**工程开发说明**

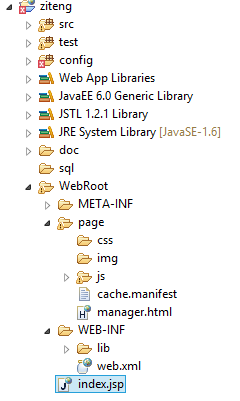
* **开发工具**

Myeclipse10.0, JDK1.6, Tomcat7, Mysql5.5.

* **开发框架**

SpringMVC + Mybatis + Mysql

* **工程目录**



src ： 源代码

test : 功能测试目录

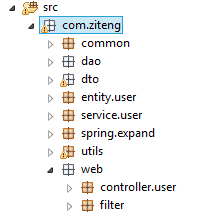
config : 配置目录，包spring, mybatis, properties

Page ： 存放前台展示页面

Css ： css样式目录

Img ： 图片资源

Js ： js脚本



Common : 定义公用常量包

Dao ： dao层

Dto ： 用于页面数据传输对象包

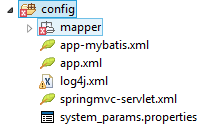
Entity ： 实体层

Service ： 服务层

Utils ： 常用工具包

Controller ： 控制器存放包

* **工程配置**



Config——各种配置。

App-mybatis.xml设置Mybatis的数据库源， dao组件设置。

App.xml 设置Spring 服务组件

Log4j.xml 设置日志

Springmvc-servlet.xml 设置Spring的mvc框架，视图解析，控制包，注解注入。

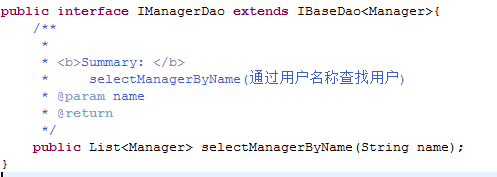
System\_params.properties 设置系统属性，如上传文件存放目录，系统连接地址等。

Mapper mybatis的dao层接口的实现文件

* **开发说明**

第一：Dao层的开发：每一个Dao都继承了IBaseDao类（里面提供了基本的增删查改方法）。当这些方法不够用事，在dao层的接口文件，添加相应的方法，然后在Myabtis的映射文件添加相应的sql处理方法（注意，这里不需要dao的接口实现）。如下例子。

Dao包下的user/IManagerDao



对应的mybatis映射文件方法 mapper/ManagerMapper.xml （可通过MybatisMapperUtil\_MYSQL类快速生成对于的实体类的mapper

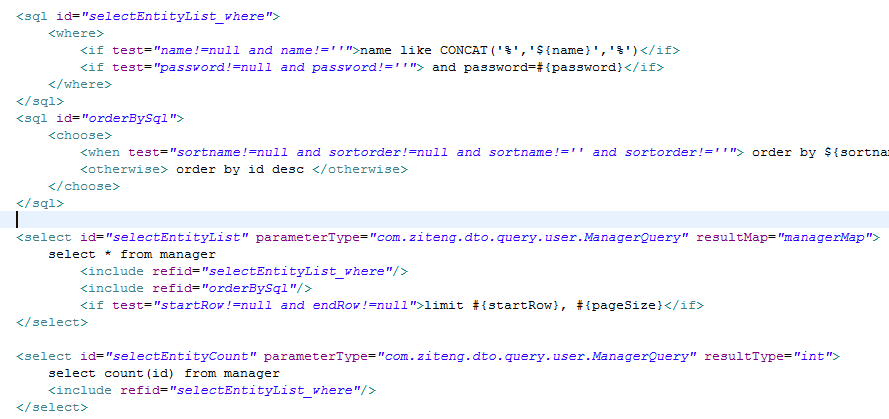


第二：分页查询。查询设置都在 com.sun.dto.query包下。

1) 查询类都继承base包下的Query类，默认该类是开启分页的（usePagination设置开启分页）。

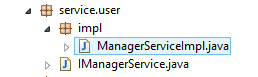
2) 对于一个分页查询类，1)添加接收前台发送参数的对于的属性，保持变量名一致，添加get，set方法。

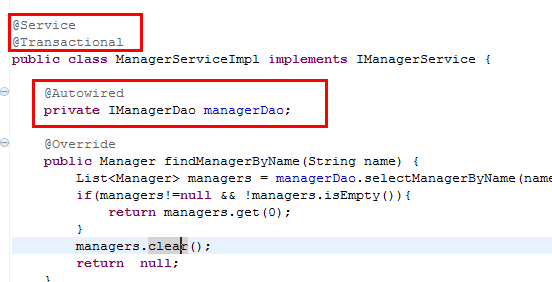
3) 写完查询类后，不需要写dao层的查询方法(IBaseDao已经定义了查询方法，不许在定义。) 在Mybatis映射文件中，添加查询方法，selectEntityList, selectEntityCount. 可参考ManagerMapper.xml或ManagerMapper.xml文件 。如下



第三： 服务开发

定义服务层接口，并实现该接口。 例如Manager为例





在ManagerServiceImpl 通过注解添加变量managerDao， 加上@Service, @Transactional注解（灰常重要，不能丢失）

第四： 控制层开发



第五： 前台访问

请求类型：get, post 。 请求参数类型：json， 返回值类型：json

例如：前台分页查询

***function*** query**(**pageNo**){**

$**(**"#tb\_body"**).**empty**();**

***var*** name **=** $**(**"#qy\_name"**).**val**();**

***var*** url **=** "/ziteng/user/managerQuery.do"**;**

***var*** param **=** **{**"name"**:**name**,**"pageNo"**:**pageNo**,** "pageSize"**:**20**};**

$**.**post**(**url**,**param**,*function*(**data**){**

***var*** json **=** eval**(**'('**+**data**+**')'**);**

***if*(**json**.**success**){**

pageSize **=** json**.**datas**.**pageSize**;**

pageNo **=** json**.**datas**.**pageNo**;**

total **=** json**.**datas**.**total**;**

***var*** managers **=** json**.**datas**.**managers**;**

currTotalPage **=** parseInt**((**total**+**pageSize**-**1**)/**pageSize**);**

$**(**'#pageInfo'**).**html**(**'当前记录'**+**managers**.**length**+**'条，总量' **+** total**+**'条，每页' **+** pageSize **+** '条'**);**

$**(**'#pageNo'**).**html**(**pageNo**+**'/' **+** currTotalPage**);**

***for*(*var*** i **=** 0**;** i **<** managers**.**length**;** i**++){**

***var*** manager **=** managers**[**i**];**

***var*** $row **=** $**(**'<tr>'**+**

'<td>'**+** manager**.**name**+**'</td>'**+**

'<td>'**+** manager**.**password**+** '</td>'**+**

'</tr>'**);**

$**(**'#tb\_body'**).**append**(**$row**);**

**}**

**}*else*** **{**

alert**(**json**.**msg**);**

**}**

**});**

**}**

* **插件推存**

Chrome插件：postman 请求插件。 在google里搜索，第一条记录就是



