**BOFOWO开发手册**

**环境搭建：**

* maven3.0+
* eclipse
* jdk1.7+

1：svn仓库代码地址：

2: 构建项目

Cmd 进入项目跟目录，执行：mvn eclipse:eclipse

3: 打开eclipse ,

点击菜单 file import Existing Projects into Workspace 选择项目代码根目录点击完成。

4：eclipse中修改maven的setting.xml文件的路径。

点击菜单 window Maven User Settings update settings 选择maven路径的setting文件。

5：修改workspace编码 utf-8

**工程：**

1:lyun-service-api

2:lymall-management

3:lymall-web

**工程间依赖关系**

**代码开发Demo**

**返回页面**

新建java类 xxxController

类名上方添加注解@Controller

添加方法 public String index(){

return “index” //对应view 目录下的index.vm

}

方法名上方添加地址映射

@requestMaping(“index”)

代码片段：

Package com.example.controller

….

@Controller

public class DemoController extends BaseController{

….

@requestMapping(“index”);

public String index(ModelMap model,String param){ //model 需要返回页面的数据，param url请求对应的参数名。

model.put(“key”,”value”); //index页面使用$!{key} 即可获得value的内容。

return “index”;

}

…

}

**返回json**

新建java类 xxxController

类名上方添加注解@Controller

添加方法 public String getObjectJson(){

return “common/json” //对应view 目录下的index.vm

}

方法名上方添加地址映射

@requestMaping(“index”)

代码片段：

Package com.example.controller

….

@Controller

public class DemoController extends BaseController{

….

@requestMapping(“getObjectJson”); //对应的url映射

public String index(ModelMap model,String param){ //model 需要返回页面的数据，param url请求对应的参数名。

this.setLayout(LayoutType.empty); //设置使用空的模板。

model.put(“json”,”json string”);

return “common/json”;

}

…

}

**上传文件**

@RequestMapping("member-img-personal")

**public** String personal\_img(ModelMap model,**@RequestParam("userImage")MultipartFile file1**){ //上传的文件file文本域对应的名称。

String userName=CurrentUserUtil.*getCurrentUserName*();

UserInfoModel userInfo=userInfoService.getByUserName(userName);

String str=file1.getName();

String fileName1=path+"/"+userInfo.getUserName();

File parentFile1=**new** File(fileName1);

**try** {

**if**(!parentFile1.exists()&&!parentFile1.isDirectory())

parentFile1.mkdirs();

File HeadImgFile=**new** File(fileName1+"/"+file1.getOriginalFilename());

**if**(!HeadImgFile.exists())

HeadImgFile.createNewFile();

file1.transferTo(HeadImgFile);

userInfo.setPicture("/"+userInfo.getUserName()+"/"+file1.getOriginalFilename());

userInfoService.update(userInfo);

} **catch** (IllegalStateException e) {

//e.printStackTrace();

**if**(log.isErrorEnabled()){

log.error(userName+":"+e.getMessage());

}

} **catch** (IOException e) {

// e.printStackTrace();

**if**(log.isErrorEnabled()){

log.error(userName+":"+e.getMessage());

}

}

**return** "redirect:member-personalCenter-load.htm";

}

**数据路由**

**项目发布**

1：cmd进项项目根目录

2: 执行命令: mvn package

3: 命令执行完后，会在各个工程的target目录下生成相应的jar 或者 war包

4：登陆服务器控制台：[http://192.168.0.126:9990](http://192.168.0.126:9990/) (临时测试环境)

5：登陆，用户名密码：admin/123456

点击Deployement tab页面。进入发布页面

发布新的工程点击 Add 按钮，更新项目点击replace按钮，选择生成的war包点击。

6发布成功右上角messages会有相应的提示。

Note：1:如果长时间没有提示发布成功，需要重启jboss 服务器。

2:mvn package 打包过程中出现找不到类， 在eclipse中执行 project –》 clean 然后刷新项目。