmsArticleCotroller **extends** BaseController {

//不需要路径参数，

**public** ModelAndView tochannel(@PathVariable Integer id, @RequestParam Integer pageNum) {

@RequestMapping(value = "hangye")

**public** ModelAndView channelTo() {

CmsChannel channel = cmsChannelService.getChannelByPinyin("hangye");

Integer id = channel.getId();

**return** tochannel(id, 1);

}

公用：---竟然可以这样写； | 我害怕找不到，竟然能找到！！

@RequestMapping(value = "/{pinyin}")

**public** ModelAndView channelTo3(@PathVariable("pinyin") String pinyin) {

CmsChannel channel = cmsChannelService.getChannelByPinyin(pinyin);

Integer id = channel.getId();

**return** tochannel(id, 1);

}

分页支持： 有分页后缀和没有的不同的url，

</li><li><a href=*"*${path}*/hangye.html"*>行业新闻</a>

**function** findPage(pageNum) {

**var** urlCode = "${cmsChannel.urlCode }";

location.href = "${path }/"+urlCode+".html?pageNum="+pageNum;

}

//使用request接收，

@RequestMapping(value = "/{pinyin}")

**public** ModelAndView channelTo3(@PathVariable("pinyin") String pinyin, HttpServletRequest request) {

CmsChannel channel = cmsChannelService.getChannelByPinyin(pinyin);

Integer id = channel.getId();

Integer pageNum = 1;

**if**(request.getParameter("pageNum")!=**null**){

pageNum = Integer.*parseInt*(request.getParameter("pageNum"));

}

**return** tochannel(id, pageNum);

}

-------------

url带拼接的实现方式1：

页面：

</li><li><a href=*"*${path }*/*baike*.html"*>个人理财</a>

</li><li><a href=*"*${path }*/*baike*-1.html"*>个人理财</a>

方法1，| 拆分；

// 栏目跳转---跳转到公共处理方法中；

@RequestMapping(value = "/{pinyin}")

**public** ModelAndView channelTo3(@PathVariable("pinyin") String pinyin, HttpServletRequest request) {

Integer pageNum = 1;

String url1 = pinyin;

**if** (url1.indexOf("-") != -1) {

String[] c = url1.split("-");

pinyin = c[0];

pageNum = Integer.*parseInt*(c[1]);

}

CmsChannel channel = cmsChannelService.getChannelByPinyin(pinyin);

Integer id = channel.getId();

**return** tochannel(id, pageNum);

}

方法2，| 实现，不带映射路径；

**public** ModelAndView tochannel(@PathVariable Integer id, @RequestParam Integer pageNum) {

url带拼接的实现方式2：

页面：

</li><li><a href=*"*${path }*/*baike*.html"*>个人理财</a>

</li><li><a href=*"*${path }*/*baike*-1.html"*>个人理财</a>

controller中：---2个方法就好了；

方法1，| 有-1的进这个，

@RequestMapping(value = "/baike-{initial}")

**public** ModelAndView indexInitial(@PathVariable("initial") String initial) **throws** Exception {

**return** index(initial);

}

//方法2，| 无-1的进这个；

@RequestMapping(value = "/baike")

**public** ModelAndView index(String initai) **throws** Exception {

////////////////////////////////////////////////////////////////////

Controller：业务逻辑处理；

//日志记录，

Logger logger = LoggerFactory.*getLogger*(CmsArticleCotroller.**class**);

//引入需要的service，

@Autowired

**private** CmsArticleService cmsArticleService;

//数据查询操作等；

//返回页面，或携带数据；

ModelAndView mv = **new** ModelAndView("cms/informationChannel");

mv.addObject("cmsChannel", cmsChannel);

**return** mv;

页面显示数据：

${cmsChannel.name } //实体类的变量

////////////////////////////////////////////////

普通的form表单提交数据：---不使用ajax；

<form action=*"*${path }*/saveComment.html"* method=*"post"*>

<textarea rows=*"8"* cols=*"93"* id=*"commentCon"* name=*"commentCon"* > </textarea>

<br /><br />

<input type=*"hidden"* name=*"aid"* id=*"aid"* value=*"*${cmsArticle.id }*"* />

<input type=*"submit"* value=*"提交"* class=*"login\_btn"* style="cursor:*pointer*;" />

</form>

需要js验证，这里使用button,先一个onclick，再提交；

后台，同样的controller等；