

SpringMVC笔记

目录

第一章、HelloSpringMVC

什么是SpringMVC?

MVC在b/s系统下的应用:

SpringMVC框架

Model (模型)

Controller (控制器)

View (视图)

MVC的优点:

MVC的缺点:

控制器(controller)

模型(model)

视图(view)

执行流程:

Spring MVC运行原理

SpringMVC框架原理

前端控制器DispatcherServlet (不需要程序员开发)

处理器映射器HandlerMapping(不需要程序员开发)

处理器适配器HandlerAdapter

处理器Handler (需要程序员开发)

视图解析器View resolver(不需要程序员开发)

视图View (需要程序员开发)

如何解决WebMVC的缺点

SpringMVC常用注解:

@Controller: 注解一个类表示控制器, Spring MVC会自动扫描标注了这个注解的类。

@RequestMapping: 用于处理请求地址映射, 可以作用于类和方法上。

@RequestParam: 用于获取传入参数的值

@PathVariable: 用于定义路径参数值

@RequestBody

@ResponseBody: 作用于方法上, 可以将整个返回结果以某种格式返回, 如json或xml格式。

@CookieValue: 用于获取请求的Cookie值

@ModelAttribute: 用于把参数保存到model中

@SessionAttributes

案例-HelloSpringMVC使用BeanNameUrlHandlerMapping

案例-HelloSpringMVC使用注解

案例-参数传递从视图到控制器

案例-参数传递从控制器到视图-使用ModelAndView

案例-参数传递从控制器向视图-使用Model对象

案例-传递参数从控制器到视图-使用Map

第二章、SSM整合-项目申报系统

项目结构:

数据库表:

页面效果

环境搭建

1.创建Maven项目 (打成war包)

2.配置pom.xml引入项目依赖jar包

实体类: `com.yuntu.curd.bean`

控制器: `com.yuntu.curd.controller`

数据访问层: `com.yuntu.curd.dao`

业务逻辑层: `com.yuntu.curd.service`

测试: `com.yuntu.curd.test`

`src/main/resources`

`web.xml`

`index.jsp`

`mbg.xml`

第一章、HelloSpringMVC

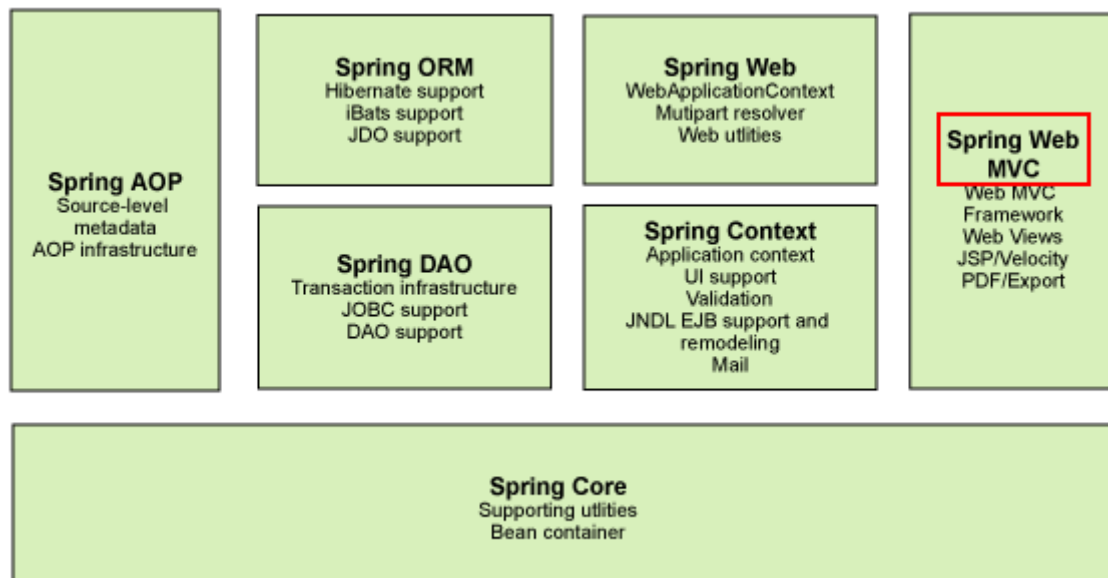
什么是SpringMVC?

SpringMVC是spring框架的一个模块，springmvc和spring无需通过中间整合层进行整合。

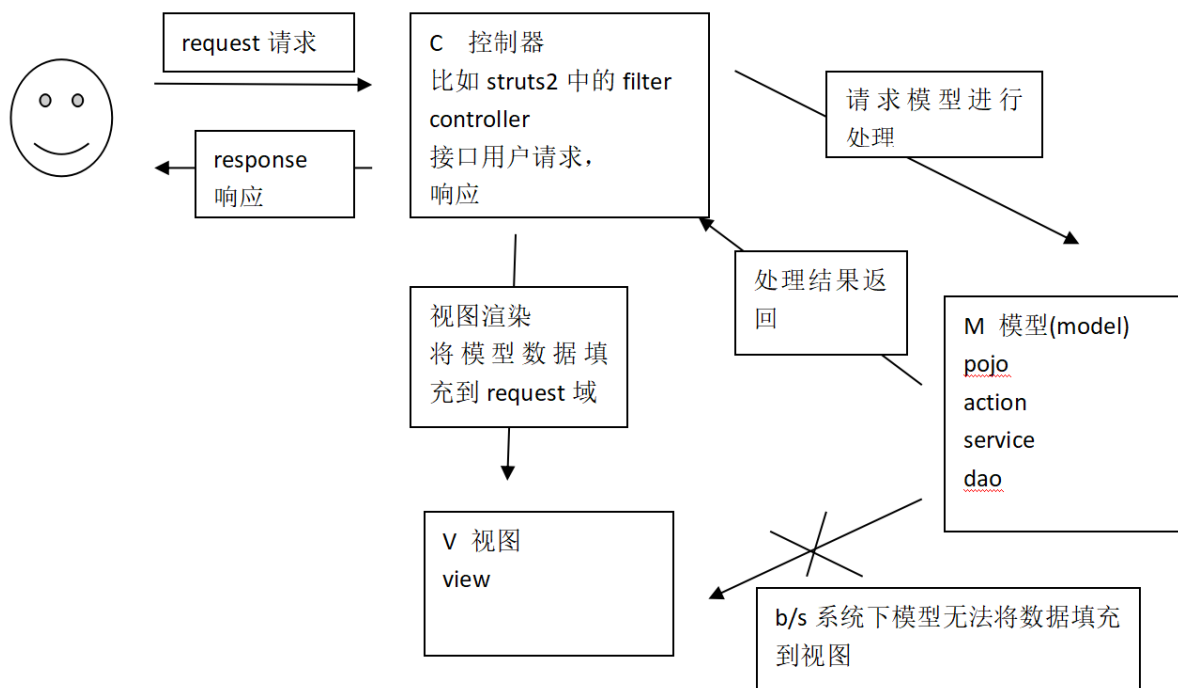
springmvc是一个基于mvc的web框架。

提供了构建 Web 应用程序的全功能 MVC 模块。使用 Spring 可插入的 MVC 架构，从而在使用Spring进行WEB开发时，可以选择使用Spring的Spring MVC框架或集成



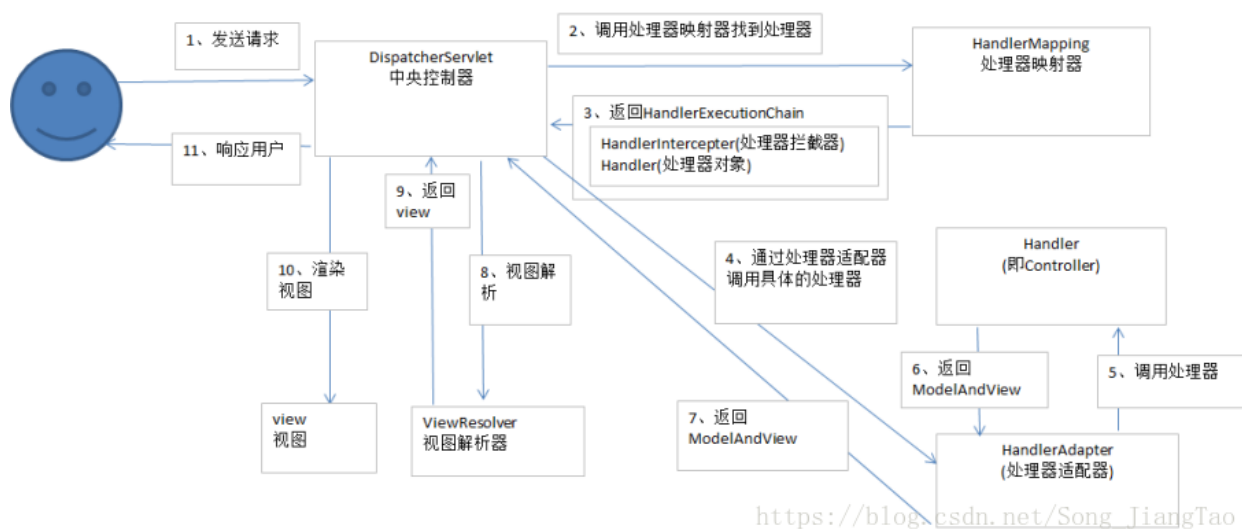


MVC在b/s系统下的应用:



https://blog.csdn.net/Song_JiangTao

SpringMVC框架



Model (模型)

- 封装应用状态（封装应用数据）
- 响应状态查询（对数据进行增删改查）
- 暴露应用的功能（暴露接口<public>）

Controller (控制器)

- 验证HTTP请求的数据（收集组织数据）
- 将用户数据与模型的更新相映射（调用逻辑层）
- 选择用于响应的视图（选择下一个界面）

View (视图)

- 产生HTML响应（展示数据）
- 请求模型的更新（人机交互）
- 提供HTML form用于用户请求（收集参数，调用逻辑层api）

MVC的优点：

- 低耦合性：视图层和业务层分离
- 高重用性和可适用性
- 可维护性
- 有利于软件工程化管理
- 提高软件的健壮性

MVC的缺点:

工作量大，增加工作的复杂性，MVC不适合小型甚至中等规模的应用，程序将状态查询封装在Model中，使Model过于臃肿。

如果没有MVC设计模式。程序间的各层之间依赖非常强，耦合度高。严重违背了高内聚低耦合的设计原则。而WebMVC将控制逻辑和功能处理，模型和视图进行了分离。降低耦合但是WebMVC也有严重的缺点:

控制器(controller)

- 1.控制逻辑较为复杂，而且每个模块都需要一个控制器，（可能每一个页面都需要一个控制器）
- 2.请求参数到模型的封装麻烦。
- 3.视图的选择严重依赖于Servlet API，这样很能甚至不可能更换视图
- 4.给视图传入模型数据 也依赖于Servlet API.如果要更换视图那么技术也要改变。

模型(model)

使用JavaBean组件，（域模型层+业务层+持久层）导致JavaBean组件类庞大。不利于开发人员管理

视图(view)

被绑定于Jsp,很难更换视图

执行流程:

- 1、用户发送请求至前端控制器DispatcherServlet
- 2、DispatcherServlet收到请求调用HandlerMapping处理器映射器。
- 3、处理器映射器根据请求url找到具体的处理器，生成处理器对象及处理器拦截器(如果有则生成)一并返回给DispatcherServlet。
- 4、DispatcherServlet通过HandlerAdapter处理器适配器调用处理器
- 5、执行处理器(Controller，也叫后端控制器)。

- 6、Controller执行完成返回ModelAndView
- 7、HandlerAdapter将controller执行结果ModelAndView返回给DispatcherServlet
- 8、DispatcherServlet将ModelAndView传给ViewResolver视图解析器
- 9、ViewResolver解析后返回具体View
- 10、DispatcherServlet对View进行渲染视图（即将模型数据填充至视图中）。
- 11、DispatcherServlet响应用户

Spring MVC运行原理

整个处理过程从一个HTTP请求开始：

- 1.Tomcat在启动时加载解析web.xml,找到spring mvc的前端总控制器DispatcherServlet,并且通过DispatcherServlet来加载相关的配置文件信息。
- 2.DispatcherServlet接收到客户端请求，找到对应HandlerMapping，根据映射规则，找到对应的处理器（Handler）。
- 3.调用相应处理器中的处理方法，处理该请求后，会返回一个ModelAndView。
- 4.DispatcherServlet根据得到的ModelAndView中的视图对象，找到一个合适的ViewResolver（视图解析器），根据视图解析器的配置，DispatcherServlet将要显示的数据传给对应的视图，最后显示给用户。

SpringMVC框架原理

前端控制器DispatcherServlet（不需要程序员开发）

作用：

接收请求，响应结果，相当于转发器，中央处理器。有了DispatcherServlet减少了其它组件之间的耦合度。

处理器映射器HandlerMapping(不需要程序员开发)

作用：根据请求的url查找Handler

处理器适配器HandlerAdapter

作用：按照特定规则（HandlerAdapter要求的规则）去执行Handler

处理器Handler (需要程序员开发)

注意：编写Handler时按照HandlerAdapter的要求去做，这样适配器才可以去正确执行Handler

视图解析器View resolver(不需要程序员开发)

作用：进行视图解析，根据逻辑视图名解析成真正的视图（view）

视图View (需要程序员开发)

View是一个接口，实现类支持不同的View类型（jsp、freemarker、pdf...）

如何解决WebMVC的缺点

从服务到工作者模式：

服务到工作者：Front Controller + Application Controller + Page Controller + Context即，前端控制器+应用控制器+页面控制器（也有称其为动作）+上下文

前端控制器：

负责为表现层提供统一访问点，从而避免WebMVC中出现的重复的控制逻辑,可以为多个请求提供共用的逻辑（如准备上下文等等），将选择具体视图和具体的功能处理分离。

应用控制器：

前端控制器分离选择具体视图和具体的功能处理之后，需要有人来管理，应用控制器就是用来选择具体视图技术（视图的管理）和具体的功能处理（页面控制器/命令对象/动作管理），一种策略设计模式的应用，可以很容易的切换视图/页面控制器，相互不产生影响

页面控制器/动作/处理器：

功能处理代码，收集参数、封装参数到模型，转调业务对象处理模型，返回逻辑视图名交给前端控制器（和具体的视图技术解耦），由前端控制器委托给应用控制器选择具体的视图来展示，可以是命令设计模式的实现。页面控制器也被称为处理器或动作。

上下文

WebMVC中为视图准备要展示的数据模型，我们直接放在request中（Servlet API相关），有了上下文之后，我们就可以将相关数据放置在上下文，从而与协议无关（如Servlet API）的访问/设置模型数据，一般通过ThreadLocal模式实现。

SpringMVC常用注解：

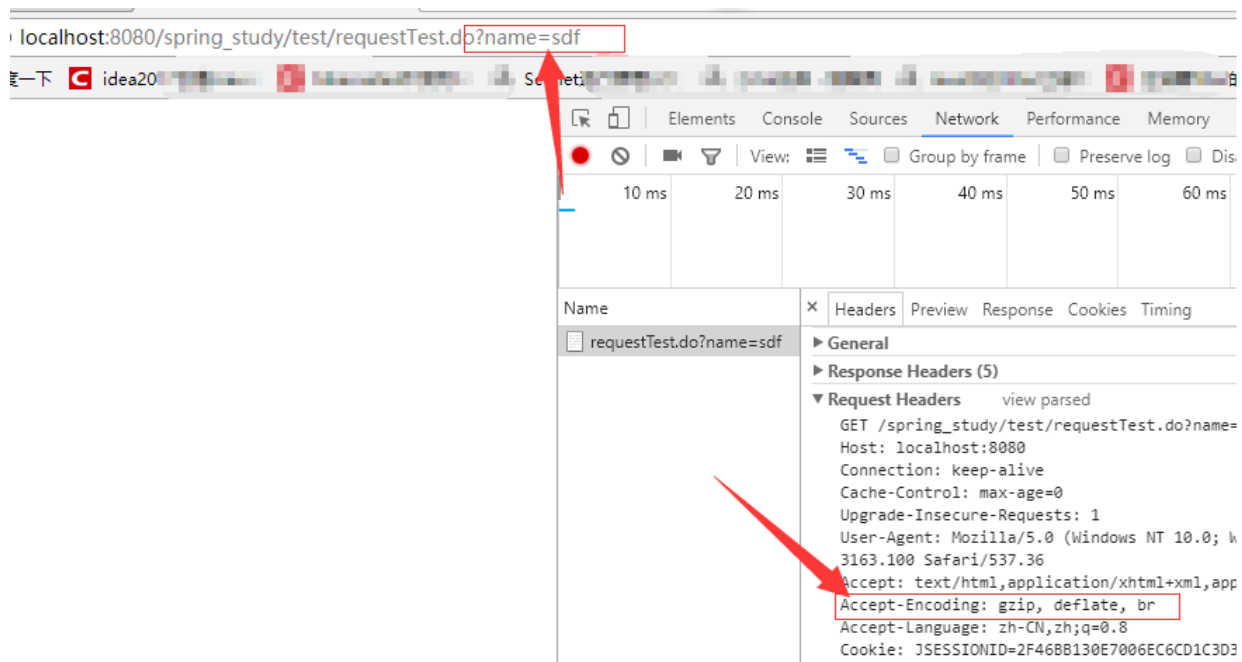
@Controller：注解一个类表示控制器，Spring MVC会自动扫描标注了这个注解的类。

@RequestMapping：用于处理请求地址映射，可以作用于类和方法上。

- value：定义request请求的映射地址
- method：定义request地址请求的方式，包括【GET, POST, HEAD, OPTIONS, PUT, PATCH, DELETE, TRACE.】默认接受get请求，如果请求方式和定义的方式不一样则请求无法成功。
- params：定义request请求中必须包含的参数值。
- headers：定义request请求中必须包含某些指定的请求头，如：RequestMapping(value = "/something", headers = "content-type=text/*")说明请求中必须要包含"text/html", "text/plain"这中类型的Content-type头，才是一个匹配的请求。
- consumes：定义请求提交内容的类型。
- produces：指定返回的内容类型，仅当request请求头中的(Accept)类型中包含该指定类型才返回

```
1 @RequestMapping(value="/requestTest.do",params = {"name=sdf"},headers = {"Accept-Encoding=gzip, deflate, br"},method = RequestMethod.GET)
2 public String getIndex(){
3     System.out.println("请求成功");
4     return "index";
5 }
```

上面代码表示请求的方式为GET请求，请求参数必须包含name=sdf这一参数，然后请求头中必须有 Accept-Encoding=gzip, deflate, br这个类型头。



这样通过注解就能对一个请求进行约束了。

@RequestParam: 用于获取传入参数的值

- value: 参数的名称
- required: 定义该传入参数是否必须, 默认为true, (和 @RequestMapping的params属性有点类似)

```
1 @RequestMapping("/requestParams1.do")
2 public String requestParams1(@RequestParam(required = false) String name){
3     System.out.println("name = "+name);
4     return "index";
5 }
6 @RequestMapping("/requestParams2.do")
7 public String requestParams2(@RequestParam(value = "name", required = false) String names){
8     System.out.println("name = "+names);
9     return "index";
10 }
```

两种传入参方式是一样的, 显示声明value的名称时, 入参参数名和value一样, 没有显示声明的话, 像第一种方式声明的, 入参参数名和函数参数变量名一样。

@PathVariable: 用于定义路径参数值

- value: 参数的名称

- required: 定义传入参数是否为必须值

```
1 @RequestMapping("/{myname}/pathVariable2.do")
2 public String pathVariable2(@PathVariable(value = "myname") String name){
3     System.out.println("myname = "+name);
4     return "index";
5 }
```

这个路径声明了{myname}作为路径参数，那么这一段路径将为任意值，@PathVariable将可以根据value获取路径的值。

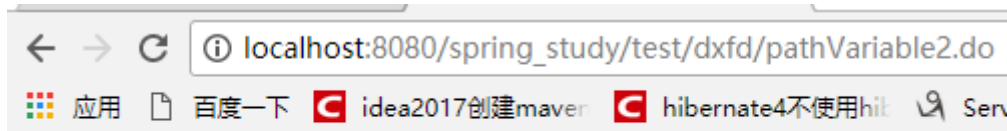
@RequestBody

放在参数前，表示参数从request body中获取，而不是从地址栏获取，所以这肯定是接收一个POST请求的非a=b格式的数据，即 Content-Type不为 application/x-www-form-urlencoded类型的内容。

@ResponseBody: 作用于方法上，可以将整个返回结果以某种格式返回，如json或xml格式。

```
1 @RequestMapping("/{myname}/pathVariable2.do")
2 @ResponseBody
3 public String pathVariable2(@PathVariable(value = "myname") String name){
4     System.out.println("myname = "+name);
5     return "index";
6 }
```

和上一段例子一样，只是添加了@ResponseBody标签，然后返回结果为：



index

它返回的不是一个页面，而是把字符串“index”直接在页面打印出来了，这其实和如下代码时类似的。

```
1 PrintWriter out = resp.getWriter();
2 out.print("index");
3 out.flush();
```

@CookieValue: 用于获取请求的Cookie值

```
1 @RequestMapping("/requestParams.do")
2 public String requestParams(@CookieValue("JSESSIONID") String cookie){
3     return "index";
4 }
```

@ModelAttribute: 用于把参数保存到model中

@ModelAttribute: 用于把参数保存到model中，可以注解方法或参数，注解在方法上的时候，该方法将在处理器方法执行之前执行，然后把返回的对象存放在 session（前提时要有@SessionAttributes注解）或模型属性中，@ModelAttribute("attributeName") 在标记方法的时候指定，若未指定，则使用返回类型的类名称（首字母小写）作为属性名称。

```
1 @ModelAttribute("user")
2 public UserEntity getUser(){
3     UserEntity userEntityr = new UserEntity();
4     userEntityr.setUsername("asdf");
5     return userEntityr;
6 }
7
8 @RequestMapping("/modelTest.do")
9 public String getUsers(@ModelAttribute("user") UserEntity user)
10 {
11     System.out.println(user.getUsername());
12     return "/index";
13 }
```

如上代码中，使用了@ModelAttribute("user")注解，在执行控制器前执行，然后将生成一个名称为user的model数据，在控制器中我们通过注解在参数上的@ModelAttribute获取参数，然后将model应用到控制器中，在jsp页面中我们同样可以使用它，

```
1 <body>
2     ${user.username}
3 </body>
```

@SessionAttributes

默认情况下Spring MVC将模型中的数据存储到request域中。当一个请求结束后，数据就失效了。如果要跨页面使用。那么需要使用到session。而@SessionAttributes注解就可以使得模型中的数据存储一份到session域中。配合@ModelAttribute("user")使用的时候,会将对应的名称的model值存到session中,

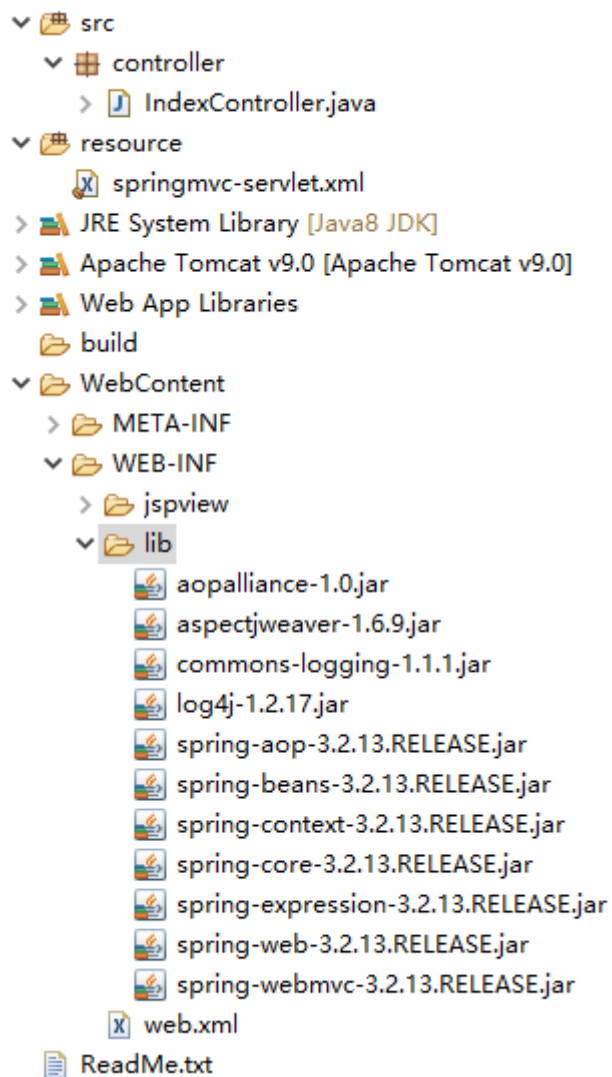
```
1 @Controller
2 @RequestMapping("/test")
3 @SessionAttributes(value = {"user", "test1"})
4 public class LoginController{
5     @ModelAttribute("user")
6     public UserEntity getUser(){
7         UserEntity userEntityr = new UserEntity();
8         userEntityr.setUsername("asdf");
9         return userEntityr;
10    }
11
12    @RequestMapping("/modelTest.do")
13    public String getUsers(@ModelAttribute("user") UserEntity user
14        ,HttpSession session){
15        System.out.println(user.getUsername());
16        System.out.println(session.getAttribute("user"));
17        return "/index";
18    }
19 }
```

结合上一个例子的代码，加了@SessionAttributes注解，然后请求了两次，第一次session中不存在属性名为user的值，第二次请求的时候发现session中又有了，这是因为，这是因为第一次请求时，model数据还未保存到session中请求结束返回的时候才保存，在第二次请求的时候已经可以获取上一次的model了，

注意：@ModelAttribute("user") UserEntity user获取注解内容的时候，会先查询session中是否有对应的属性值，没有才去查询Model。

案例-HelloSpringMVC使用BeanNameUrlHandlerMapping

项目结构：



IndexController.java

```
1 package controller;
2
3 import javax.servlet.http.HttpServletRequest;
4 import javax.servlet.http.HttpServletResponse;
5 import org.springframework.web.servlet.ModelAndView;
6 import org.springframework.web.servlet.mvc.AbstractController;
7
8 public class IndexController extends AbstractController {
9
10     @Override
11     protected ModelAndView handleRequestInternal(HttpServletRequest arg0, HttpServletResponse arg1) throws Exception {
12         System.out.println("hello.springmvc的controller");
13         //返回视图模型，名字为index的
```

```
14  return new ModelAndView("index");
15  }
16
17  }
18
```

springmvc-servlet.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4  xmlns:p="http://www.springframework.org/schema/p"
5  xmlns:context="http://www.springframework.org/schema/context"
6  xmlns:mvc="http://www.springframework.org/schema/mvc"
7  xmlns:aop="http://www.springframework.org/schema/aop"
8  xmlns:tx="http://www.springframework.org/schema/tx"
9  xmlns:task="http://www.springframework.org/schema/task"
10  xsi:schemaLocation="
11  http://www.springframework.org/schema/beans
12  http://www.springframework.org/schema/beans/spring-beans-3.2.x
13  sd
14  http://www.springframework.org/schema/context
15  http://www.springframework.org/schema/context/spring-context-
16  3.2.xsd
17  http://www.springframework.org/schema/mvc
18  http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd
19  http://www.springframework.org/schema/aop
20  http://www.springframework.org/schema/aop/spring-aop-3.2.xsd
21  http://www.springframework.org/schema/tx
22  http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
23  http://www.springframework.org/schema/task
24  http://www.springframework.org/schema/task/spring-task-3.2.xs
25  d">
26
27  <!-- 通过BeanNameUrlHandlerMapping方式完成请求与Controller之间的
28  映射关系，若有多个请求
```

```

25 就在springmvc-servlet.xml文件中配置多个映射关系，同时要建立多个javaBean作为控制器进行处
26 理的处理
27 缺点：
28 配置过多，controller过多，业务复杂的话，是不合理的
29 -->
30 <!-- 页面的转化对象，用来配置访问名字 -->
31 <bean name="/index.html" class="controller.IndexController">
32 </bean>
33 <!-- 完成视图的对应处理-->
34 <!-- 处理路径问题
35 对转向页面的路径解析：
36 prefix: 前缀
37 suffix: 后缀
38 -->
39 <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
40 <property name="prefix" value="/WEB-INF/jspview/"></property>
41 <property name="suffix" value=".jsp"></property>
42 </bean>
43 </beans>

```

index.jsp

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <h1>HelloSpringMVC</h1>
11 </body>
12 </html>

```


WEB-INF/web.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
3   <display-name>springmvc-class1-demo1-hellospringmvc使用BeanNameHandlerMapping</display-name>
4
5   <servlet>
6     <servlet-name>springmvc</servlet-name>
7     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
8     <!-- 初始参数 -->
9     <init-param>
10      <!-- 初始化springmvc上下文配置的路径 -->
11      <param-name>contextConfigLocation</param-name>
12      <param-value>classpath:springmvc-servlet.xml</param-value>
13    </init-param>
14
15    <!-- 是否应该在启动的时候就加载当前这个servlet，是一个整数；
16     数值表示servlet应该被载入的顺序，值越小则该servlet优先级越高，应用启动是就越先加载
17     -->
18    <load-on-startup>1</load-on-startup>
19  </servlet>
20  <servlet-mapping>
21    <servlet-name>springmvc</servlet-name>
22    <url-pattern>/</url-pattern>
23  </servlet-mapping>
24
25 </web-app>
```

程序运行结果：

HelloSpringMVC

案例-HelloSpringMVC使用注解

项目结构：

- src
 - controller
 - IndexController.java
 - resource
 - log4j.properties
 - springmvc-servlet.xml
 - JRE System Library [Java8 JDK]
 - Apache Tomcat v9.0 [Apache Tomcat v9.0]
 - Web App Libraries
 - build
- WebContent
 - META-INF
 - WEB-INF
 - jspview
 - index.jsp
 - lib
 - aopalliance-1.0.jar
 - aspectjweaver-1.6.9.jar
 - commons-logging-1.1.1.jar
 - log4j-1.2.17.jar
 - spring-aop-3.2.13.RELEASE.jar
 - spring-beans-3.2.13.RELEASE.jar
 - spring-context-3.2.13.RELEASE.jar
 - spring-core-3.2.13.RELEASE.jar
 - spring-expression-3.2.13.RELEASE.jar
 - spring-web-3.2.13.RELEASE.jar
 - spring-webmvc-3.2.13.RELEASE.jar
 - web.xml
- ReadMe.txt

IndexController.java

```
1 package controller;
```

```

2
3 import org.apache.log4j.Logger;
4 import org.springframework.stereotype.Controller;
5 import org.springframework.web.bind.annotation.RequestMapping;
6
7 //当前类为控制器类（进行页面的映射）
8 @Controller
9 @RequestMapping("/login")
10 public class IndexController{
11     private Logger log=Logger.getLogger(IndexController.class);
12
13     //@RequestMapping(请求的映射路径):表示用哪一个url来对应此方法执行
14     @RequestMapping("/indexview")
15     public String index() {
16         log.info("hello.springmvc,注解形式");
17         return "index";
18     }
19 }
20

```

springmvc-servlet.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:context="http://www.springframework.org/schema/context"
6     xmlns:mvc="http://www.springframework.org/schema/mvc"
7     xmlns:aop="http://www.springframework.org/schema/aop"
8     xmlns:tx="http://www.springframework.org/schema/tx"
9     xmlns:task="http://www.springframework.org/schema/task"
10     xsi:schemaLocation="
11         http://www.springframework.org/schema/beans
12         http://www.springframework.org/schema/beans/spring-beans-3.2.x
13         sd
14         http://www.springframework.org/schema/context

```

```

14 http://www.springframework.org/schema/context/spring-context-
   3.2.xsd
15 http://www.springframework.org/schema/mvc
16 http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd
17 http://www.springframework.org/schema/aop
18 http://www.springframework.org/schema/aop/spring-aop-3.2.xsd
19 http://www.springframework.org/schema/tx
20 http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
21 http://www.springframework.org/schema/task
22 http://www.springframework.org/schema/task/spring-task-3.2.xs
   d">
23
24 <!-- 扫描注解的控制器包 -->
25 <context:component-scan base-package="controller"></context:co
   mponent-scan>
26 <!-- 一键式配置，通过注解方式进行springmvc开发 -->
27 <mvc:annotation-driven></mvc:annotation-driven>
28
29 <!-- 完成视图的对应处理-->
30 <!-- 处理路径问题
31 对转向页面的路径解析：
32 prefix: 前缀
33 suffix: 后缀
34 -->
35 <bean class="org.springframework.web.servlet.view.InternalReso
   urceViewResolver">
36 <property name="prefix" value="/WEB-INF/jspview/"></property>
37 <property name="suffix" value=".jsp"></property>
38 </bean>
39 </beans>

```

WebContent/WEB-INF/jspview/index.jsp

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>

```

```

5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <h1>HelloSpringMVC,注解形式</h1>
11 </body>
12 </html>

```

WebContent/WEB-INF/web.xml

```

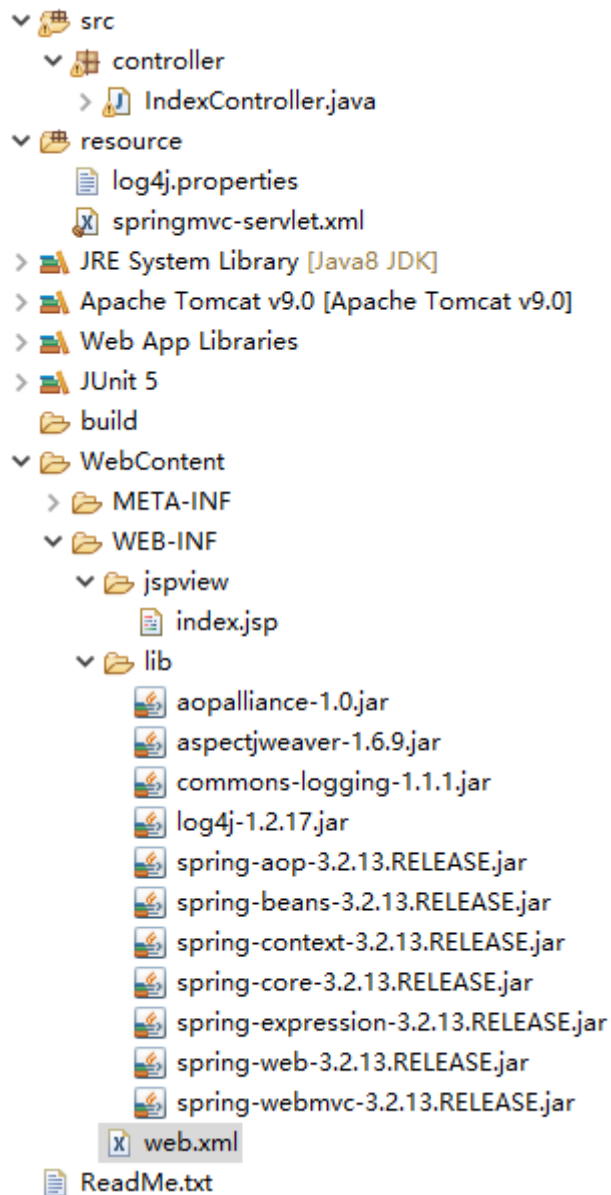
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" version="4.0">
3 <display-name>springmvc-class1-demo2-hellospringmvc使用注解</display-name>
4 <welcome-file-list>
5 <welcome-file>index.html</welcome-file>
6 <welcome-file>index.htm</welcome-file>
7 <welcome-file>index.jsp</welcome-file>
8 <welcome-file>default.html</welcome-file>
9 <welcome-file>default.htm</welcome-file>
10 <welcome-file>default.jsp</welcome-file>
11 </welcome-file-list>
12
13 <servlet>
14 <servlet-name>springmvc</servlet-name>
15 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
16 <!-- 初始参数 -->
17 <init-param>
18 <!-- 初始化springmvc上下文配置的路径 -->
19 <param-name>contextConfigLocation</param-name>
20 <param-value>classpath:springmvc-servlet.xml</param-value>
21 </init-param>
22

```

```
23  <!-- 是否应该在启动的时候就加载当前这个servlet，是一个整数；
24  数值表示servlet应该被载入的顺序，值越小则该servlet优先级越高，应用
    启动是就越先加载
25  -->
26  <load-on-startup>1</load-on-startup>
27  </servlet>
28  <servlet-mapping>
29  <servlet-name>springmvc</servlet-name>
30  <url-pattern>/</url-pattern>
31  </servlet-mapping>
32  </web-app>
```

案例-参数传递从视图到控制器

项目结构：



IndexController.java

```
1 package controller;
2
3 import org.apache.log4j.Logger;
4 import org.springframework.stereotype.Controller;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestMethod;
7 import org.springframework.web.bind.annotation.RequestParam;
8
9 //当前类为控制器类（进行页面的映射）
10 @Controller
11 public class IndexController{
12     private Logger log=Logger.getLogger(IndexController.class);
```

```
13
14 // @RequestMapping(请求的映射路径):表示用哪一个url来对应此方法执行
15 @RequestMapping("/indexview")
16 public String index() {
17     log.info("hello.springmvc,注解形式");
18     return "index";
19 }
20
21
22 /**
23  * 参数传递: view to controller
24  * 视图层页面的请求路径向控制器中指定的方法跳转并携带参数
25  * 控制器需要对应接收携带的信息
26  */
27 // @RequestMapping("/welcome")
28 // @RequestParam: 请求中携带的参数信息
29 // (如果页面携带的参数名称不是username, 可以在@RequestParam("携带
    参数名称")指明携带参数的名称)
30 // public String welcome(@RequestParam("uname") String usernam
    e) {
31 //     log.info("接收的内容: "+username);
32 //     return "index";
33 // }
34
35 //方法的请求映射结果, 任意一个均可访问
36 // @RequestMapping({"/welcome","/"})
37 // @RequestParam: 请求中携带的参数信息
38 // (如果页面携带的参数名称不是username, 可以在@RequestParam("携带
    参数名称")指明携带参数的名称)
39 // value: 参数的名称
40 // required: 定义该传入参数是否必须, 默认为true
41 // public String welcome(@RequestParam(value="uname",required=f
    alse) String username) {
42 //     log.info("接收的内容: "+username);
43 //     return "index";
44 // }
45
```



```

46
47  /**
48   * 请求映射配置中，直接添加设置请求方法和所携带参数
49   * value: 定义request请求的映射地址
50   * method: 定义地request址请求的方式，包括
51   * 【GET, POST, HEAD, OPTIONS, PUT, PATCH, DELETE, TRACE.】默认
52   * 接受get请求，如果请求方式和定义的方式不一样则请求无法成功。
53   * params: 定义request请求中必须包含的参数值。
54   */
55
56   @RequestMapping(value="/welcome",method=RequestMethod.GET)
57   public String welcome(String username) {
58       log.info("接收的内容: "+username);
59       return "index";
60   }
61 }

```

springmvc-servlet.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:context="http://www.springframework.org/schema/context"
6     xmlns:mvc="http://www.springframework.org/schema/mvc"
7     xmlns:aop="http://www.springframework.org/schema/aop"
8     xmlns:tx="http://www.springframework.org/schema/tx"
9     xmlns:task="http://www.springframework.org/schema/task"
10    xsi:schemaLocation="
11        http://www.springframework.org/schema/beans
12        http://www.springframework.org/schema/beans/spring-beans-3.2.x
13        sd
14        http://www.springframework.org/schema/context
15        http://www.springframework.org/schema/context/spring-context-
16        3.2.xsd
17        http://www.springframework.org/schema/mvc

```

```

16 http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd
17 http://www.springframework.org/schema/aop
18 http://www.springframework.org/schema/aop/spring-aop-3.2.xsd
19 http://www.springframework.org/schema/tx
20 http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
21 http://www.springframework.org/schema/task
22 http://www.springframework.org/schema/task/spring-task-3.2.xs
  d">
23
24 <!-- 扫描注解的控制器包 -->
25 <context:component-scan base-package="controller"></context:co
  mponent-scan>
26 <!-- 一键式配置，通过注解方式进行springmvc开发 -->
27 <mvc:annotation-driven></mvc:annotation-driven>
28
29 <!-- 完成视图的对应处理-->
30 <!-- 处理路径问题
31 对转向页面的路径解析；
32 prefix: 前缀
33 suffix: 后缀
34 -->
35 <bean class="org.springframework.web.servlet.view.InternalReso
  urceViewResolver">
36 <property name="prefix" value="/WEB-INF/jspview/"></property>
37 <property name="suffix" value=".jsp"></property>
38 </bean>
39 </beans>

```

WebContent/WEB-INF/jspview/index.jsp

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>

```

```

8 </head>
9 <body>
10 <h1>HelloSpringMVC</h1>
11 </body>
12 </html>

```

web.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
3   <display-name>springmvc-class1-demo3-参数传递从视图到控制器</display-name>
4   <welcome-file-list>
5     <welcome-file>index.html</welcome-file>
6     <welcome-file>index.htm</welcome-file>
7     <welcome-file>index.jsp</welcome-file>
8     <welcome-file>default.html</welcome-file>
9     <welcome-file>default.htm</welcome-file>
10    <welcome-file>default.jsp</welcome-file>
11  </welcome-file-list>
12
13  <servlet>
14    <servlet-name>springmvc</servlet-name>
15    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
16    <!-- 初始参数 -->
17    <init-param>
18      <!-- 初始化springmvc上下文配置的路径 -->
19      <param-name>contextConfigLocation</param-name>
20      <param-value>classpath:springmvc-servlet.xml</param-value>
21    </init-param>
22
23    <!-- 是否应该在启动的时候就加载当前这个servlet，是一个整数；
24    数值表示servlet应该被载入的顺序，值越小则该servlet优先级越高，应用启动是就越先加载

```

```

25  -->
26  <load-on-startup>1</load-on-startup>
27  </servlet>
28  <servlet-mapping>
29  <servlet-name>springmvc</servlet-name>
30  <url-pattern>/</url-pattern>
31  </servlet-mapping>
32  </web-app>

```

案例-参数传递从控制器到视图-使用ModelAndView

此处仅展示部分代码，其他相关代码，见上一案例

IndexController.java

```

1  package controller;
2
3  import org.apache.log4j.Logger;
4  import org.springframework.stereotype.Controller;
5  import org.springframework.web.bind.annotation.RequestMapping;
6  import org.springframework.web.bind.annotation.RequestMethod;
7  import org.springframework.web.bind.annotation.RequestParam;
8  import org.springframework.web.servlet.ModelAndView;
9
10 //当前类为控制器类（进行页面的映射）
11 @Controller
12 public class IndexController{
13     private Logger log=Logger.getLogger(IndexController.class);
14
15     //@RequestMapping(请求的映射路径):表示用哪一个url来对应此方法执行
16     @RequestMapping("/indexview")
17     public String index() {
18         log.info("hello.springmvc,注解形式");
19         return "index";
20     }
21
22     /*

```

```

23  * 参数传递: controller to view -ModelAndView
24  * 从控制器向视图中传递数据, 以返回后ModelAndView的方式
25  * 既包含视图信息, 又包含数据信息
26  */
27  @RequestMapping("/index1")
28  public ModelAndView index1(String username) {
29      log.info("控制器中的数据: "+username);
30      //创建视图信息的和数据信息的对象
31      ModelAndView mView = new ModelAndView();
32      //添加数据到mView中
33      mView.addObject("username",username);
34      //添加视图mView中
35      mView.setViewName("index");
36      //返回
37      return mView;
38  }
39  }
40

```

WebContent/WEB-INF/jspview/index.jsp

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="UTF-8">
7  <title>Insert title here</title>
8  </head>
9  <body>
10   <h1>HelloSpringMVC + ${username }</h1>
11 </body>
12 </html>

```

案例-参数传递从控制器向视图-使用Model对象

User.java

```
1 package entity;
2
3 public class User {
4     private Integer id;
5     private String uName;
6     private Integer uAge;
7
8     public User(Integer id, String uName, Integer uAge) {
9         super();
10        this.id = id;
11        this.uName = uName;
12        this.uAge = uAge;
13    }
14    public User() {
15        super();
16    }
17
18    public Integer getId() {
19        return id;
20    }
21    public void setId(Integer id) {
22        this.id = id;
23    }
24    public String getuName() {
25        return uName;
26    }
27    public void setuName(String uName) {
28        this.uName = uName;
29    }
30    public Integer getuAge() {
31        return uAge;
32    }
33    public void setuAge(Integer uAge) {
34        this.uAge = uAge;
```

```
35 }  
36 }  
37
```

IndexController.java

```
1 package controller;  
2  
3 import org.apache.log4j.Logger;  
4 import org.springframework.stereotype.Controller;  
5 import org.springframework.ui.Model;  
6 import org.springframework.web.bind.annotation.RequestMapping;  
7 import org.springframework.web.servlet.ModelAndView;  
8  
9 import entity.User;  
10  
11 //当前类为控制器类（进行页面的映射）  
12 @Controller  
13 public class IndexController{  
14     private Logger log=Logger.getLogger(IndexController.class);  
15  
16     //RequestMapping(请求的映射路径):表示用哪一个url来对应此方法执行  
17     @RequestMapping("/indexview")  
18     public String index() {  
19         log.info("hello.springmvc,注解形式");  
20         return "index";  
21     }  
22  
23     /*  
24     * 参数传递: controller to view -ModelAndView  
25     * 从控制器向视图中传递数据，以返回后ModelAndView的方式  
26     * 既包含视图信息，又包含数据信息  
27     */  
28     @RequestMapping("/index1")  
29     public ModelAndView index1(String username) {  
30         log.info("控制器中的数据: "+username);  
31     }  
32 }  
33
```

```
31 //创建视图信息的和数据信息的对象
32 ModelAndView mView = new ModelAndView();
33 //添加数据到mView中
34 mView.addObject("username",username);
35 //添加视图mView中
36 mView.setViewName("index");
37 //返回
38 return mView;
39 }
40
41 /**
42  * 参数传递 controller to view
43  * 控制器处理方法返回的是String（逻辑视图名称），模型数据信息放入Model中;
44  *
45  * @param username
46  * @param model
47  * @return
48  */
49 @RequestMapping("/index2")
50 public String index2(String username,Model model) {
51     log.info("控制器中的数据: "+username);
52
53     //向Model中放入属性字符串
54     model.addAttribute("currentusername",username+"字符串");
55     //直接放入字符串
56     model.addAttribute(username);
57     //_____
58     User user = new User();
59     user.setName(username);
60     //向Model中传递对象
61     model.addAttribute("currentuser",user);
62     //放入对象
63     model.addAttribute(user);
64
65     return "index";
```



```
66  }
67  }
68
```

WebContent/WEB-INF/jspview/index.jsp

```
1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2    pageEncoding="UTF-8"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="UTF-8">
7  <title>Insert title here</title>
8  </head>
9  <body>
10   <h1>名字 (key为currentusername) + ${currentusername }</h1>
11   <h1>名字 (key为string) + ${string }</h1>
12   <h1>名字 (key为currentuser) + ${currentuser.uName }</h1>
13   <h1>名字 (key为用户) + ${user.uName }</h1>
14 </body>
15 </html>
```

案例-传递参数从控制器到视图-使用Map

User.java

```
1  package entity;
2
3  public class User {
4      private Integer id;
5      private String uName;
6      private Integer uAge;
7
8      public User(Integer id, String uName, Integer uAge) {
9          super();
10         this.id = id;
11         this.uName = uName;
```

```
12  this.uAge = uAge;
13  }
14  public User() {
15      super();
16  }
17
18  public Integer getId() {
19      return id;
20  }
21  public void setId(Integer id) {
22      this.id = id;
23  }
24  public String getuName() {
25      return uName;
26  }
27  public void setuName(String uName) {
28      this.uName = uName;
29  }
30  public Integer getuAge() {
31      return uAge;
32  }
33  public void setuAge(Integer uAge) {
34      this.uAge = uAge;
35  }
36  }
```

IndexController.java

```
1  package controller;
2
3  import java.util.Map;
4
5  import org.apache.log4j.Logger;
6  import org.springframework.stereotype.Controller;
7  import org.springframework.ui.Model;
8  import org.springframework.web.bind.annotation.RequestMapping;
```

















































```
9 import org.springframework.web.servlet.ModelAndView;
10
11 import entity.User;
12
13 //当前类为控制器类（进行页面的映射）
14 @Controller
15 public class IndexController{
16     private Logger log=Logger.getLogger(IndexController.class);
17
18     /**
19      * 参数传递采用的是 controller to view -Map<String, Object>
20      * 从ModelAndView和Model中可以发现，SpringMVC的模板就是一个map
21      * 的数据结构，故而可以使用Map直接作为处理方法入参
22      *
23      * SpringMVC的处理方法中若有map或model类型的入参，就会将请求中的隐含
24      * 的模型对象传递给入参
25      * 因此方法体内可以通过这个入参对模型中的数据进行读写
26      * 注意：SpringMVC的标准用法中，强烈推荐使用Model
27      * @param username
28      * @param model
29      * @return
30      */
31     @RequestMapping("/index3")
32     public String index3 (String username, Map<String, Object> model) {
33         log.info("控制器中的数据: "+username);
34         model.put("currentusername", username);
35
36         User user = new User();
37         user.setName(username);
38         model.put("currentuser", user);
39
40         return "index";
41     }
42 }
```

WebContent/WEB-INF/jspview/index.jsp

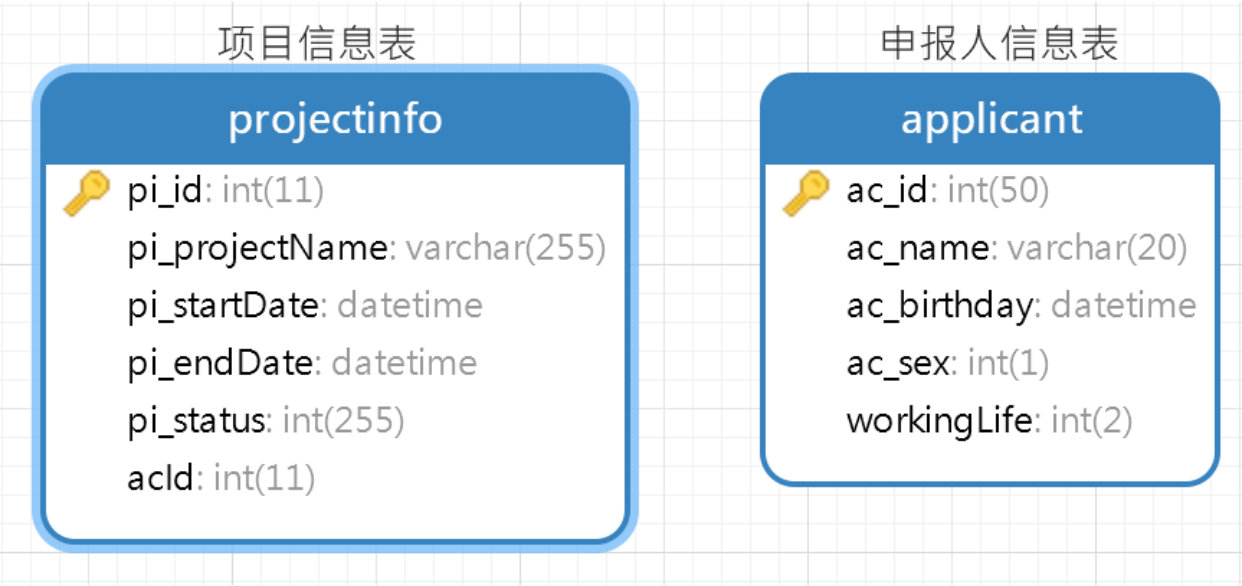
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10  <h1>名字（key为currentusername） + ${currentusername }</h1>
11  <h1>名字（key为currentuser） + ${currentuser.uName }</h1>
12 </body>
13 </html>
```

第二章、SSM整合-项目申报系统

项目结构：

- ▼  yuntu-dpm-curd
 - ▼  src/main/java
 - ▼  com.yuntu.curd.bean
 - >  Applicant.java
 - >  ApplicantExample.java
 - >  Msg.java
 - >  Projectinfo.java
 - >  ProjectinfoExample.java
 - ▼  com.yuntu.curd.controller
 - >  ApplicantController.java
 - >  ProjectInfoController.java
 - ▼  com.yuntu.curd.dao
 - >  ApplicantMapper.java
 - >  ProjectinfoMapper.java
 - ▼  com.yuntu.curd.service
 - >  ApplicantService.java
 - >  ProjectInfoService.java
 - ▼  com.yuntu.curd.test
 - >  MapperTest.java
 - >  MBGTest.java
 - >  MVCTest.java
 -  com.yuntu.curd.utils
 - ▼  src/main/resources
 - ▼  mapper
 -  ApplicantMapper.xml
 -  ProjectinfoMapper.xml
 -  applicationContext.xml
 -  dbconfig.properties
 -  mybatis-config.xml
 -  src/test/java
 -  src/test/resources
 - >  JRE System Library [J2SE-1.5]
 - >  Maven Dependencies
 - ▼  src
 - ▼  main
 - ▼  webapp
 - ▼  static
 - >  bootstrap-3.3.7-dist
 - >  js
 - ▼  WEB-INF
 -  views
 -  dispatcherServlet-servlet.xml
 -  web.xml
 -  index.jsp
 -  test
 - >  target
 -  mbg.xml
 -  pom.xml

数据库表：



页面效果

首页

YUNTU_DPM_CURD

新增删除

	项目编号	项目名称 (#)	开始日期	结束日期	申报状态	申报人	性别	工作年限	操作
<input type="checkbox"/>	1	飞机拧螺丝	2018-07-13	2019-03-05	0	李四	男	2	编辑 删除
<input type="checkbox"/>	2	火车修轮胎	2017-07-21	2019-02-13	1	李莎	女	1	编辑 删除
<input type="checkbox"/>	3	长城贴瓷砖	2017-12-21	2019-02-01	2	张丽鑫	男	2	编辑 删除
<input type="checkbox"/>	4	原子弹抛光	2018-12-21	2019-02-21	1	李莎	女	1	编辑 删除
<input type="checkbox"/>	5	苍蝇拉屎	2017-11-21	2019-02-21	2	王丽	女	5	编辑 删除

当前第1页，总共11页，总共55条记录

首页 < 1 2 3 4 5 > 末页

新增

YUNTU_DPM_CURD

项目申报

ProjectName

ProjectName

StartDate

年 / 月 / 日

EndDate

年 / 月 / 日

Status

已申报

Applicant

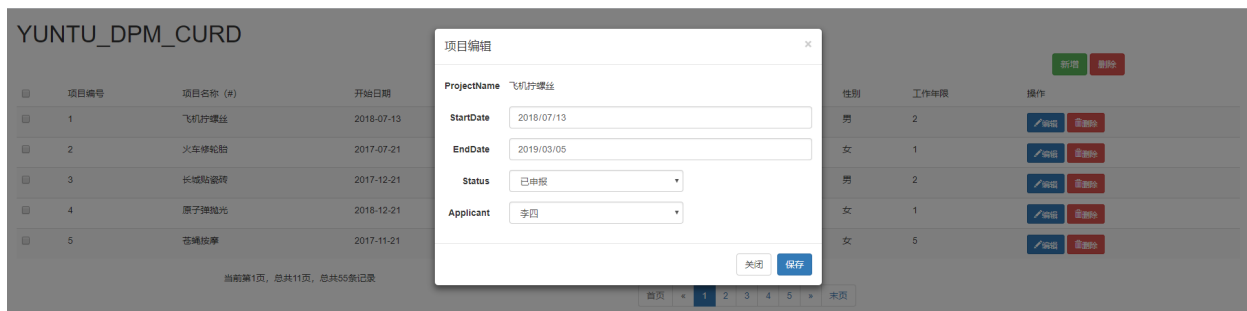
张三

关闭

保存

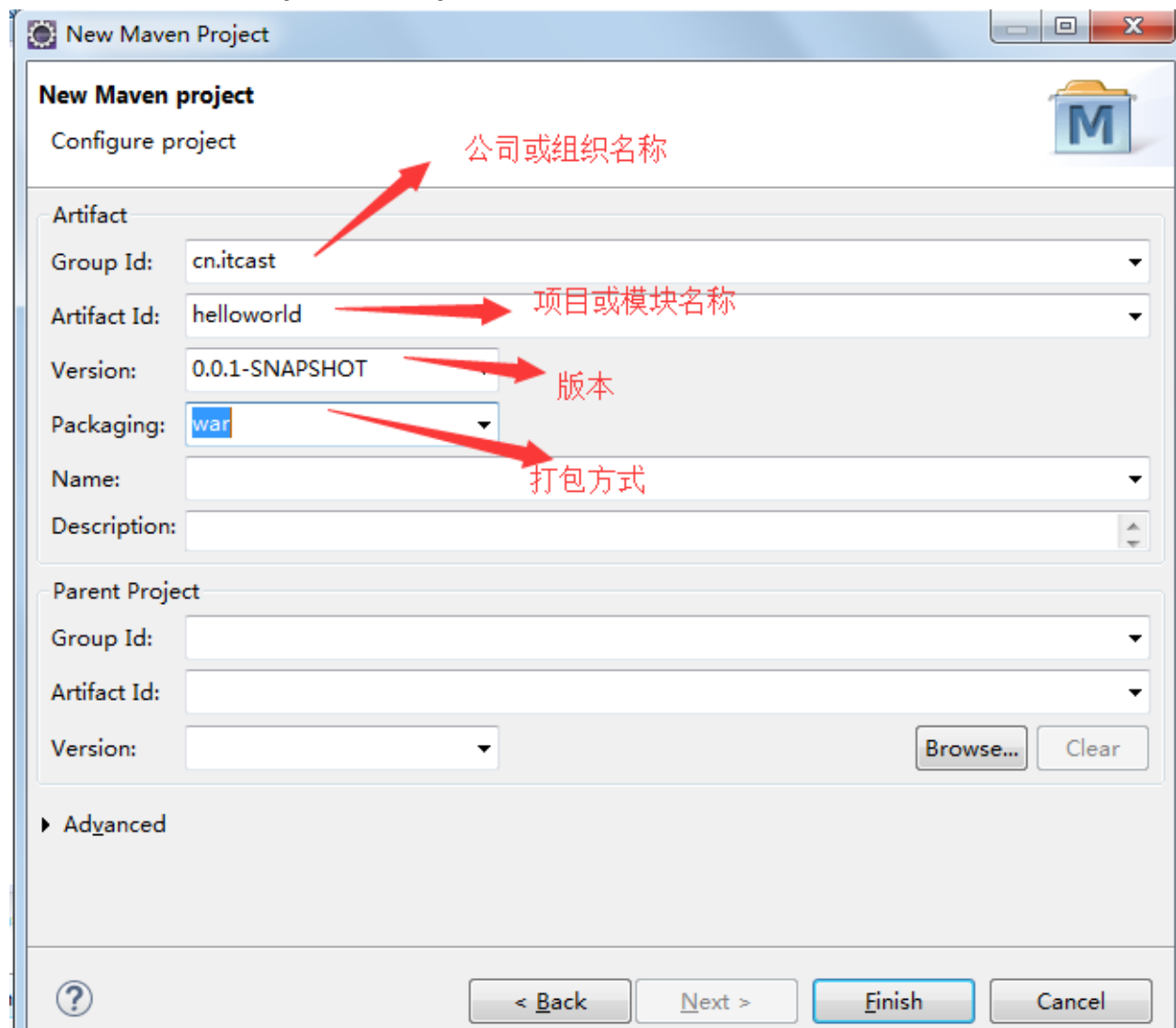
性别	工作年限	操作
男	2	编辑 删除
女	1	编辑 删除
男	2	编辑 删除
女	1	编辑 删除
女	5	编辑 删除

编辑



环境搭建

1.创建Maven项目 (打成war包)



2.配置pom.xml引入项目依赖jar包

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4     http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <!-- -->

```

```
6 <modelVersion>4.0.0</modelVersion>
7 <!-- 公司或者组织域名倒序+项目名 -->
8 <groupId>com.yuntu</groupId>
9 <!-- 模块名 -->
10 <artifactId>yuntu-dpm-curd</artifactId>
11 <!-- 版本 -->
12 <version>0.0.1-SNAPSHOT</version>
13 <packaging>war</packaging>
14
15 <!-- 依赖组 -->
16 <!-- 引入项目依赖jar包 -->
17 <dependencies>
18 <!-- 引入SpringMVC, Spring -->
19 <!-- https://mvnrepository.com/artifact/org.springframework/sp
ring-webmvc -->
20 <dependency>
21 <!-- jar包所在的package -->
22 <groupId>org.springframework</groupId>
23 <!-- jar包名称 -->
24 <artifactId>spring-webmvc</artifactId>
25 <!-- jar包版本 -->
26 <version>4.3.18.RELEASE</version>
27 </dependency>
28 <!-- 引入Spring支持的JDBC -->
29 <!-- https://mvnrepository.com/artifact/org.springframework/sp
ring-jdbc -->
30 <dependency>
31 <groupId>org.springframework</groupId>
32 <artifactId>spring-jdbc</artifactId>
33 <version>4.3.18.RELEASE</version>
34 </dependency>
35 <!-- 引入Spring的面向切面 spring-aspects -->
36 <!-- https://mvnrepository.com/artifact/org.springframework/sp
ring-aspects -->
37 <dependency>
38 <groupId>org.springframework</groupId>
```



```
39 <artifactId>spring-aspects</artifactId>
40 <version>4.3.18.RELEASE</version>
41 </dependency>
42 <!-- 引入Spring提供的测试Spring-test -->
43 <!-- https://mvnrepository.com/artifact/org.springframework/sp
ring-test -->
44 <dependency>
45 <groupId>org.springframework</groupId>
46 <artifactId>spring-test</artifactId>
47 <version>4.3.18.RELEASE</version>
48 </dependency>
49
50
51
52 <!-- 引入mybatis -->
53 <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis --
>
54 <dependency>
55 <groupId>org.mybatis</groupId>
56 <artifactId>mybatis</artifactId>
57 <version>3.4.6</version>
58 </dependency>
59 <!-- 引入mybatis整合Spring -->
60 <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-sp
ring -->
61 <dependency>
62 <groupId>org.mybatis</groupId>
63 <artifactId>mybatis-spring</artifactId>
64 <version>1.3.1</version>
65 </dependency>
66 <!-- 引入数据库连接池，驱动 c3p0 -->
67 <!-- https://mvnrepository.com/artifact/com.mchange/c3p0 -->
68 <dependency>
69 <groupId>com.mchange</groupId>
70 <artifactId>c3p0</artifactId>
71 <version>0.9.5.2</version>
```

```
72 </dependency>
73 <!-- mysql驱动 -->
74 <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-
java -->
75 <dependency>
76 <groupId>mysql</groupId>
77 <artifactId>mysql-connector-java</artifactId>
78 <version>5.1.46</version>
79 </dependency>
80
81
82
83 <!-- 引入jstl -->
84 <!-- https://mvnrepository.com/artifact/jstl/jstl -->
85 <dependency>
86 <groupId>jstl</groupId>
87 <artifactId>jstl</artifactId>
88 <version>1.2</version>
89 </dependency>
90 <!-- 引入servlet-api -->
91 <!-- https://mvnrepository.com/artifact/javax.servlet/javax.se
rvlet-api -->
92 <dependency>
93 <groupId>javax.servlet</groupId>
94 <artifactId>javax.servlet-api</artifactId>
95 <version>3.0.1</version>
96 <!--
97 scope依赖传递
98 这个标签有以下参数：
99 compile（编译）
100 test（测试）
101 runtime（运行时）
102 provided（执行前面三个过程，但是不打包（解决Tomcat也有servlet-api
的问题））
103 system（与provided大致相同，但是依赖项从本地文件系统导入，不走mave
n仓库） -->
104 <scope>provided</scope>
```

```
105     </dependency>
106     <!-- 引入junit测试 -->
107     <!-- https://mvnrepository.com/artifact/junit/junit -->
108     <dependency>
109         <groupId>junit</groupId>
110         <artifactId>junit</artifactId>
111         <version>4.12</version>
112     </dependency>
113
114
115     <!-- mybatis逆向工程 -->
116     <!--
117     https://mvnrepository.com/artifact/org.mybatis.generator/mybatis-generator-core -->
117     <dependency>
118         <groupId>org.mybatis.generator</groupId>
119         <artifactId>mybatis-generator-core</artifactId>
120         <version>1.3.5</version>
121     </dependency>
122
123
124     <!-- 引入PageHelper分页插件 -->
125     <!-- 使用pagehelper只需要在classpath中包含jsqlparser-x.x.x.jar和
126     pagehelper-x.x.x.jar -->
126     <!--
127     https://mvnrepository.com/artifact/com.github.pagehelper/pagehelper-spring-boot-starter -->
127     <dependency>
128         <groupId>com.github.pagehelper</groupId>
129         <artifactId>pagehelper</artifactId>
130         <version>5.0.0</version>
131     </dependency>
132
133     <!-- 引入jackson进行json数据解析 -->
134     <!--
135     https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
135     <dependency>
```

```

136 <groupId>com.fasterxml.jackson.core</groupId>
137 <artifactId>jackson-databind</artifactId>
138 <version>2.9.5</version>
139 </dependency>
140
141 <!-- 引入jsr303进行后台数据校验
142 说明:
143 Tomcat7及以上的服务器是没有问题的，可以直接导入hibernate-validator
144 Tomcat7以下的服务器（EL表达式版本过低），需要额外给服务器的lib包中
    替换新的EL包 -->
145 <!-- https://mvnrepository.com/artifact/org.hibernate.validator/hibernate-validator -->
146 <dependency>
147 <groupId>org.hibernate.validator</groupId>
148 <artifactId>hibernate-validator</artifactId>
149 <version>6.0.13.Final</version>
150 </dependency>
151 </dependencies>
152 </project>

```

实体类：com.yuntu.curd.bean

Applicant.java

```

1 package com.yuntu.curd.bean;
2
3 import java.util.Date;
4
5 public class Applicant {
6     private Integer acId;
7
8     private String acName;
9
10    private Date acBirthday;
11
12    private Integer acSex;
13

```

```
14  private Integer workinglife;
15
16  public Applicant() {
17      super();
18  }
19
20  public Applicant(Integer acId, String acName, Date acBirthday,
    Integer acSex, Integer workinglife) {
21      super();
22      this.acId = acId;
23      this.acName = acName;
24      this.acBirthday = acBirthday;
25      this.acSex = acSex;
26      this.workinglife = workinglife;
27  }
28
29  public Integer getAcId() {
30      return acId;
31  }
32
33  public void setAcId(Integer acId) {
34      this.acId = acId;
35  }
36
37  public String getAcName() {
38      return acName;
39  }
40
41  public void setAcName(String acName) {
42      this.acName = acName == null ? null : acName.trim();
43  }
44
45  public Date getAcBirthday() {
46      return acBirthday;
47  }
```

```

48
49 public void setAcBirthday(Date acBirthday) {
50     this.acBirthday = acBirthday;
51 }
52
53 public Integer getAcSex() {
54     return acSex;
55 }
56
57 public void setAcSex(Integer acSex) {
58     this.acSex = acSex;
59 }
60
61 public Integer getWorkinglife() {
62     return workinglife;
63 }
64
65 public void setWorkinglife(Integer workinglife) {
66     this.workinglife = workinglife;
67 }
68 }

```

ApplicantExample.java

```

1 package com.yuntu.curd.bean;
2
3 import java.util.ArrayList;
4 import java.util.Date;
5 import java.util.List;
6
7 public class ApplicantExample {
8     protected String orderByClause;
9
10    protected boolean distinct;
11
12    protected List<Criteria> oredCriteria;

```

```
13
14 public ApplicantExample() {
15     oredCriteria = new ArrayList<Criteria>();
16 }
17
18 public void setOrderByClause(String orderByClause) {
19     this.orderByClause = orderByClause;
20 }
21
22 public String getOrderByClause() {
23     return orderByClause;
24 }
25
26 public void setDistinct(boolean distinct) {
27     this.distinct = distinct;
28 }
29
30 public boolean isDistinct() {
31     return distinct;
32 }
33
34 public List<Criteria> getOredCriteria() {
35     return oredCriteria;
36 }
37
38 public void or(Criteria criteria) {
39     oredCriteria.add(criteria);
40 }
41
42 public Criteria or() {
43     Criteria criteria = createCriteriaInternal();
44     oredCriteria.add(criteria);
45     return criteria;
46 }
47
```

```
48 public Criteria createCriteria() {
49     Criteria criteria = createCriteriaInternal();
50     if (oredCriteria.size() == 0) {
51         oredCriteria.add(criteria);
52     }
53     return criteria;
54 }
55
56 protected Criteria createCriteriaInternal() {
57     Criteria criteria = new Criteria();
58     return criteria;
59 }
60
61 public void clear() {
62     oredCriteria.clear();
63     orderByClause = null;
64     distinct = false;
65 }
66
67 protected abstract static class GeneratedCriteria {
68     protected List<Criterion> criteria;
69
70     protected GeneratedCriteria() {
71         super();
72         criteria = new ArrayList<Criterion>();
73     }
74
75     public boolean isValid() {
76         return criteria.size() > 0;
77     }
78
79     public List<Criterion> getAllCriteria() {
80         return criteria;
81     }
82 }
```



```
83 public List<Criterion> getCriteria() {
84     return criteria;
85 }
86
87 protected void addCriterion(String condition) {
88     if (condition == null) {
89         throw new RuntimeException("Value for condition cannot be null");
90     }
91     criteria.add(new Criterion(condition));
92 }
93
94 protected void addCriterion(String condition, Object value, String property) {
95     if (value == null) {
96         throw new RuntimeException("Value for " + property + " cannot be null");
97     }
98     criteria.add(new Criterion(condition, value));
99 }
100
101 protected void addCriterion(String condition, Object value1, Object value2, String property) {
102     if (value1 == null || value2 == null) {
103         throw new RuntimeException("Between values for " + property + " cannot be null");
104     }
105     criteria.add(new Criterion(condition, value1, value2));
106 }
107
108 public Criteria andAcIdIsNull() {
109     addCriterion("ac_id is null");
110     return (Criteria) this;
111 }
112
113 public Criteria andAcIdIsNotNull() {
```

```
114     addCriterion("ac_id is not null");
115     return (Criteria) this;
116 }
117
118 public Criteria andAcIdEqualTo(Integer value) {
119     addCriterion("ac_id =", value, "acId");
120     return (Criteria) this;
121 }
122
123 public Criteria andAcIdNotEqualTo(Integer value) {
124     addCriterion("ac_id <>", value, "acId");
125     return (Criteria) this;
126 }
127
128 public Criteria andAcIdGreaterThan(Integer value) {
129     addCriterion("ac_id >", value, "acId");
130     return (Criteria) this;
131 }
132
133 public Criteria andAcIdGreaterThanOrEqualTo(Integer value) {
134     addCriterion("ac_id >=", value, "acId");
135     return (Criteria) this;
136 }
137
138 public Criteria andAcIdLessThan(Integer value) {
139     addCriterion("ac_id <", value, "acId");
140     return (Criteria) this;
141 }
142
143 public Criteria andAcIdLessThanOrEqualTo(Integer value) {
144     addCriterion("ac_id <=", value, "acId");
145     return (Criteria) this;
146 }
147
148 public Criteria andAcIdIn(List<Integer> values) {
```

```
149     addCriterion("ac_id in", values, "acId");
150     return (Criteria) this;
151 }
152
153 public Criteria andAcIdNotIn(List<Integer> values) {
154     addCriterion("ac_id not in", values, "acId");
155     return (Criteria) this;
156 }
157
158 public Criteria andAcIdBetween(Integer value1, Integer value2)
159 {
160     addCriterion("ac_id between", value1, value2, "acId");
161     return (Criteria) this;
162 }
163
164 public Criteria andAcIdNotBetween(Integer value1, Integer value2) {
165     addCriterion("ac_id not between", value1, value2, "acId");
166     return (Criteria) this;
167 }
168
169 public Criteria andAcNameIsNull() {
170     addCriterion("ac_name is null");
171     return (Criteria) this;
172 }
173
174 public Criteria andAcNameIsNotNull() {
175     addCriterion("ac_name is not null");
176     return (Criteria) this;
177 }
178
179 public Criteria andAcNameEqualTo(String value) {
180     addCriterion("ac_name =", value, "acName");
181     return (Criteria) this;
182 }
```

```
183 public Criteria andAcNameNotEqualTo(String value) {
184     addCriterion("ac_name <>", value, "acName");
185     return (Criteria) this;
186 }
187
188 public Criteria andAcNameGreaterThan(String value) {
189     addCriterion("ac_name >", value, "acName");
190     return (Criteria) this;
191 }
192
193 public Criteria andAcNameGreaterThanOrEqualTo(String value) {
194     addCriterion("ac_name >=", value, "acName");
195     return (Criteria) this;
196 }
197
198 public Criteria andAcNameLessThan(String value) {
199     addCriterion("ac_name <", value, "acName");
200     return (Criteria) this;
201 }
202
203 public Criteria andAcNameLessThanOrEqualTo(String value) {
204     addCriterion("ac_name <=", value, "acName");
205     return (Criteria) this;
206 }
207
208 public Criteria andAcNameLike(String value) {
209     addCriterion("ac_name like", value, "acName");
210     return (Criteria) this;
211 }
212
213 public Criteria andAcNameNotLike(String value) {
214     addCriterion("ac_name not like", value, "acName");
215     return (Criteria) this;
216 }
217
```

```
218 public Criteria andAcNameIn(List<String> values) {
219     addCriterion("ac_name in", values, "acName");
220     return (Criteria) this;
221 }
222
223 public Criteria andAcNameNotIn(List<String> values) {
224     addCriterion("ac_name not in", values, "acName");
225     return (Criteria) this;
226 }
227
228 public Criteria andAcNameBetween(String value1, String value2)
229 {
230     addCriterion("ac_name between", value1, value2, "acName");
231     return (Criteria) this;
232 }
233
234 public Criteria andAcNameNotBetween(String value1, String value2) {
235     addCriterion("ac_name not between", value1, value2, "acName");
236     return (Criteria) this;
237 }
238
239 public Criteria andAcBirthdayIsNull() {
240     addCriterion("ac_birthday is null");
241     return (Criteria) this;
242 }
243
244 public Criteria andAcBirthdayIsNotNull() {
245     addCriterion("ac_birthday is not null");
246     return (Criteria) this;
247 }
248
249 public Criteria andAcBirthdayEqualTo(Date value) {
250     addCriterion("ac_birthday =", value, "acBirthday");
251     return (Criteria) this;
252 }
```

```
252
253 public Criteria andAcBirthDayNotEqualTo(Date value) {
254     addCriterion("ac_birthDay <>", value, "acBirthDay");
255     return (Criteria) this;
256 }
257
258 public Criteria andAcBirthDayGreaterThan(Date value) {
259     addCriterion("ac_birthDay >", value, "acBirthDay");
260     return (Criteria) this;
261 }
262
263 public Criteria andAcBirthDayGreaterThanOrEqualTo(Date value)
264 {
265     addCriterion("ac_birthDay >=", value, "acBirthDay");
266     return (Criteria) this;
267 }
268
269 public Criteria andAcBirthDayLessThan(Date value) {
270     addCriterion("ac_birthDay <", value, "acBirthDay");
271     return (Criteria) this;
272 }
273
274 public Criteria andAcBirthDayLessThanOrEqualTo(Date value) {
275     addCriterion("ac_birthDay <=", value, "acBirthDay");
276     return (Criteria) this;
277 }
278
279 public Criteria andAcBirthDayIn(List<Date> values) {
280     addCriterion("ac_birthDay in", values, "acBirthDay");
281     return (Criteria) this;
282 }
283
284 public Criteria andAcBirthDayNotIn(List<Date> values) {
285     addCriterion("ac_birthDay not in", values, "acBirthDay");
286     return (Criteria) this;
```

```
286     }
287
288     public Criteria andAcBirthdayBetween(Date value1, Date value2)
289     {
290         addCriterion("ac_birthday between", value1, value2, "acBirthda
291         y");
292     }
293
294     public Criteria andAcBirthdayNotBetween(Date value1, Date value
295     e2) {
296         addCriterion("ac_birthday not between", value1, value2, "acBir
297         thday");
298     }
299
300     public Criteria andAcSexIsNull() {
301         addCriterion("ac_sex is null");
302     }
303
304     public Criteria andAcSexIsNotNull() {
305         addCriterion("ac_sex is not null");
306     }
307
308     public Criteria andAcSexEqualTo(Integer value) {
309         addCriterion("ac_sex =", value, "acSex");
310     }
311
312     public Criteria andAcSexNotEqualTo(Integer value) {
313         addCriterion("ac_sex <>", value, "acSex");
314     }
315
316     }
317
```

```
318 public Criteria andAcSexGreaterThan(Integer value) {
319     addCriterion("ac_sex >", value, "acSex");
320     return (Criteria) this;
321 }
322
323 public Criteria andAcSexGreaterThanOrEqualTo(Integer value) {
324     addCriterion("ac_sex >=", value, "acSex");
325     return (Criteria) this;
326 }
327
328 public Criteria andAcSexLessThan(Integer value) {
329     addCriterion("ac_sex <", value, "acSex");
330     return (Criteria) this;
331 }
332
333 public Criteria andAcSexLessThanOrEqualTo(Integer value) {
334     addCriterion("ac_sex <=", value, "acSex");
335     return (Criteria) this;
336 }
337
338 public Criteria andAcSexIn(List<Integer> values) {
339     addCriterion("ac_sex in", values, "acSex");
340     return (Criteria) this;
341 }
342
343 public Criteria andAcSexNotIn(List<Integer> values) {
344     addCriterion("ac_sex not in", values, "acSex");
345     return (Criteria) this;
346 }
347
348 public Criteria andAcSexBetween(Integer value1, Integer
value2) {
349     addCriterion("ac_sex between", value1, value2, "acSex");
350     return (Criteria) this;
351 }
```



```
352
353 public Criteria andAcSexNotBetween(Integer value1, Integer value2) {
354     addCriterion("ac_sex not between", value1, value2, "acSex");
355     return (Criteria) this;
356 }
357
358 public Criteria andWorkinglifeIsNull() {
359     addCriterion("workingLife is null");
360     return (Criteria) this;
361 }
362
363 public Criteria andWorkinglifeIsNotNull() {
364     addCriterion("workingLife is not null");
365     return (Criteria) this;
366 }
367
368 public Criteria andWorkinglifeEqualTo(Integer value) {
369     addCriterion("workingLife =", value, "workinglife");
370     return (Criteria) this;
371 }
372
373 public Criteria andWorkinglifeNotEqualTo(Integer value) {
374     addCriterion("workingLife <>", value, "workinglife");
375     return (Criteria) this;
376 }
377
378 public Criteria andWorkinglifeGreaterThan(Integer value) {
379     addCriterion("workingLife >", value, "workinglife");
380     return (Criteria) this;
381 }
382
383 public Criteria andWorkinglifeGreaterThanOrEqualTo(Integer value) {
384     addCriterion("workingLife >=", value, "workinglife");
385     return (Criteria) this;
```

```
386     }
387
388     public Criteria andWorkinglifeLessThan(Integer value) {
389         addCriterion("workingLife <", value, "workinglife");
390         return (Criteria) this;
391     }
392
393     public Criteria andWorkinglifeLessThanOrEqualTo(Integer value) {
394         addCriterion("workingLife <=", value, "workinglife");
395         return (Criteria) this;
396     }
397
398     public Criteria andWorkinglifeIn(List<Integer> values) {
399         addCriterion("workingLife in", values, "workinglife");
400         return (Criteria) this;
401     }
402
403     public Criteria andWorkinglifeNotIn(List<Integer> values) {
404         addCriterion("workingLife not in", values, "workinglife");
405         return (Criteria) this;
406     }
407
408     public Criteria andWorkinglifeBetween(Integer value1, Integer value2) {
409         addCriterion("workingLife between", value1, value2, "workinglife");
410         return (Criteria) this;
411     }
412
413     public Criteria andWorkinglifeNotBetween(Integer value1, Integer value2) {
414         addCriterion("workingLife not between", value1, value2, "workinglife");
415         return (Criteria) this;
416     }
```

```
417     }
418
419     public static class Criteria extends GeneratedCriteria {
420
421         protected Criteria() {
422             super();
423         }
424     }
425
426     public static class Criterion {
427         private String condition;
428
429         private Object value;
430
431         private Object secondValue;
432
433         private boolean noValue;
434
435         private boolean singleValue;
436
437         private boolean betweenValue;
438
439         private boolean listValue;
440
441         private String typeHandler;
442
443         public String getCondition() {
444             return condition;
445         }
446
447         public Object getValue() {
448             return value;
449         }
450
451         public Object getSecondValue() {
```

```
452     return secondValue;
453 }
454
455 public boolean isNoValue() {
456     return noValue;
457 }
458
459 public boolean isSingleValue() {
460     return singleValue;
461 }
462
463 public boolean isBetweenValue() {
464     return betweenValue;
465 }
466
467 public boolean isListValue() {
468     return listValue;
469 }
470
471 public String getTypeHandler() {
472     return typeHandler;
473 }
474
475 protected Criterion(String condition) {
476     super();
477     this.condition = condition;
478     this.typeHandler = null;
479     this.noValue = true;
480 }
481
482 protected Criterion(String condition, Object value, String typeHandler) {
483     super();
484     this.condition = condition;
485     this.value = value;
```

```

486     this.typeHandler = typeHandler;
487     if (value instanceof List<?>) {
488         this.listValue = true;
489     } else {
490         this.singleValue = true;
491     }
492 }
493
494     protected Criterion(String condition, Object value) {
495         this(condition, value, null);
496     }
497
498     protected Criterion(String condition, Object value, Object secondValue, String typeHandler) {
499         super();
500         this.condition = condition;
501         this.value = value;
502         this.secondValue = secondValue;
503         this.typeHandler = typeHandler;
504         this.betweenValue = true;
505     }
506
507     protected Criterion(String condition, Object value, Object secondValue) {
508         this(condition, value, secondValue, null);
509     }
510 }
511 }

```

Msg.java

```

1 package com.yuntu.curd.bean;
2
3 /**
4  * 用来返回通用的类数据
5  * @author 阿华
6  */

```

```
7
8 import java.util.HashMap;
9 import java.util.Map;
10
11 public class Msg {
12     //状态码 200成功 400失败
13     private int code;
14     //提示信息 ("操作成功", "操作失败".....)
15     private String msg;
16     //用户要返回给浏览器的数据
17     private Map<String, Object> extend=new HashMap<String, Object>
18     ();
19
20     //成功，同时返回Msg对象
21     public static Msg success() {
22         Msg result = new Msg();
23         result.setCode(200);
24         result.setMsg("成功");
25         return result;
26     }
27
28     //失败，同时返回Msg对象
29     public static Msg fail() {
30         Msg result = new Msg();
31         result.setCode(400);
32         result.setMsg("失败");
33         return result;
34     }
35
36     //添加数据并返回
37     public Msg add(String key, Object value) {
38         this.getExtend().put(key,value);
39         return this;
40     }
41
```

```
42
43 //get set
44 public int getCode() {
45     return code;
46 }
47
48
49 public void setCode(int code) {
50     this.code = code;
51 }
52
53
54 public String getMsg() {
55     return msg;
56 }
57
58
59 public void setMsg(String msg) {
60     this.msg = msg;
61 }
62
63
64 public Map<String, Object> getExtend() {
65     return extend;
66 }
67
68
69 public void setExtend(Map<String, Object> extend) {
70     this.extend = extend;
71 }
72
73
74 public Msg(int code, String msg, Map<String, Object> extend) {
75     super();
76     this.code = code;
```

```

77     this.msg = msg;
78     this.extend = extend;
79 }
80
81
82 public Msg() {
83     super();
84 }
85
86
87
88
89
90 }
91

```

Projectinfo.java

```

1  package com.yuntu.curd.bean;
2
3  import java.util.Date;
4
5  import javax.validation.constraints.Pattern;
6
7  import org.springframework.format.annotation.DateTimeFormat;
8
9  public class Projectinfo {
10     private Integer piId;
11
12     //正则表达式中\用来转义，但是在java中属于特殊字符，所以在\前再加一个\
13     @Pattern(regexp="^[a-zA-Z0-9_-]{6,64}$|(^[\u2E80-\u9FFF]{2,32}$)",message="后台：论文名为2~32为中文，或者6~64为大小写字母、数字、下划线、横杠的组合")
14     private String piProjectname;
15
16     @DateTimeFormat(pattern="yyyy-MM-dd")
17     private Date piStartdate;

```



```
18
19 @DateTimeFormat(pattern="yyyy-MM-dd")
20 private Date piEnddate;
21
22 private Integer piStatus;
23
24 private Integer acid;
25
26 //用于查询出项目的同时，查询出申报人对象信息
27 private Applicant applicant;
28
29 public Projectinfo(Integer piId, String piProjectname, Date pi
    Startdate, Date piEnddate, Integer piStatus,
30 Integer acid, Applicant applicant) {
31     super();
32     this.piId = piId;
33     this.piProjectname = piProjectname;
34     this.piStartdate = piStartdate;
35     this.piEnddate = piEnddate;
36     this.piStatus = piStatus;
37     this.acid = acid;
38     this.applicant = applicant;
39 }
40
41 public Projectinfo() {
42 }
43
44 public Applicant getApplicant() {
45     return applicant;
46 }
47
48 public void setApplicant(Applicant applicant) {
49     this.applicant = applicant;
50 }
51
52 public Integer getPiId() {
```

```
53     return piId;
54 }
55
56 public void setPiId(Integer piId) {
57     this.piId = piId;
58 }
59
60 public String getPiProjectname() {
61     return piProjectname;
62 }
63
64 public void setPiProjectname(String piProjectname) {
65     this.piProjectname = piProjectname == null ? null : piProjectn
ame.trim();
66 }
67
68 public Date getPiStartdate() {
69     return piStartdate;
70 }
71
72 public void setPiStartdate(Date piStartdate) {
73     this.piStartdate = piStartdate;
74 }
75
76 public Date getPiEnddate() {
77     return piEnddate;
78 }
79
80 public void setPiEnddate(Date piEnddate) {
81     this.piEnddate = piEnddate;
82 }
83
84 public Integer getPiStatus() {
85     return piStatus;
86 }
```

```
87
88  public void setPiStatus(Integer piStatus) {
89      this.piStatus = piStatus;
90  }
91
92  public Integer getAcid() {
93      return acid;
94  }
95
96  public void setAcid(Integer acid) {
97      this.acid = acid;
98  }
99 }
```

ProjectinfoExample.java

```
1  package com.yuntu.curd.bean;
2
3  import java.util.ArrayList;
4  import java.util.Date;
5  import java.util.List;
6
7  public class ProjectinfoExample {
8      protected String orderByClause;
9
10     protected boolean distinct;
11
12     protected List<Criteria> oredCriteria;
13
14     public ProjectinfoExample() {
15         oredCriteria = new ArrayList<Criteria>();
16     }
17
18     public void setOrderByClause(String orderByClause) {
19         this.orderByClause = orderByClause;
20     }
```

```
21
22 public String getOrderByClause() {
23     return orderByClause;
24 }
25
26 public void setDistinct(boolean distinct) {
27     this.distinct = distinct;
28 }
29
30 public boolean isDistinct() {
31     return distinct;
32 }
33
34 public List<Criteria> getOredCriteria() {
35     return oredCriteria;
36 }
37
38 public void or(Criteria criteria) {
39     oredCriteria.add(criteria);
40 }
41
42 public Criteria or() {
43     Criteria criteria = createCriteriaInternal();
44     oredCriteria.add(criteria);
45     return criteria;
46 }
47
48 public Criteria createCriteria() {
49     Criteria criteria = createCriteriaInternal();
50     if (oredCriteria.size() == 0) {
51         oredCriteria.add(criteria);
52     }
53     return criteria;
54 }
55
```

```
56  protected Criteria createCriteriaInternal() {
57  Criteria criteria = new Criteria();
58  return criteria;
59  }
60
61  public void clear() {
62  oredCriteria.clear();
63  orderByClause = null;
64  distinct = false;
65  }
66
67  protected abstract static class GeneratedCriteria {
68  protected List<Criterion> criteria;
69
70  protected GeneratedCriteria() {
71  super();
72  criteria = new ArrayList<Criterion>();
73  }
74
75  public boolean isValid() {
76  return criteria.size() > 0;
77  }
78
79  public List<Criterion> getAllCriteria() {
80  return criteria;
81  }
82
83  public List<Criterion> getCriteria() {
84  return criteria;
85  }
86
87  protected void addCriterion(String condition) {
88  if (condition == null) {
89  throw new RuntimeException("Value for condition cannot be null");
90  }
```

```

90     }
91     criteria.add(new Criterion(condition));
92     }
93
94     protected void addCriterion(String condition, Object value, String property) {
95         if (value == null) {
96             throw new RuntimeException("Value for " + property + " cannot be null");
97         }
98         criteria.add(new Criterion(condition, value));
99     }
100
101     protected void addCriterion(String condition, Object value1, Object value2, String property) {
102         if (value1 == null || value2 == null) {
103             throw new RuntimeException("Between values for " + property + " cannot be null");
104         }
105         criteria.add(new Criterion(condition, value1, value2));
106     }
107
108     public Criteria andPiIdIsNull() {
109         addCriterion("pi_id is null");
110         return (Criteria) this;
111     }
112
113     public Criteria andPiIdIsNotNull() {
114         addCriterion("pi_id is not null");
115         return (Criteria) this;
116     }
117
118     public Criteria andPiIdEqualTo(Integer value) {
119         addCriterion("pi_id =", value, "piId");
120         return (Criteria) this;
121     }

```

```
122
123 public Criteria andPiIdNotEqualTo(Integer value) {
124     addCriterion("pi_id <>", value, "piId");
125     return (Criteria) this;
126 }
127
128 public Criteria andPiIdGreaterThan(Integer value) {
129     addCriterion("pi_id >", value, "piId");
130     return (Criteria) this;
131 }
132
133 public Criteria andPiIdGreaterThanOrEqualTo(Integer value) {
134     addCriterion("pi_id >=", value, "piId");
135     return (Criteria) this;
136 }
137
138 public Criteria andPiIdLessThan(Integer value) {
139     addCriterion("pi_id <", value, "piId");
140     return (Criteria) this;
141 }
142
143 public Criteria andPiIdLessThanOrEqualTo(Integer value) {
144     addCriterion("pi_id <=", value, "piId");
145     return (Criteria) this;
146 }
147
148 public Criteria andPiIdIn(List<Integer> values) {
149     addCriterion("pi_id in", values, "piId");
150     return (Criteria) this;
151 }
152
153 public Criteria andPiIdNotIn(List<Integer> values) {
154     addCriterion("pi_id not in", values, "piId");
155     return (Criteria) this;
156 }
```

```
157
158 public Criteria andPiIdBetween(Integer value1, Integer value2)
159 {
160     addCriterion("pi_id between", value1, value2, "piId");
161     return (Criteria) this;
162 }
163
164 public Criteria andPiIdNotBetween(Integer value1, Integer value2) {
165     addCriterion("pi_id not between", value1, value2, "piId");
166     return (Criteria) this;
167 }
168
169 public Criteria andPiProjectnameIsNull() {
170     addCriterion("pi_projectName is null");
171     return (Criteria) this;
172 }
173
174 public Criteria andPiProjectnameIsNotNull() {
175     addCriterion("pi_projectName is not null");
176     return (Criteria) this;
177 }
178
179 public Criteria andPiProjectnameEqualTo(String value) {
180     addCriterion("pi_projectName =", value, "piProjectname");
181     return (Criteria) this;
182 }
183
184 public Criteria andPiProjectnameNotEqualTo(String value) {
185     addCriterion("pi_projectName <>", value, "piProjectname");
186     return (Criteria) this;
187 }
188
189 public Criteria andPiProjectnameGreaterThan(String value) {
190     addCriterion("pi_projectName >", value, "piProjectname");
191     return (Criteria) this;
```



```
191     }
192
193     public Criteria andPiProjectnameGreaterThanOrEqualTo(String value) {
194         addCriterion("pi_projectName >=", value, "piProjectname");
195         return (Criteria) this;
196     }
197
198     public Criteria andPiProjectnameLessThan(String value) {
199         addCriterion("pi_projectName <", value, "piProjectname");
200         return (Criteria) this;
201     }
202
203     public Criteria andPiProjectnameLessThanOrEqualTo(String
value) {
204         addCriterion("pi_projectName <=", value, "piProjectname");
205         return (Criteria) this;
206     }
207
208     public Criteria andPiProjectnameLike(String value) {
209         addCriterion("pi_projectName like", value, "piProjectname");
210         return (Criteria) this;
211     }
212
213     public Criteria andPiProjectnameNotLike(String value) {
214         addCriterion("pi_projectName not like", value,
"piProjectname");
215         return (Criteria) this;
216     }
217
218     public Criteria andPiProjectnameIn(List<String> values) {
219         addCriterion("pi_projectName in", values, "piProjectname");
220         return (Criteria) this;
221     }
222
223     public Criteria andPiProjectnameNotIn(List<String> values) {
```

```
224     addCriterion("pi_projectName not in", values,  
    "piProjectname");  
225     return (Criteria) this;  
226 }  
227  
228 public Criteria andPiProjectnameBetween(String value1, String  
    value2) {  
229     addCriterion("pi_projectName between", value1, value2, "piProj  
    ectname");  
230     return (Criteria) this;  
231 }  
232  
233 public Criteria andPiProjectnameNotBetween(String value1, Stri  
    ng value2) {  
234     addCriterion("pi_projectName not between", value1, value2, "pi  
    Projectname");  
235     return (Criteria) this;  
236 }  
237  
238 public Criteria andPiStartdateIsNull() {  
239     addCriterion("pi_startDate is null");  
240     return (Criteria) this;  
241 }  
242  
243 public Criteria andPiStartdateIsNotNull() {  
244     addCriterion("pi_startDate is not null");  
245     return (Criteria) this;  
246 }  
247  
248 public Criteria andPiStartdateEqualTo(Date value) {  
249     addCriterion("pi_startDate =", value, "piStartdate");  
250     return (Criteria) this;  
251 }  
252  
253 public Criteria andPiStartdateNotEqualTo(Date value) {  
254     addCriterion("pi_startDate <>", value, "piStartdate");
```

```
255     return (Criteria) this;
256 }
257
258 public Criteria andPiStartdateGreaterThan(Date value) {
259     addCriterion("pi_startDate >", value, "piStartdate");
260     return (Criteria) this;
261 }
262
263 public Criteria andPiStartdateGreaterThanOrEqualTo(Date value) {
264     addCriterion("pi_startDate >=", value, "piStartdate");
265     return (Criteria) this;
266 }
267
268 public Criteria andPiStartdateLessThan(Date value) {
269     addCriterion("pi_startDate <", value, "piStartdate");
270     return (Criteria) this;
271 }
272
273 public Criteria andPiStartdateLessThanOrEqualTo(Date value) {
274     addCriterion("pi_startDate <=", value, "piStartdate");
275     return (Criteria) this;
276 }
277
278 public Criteria andPiStartdateIn(List<Date> values) {
279     addCriterion("pi_startDate in", values, "piStartdate");
280     return (Criteria) this;
281 }
282
283 public Criteria andPiStartdateNotIn(List<Date> values) {
284     addCriterion("pi_startDate not in", values, "piStartdate");
285     return (Criteria) this;
286 }
287
288 public Criteria andPiStartdateBetween(Date value1, Date
value2) {
```

```
289     addCriterion("pi_startDate between", value1, value2, "piStartDate");
290     return (Criteria) this;
291 }
292
293 public Criteria andPiStartdateNotBetween(Date value1, Date value2) {
294     addCriterion("pi_startDate not between", value1, value2, "piStartDate");
295     return (Criteria) this;
296 }
297
298 public Criteria andPiEnddateIsNull() {
299     addCriterion("pi_endDate is null");
300     return (Criteria) this;
301 }
302
303 public Criteria andPiEnddateIsNotNull() {
304     addCriterion("pi_endDate is not null");
305     return (Criteria) this;
306 }
307
308 public Criteria andPiEnddateEqualTo(Date value) {
309     addCriterion("pi_endDate =", value, "piEnddate");
310     return (Criteria) this;
311 }
312
313 public Criteria andPiEnddateNotEqualTo(Date value) {
314     addCriterion("pi_endDate <>", value, "piEnddate");
315     return (Criteria) this;
316 }
317
318 public Criteria andPiEnddateGreaterThan(Date value) {
319     addCriterion("pi_endDate >", value, "piEnddate");
320     return (Criteria) this;
321 }
```

```
322
323 public Criteria andPiEnddateGreaterThanOrEqualTo(Date value) {
324     addCriterion("pi_endDate >=", value, "piEnddate");
325     return (Criteria) this;
326 }
327
328 public Criteria andPiEnddateLessThan(Date value) {
329     addCriterion("pi_endDate <", value, "piEnddate");
330     return (Criteria) this;
331 }
332
333 public Criteria andPiEnddateLessThanOrEqualTo(Date value) {
334     addCriterion("pi_endDate <=", value, "piEnddate");
335     return (Criteria) this;
336 }
337
338 public Criteria andPiEnddateIn(List<Date> values) {
339     addCriterion("pi_endDate in", values, "piEnddate");
340     return (Criteria) this;
341 }
342
343 public Criteria andPiEnddateNotIn(List<Date> values) {
344     addCriterion("pi_endDate not in", values, "piEnddate");
345     return (Criteria) this;
346 }
347
348 public Criteria andPiEnddateBetween(Date value1, Date value2)
349 {
350     addCriterion("pi_endDate between", value1, value2,
351         "piEnddate");
352     return (Criteria) this;
353 }
354
355 public Criteria andPiEnddateNotBetween(Date value1, Date value2) {
```

```
354     addCriterion("pi_endDate not between", value1, value2, "piEndDate");
355     return (Criteria) this;
356 }
357
358 public Criteria andPiStatusIsNull() {
359     addCriterion("pi_status is null");
360     return (Criteria) this;
361 }
362
363 public Criteria andPiStatusIsNotNull() {
364     addCriterion("pi_status is not null");
365     return (Criteria) this;
366 }
367
368 public Criteria andPiStatusEqualTo(Integer value) {
369     addCriterion("pi_status =", value, "piStatus");
370     return (Criteria) this;
371 }
372
373 public Criteria andPiStatusNotEqualTo(Integer value) {
374     addCriterion("pi_status <>", value, "piStatus");
375     return (Criteria) this;
376 }
377
378 public Criteria andPiStatusGreaterThan(Integer value) {
379     addCriterion("pi_status >", value, "piStatus");
380     return (Criteria) this;
381 }
382
383 public Criteria andPiStatusGreaterThanOrEqualTo(Integer value) {
384     addCriterion("pi_status >=", value, "piStatus");
385     return (Criteria) this;
386 }
387
```

```
388 public Criteria andPiStatusLessThan(Integer value) {
389     addCriterion("pi_status <", value, "piStatus");
390     return (Criteria) this;
391 }
392
393 public Criteria andPiStatusLessThanOrEqualTo(Integer value) {
394     addCriterion("pi_status <=", value, "piStatus");
395     return (Criteria) this;
396 }
397
398 public Criteria andPiStatusIn(List<Integer> values) {
399     addCriterion("pi_status in", values, "piStatus");
400     return (Criteria) this;
401 }
402
403 public Criteria andPiStatusNotIn(List<Integer> values) {
404     addCriterion("pi_status not in", values, "piStatus");
405     return (Criteria) this;
406 }
407
408 public Criteria andPiStatusBetween(Integer value1, Integer value2) {
409     addCriterion("pi_status between", value1, value2, "piStatus");
410     return (Criteria) this;
411 }
412
413 public Criteria andPiStatusNotBetween(Integer value1, Integer value2) {
414     addCriterion("pi_status not between", value1, value2, "piStatus");
415     return (Criteria) this;
416 }
417
418 public Criteria andAcidIsNull() {
419     addCriterion("acId is null");
420     return (Criteria) this;
```

```
421 }
422
423 public Criteria andAcidIsNotNull() {
424     addCriterion("acId is not null");
425     return (Criteria) this;
426 }
427
428 public Criteria andAcidEqualTo(Integer value) {
429     addCriterion("acId =", value, "acid");
430     return (Criteria) this;
431 }
432
433 public Criteria andAcidNotEqualTo(Integer value) {
434     addCriterion("acId <>", value, "acid");
435     return (Criteria) this;
436 }
437
438 public Criteria andAcidGreaterThan(Integer value) {
439     addCriterion("acId >", value, "acid");
440     return (Criteria) this;
441 }
442
443 public Criteria andAcidGreaterThanOrEqualTo(Integer value) {
444     addCriterion("acId >=", value, "acid");
445     return (Criteria) this;
446 }
447
448 public Criteria andAcidLessThan(Integer value) {
449     addCriterion("acId <", value, "acid");
450     return (Criteria) this;
451 }
452
453 public Criteria andAcidLessThanOrEqualTo(Integer value) {
454     addCriterion("acId <=", value, "acid");
455     return (Criteria) this;
```



```
456 }
457
458 public Criteria andAcidIn(List<Integer> values) {
459     addCriterion("acId in", values, "acid");
460     return (Criteria) this;
461 }
462
463 public Criteria andAcidNotIn(List<Integer> values) {
464     addCriterion("acId not in", values, "acid");
465     return (Criteria) this;
466 }
467
468 public Criteria andAcidBetween(Integer value1, Integer value2)
469 {
470     addCriterion("acId between", value1, value2, "acid");
471     return (Criteria) this;
472 }
473
474 public Criteria andAcidNotBetween(Integer value1, Integer value2) {
475     addCriterion("acId not between", value1, value2, "acid");
476     return (Criteria) this;
477 }
478
479 public static class Criteria extends GeneratedCriteria {
480
481     protected Criteria() {
482         super();
483     }
484 }
485
486 public static class Criterion {
487     private String condition;
488
489     private Object value;
```

```
490
491 private Object secondValue;
492
493 private boolean noValue;
494
495 private boolean singleValue;
496
497 private boolean betweenValue;
498
499 private boolean listValue;
500
501 private String typeHandler;
502
503 public String getCondition() {
504     return condition;
505 }
506
507 public Object getValue() {
508     return value;
509 }
510
511 public Object getSecondValue() {
512     return secondValue;
513 }
514
515 public boolean isNoValue() {
516     return noValue;
517 }
518
519 public boolean isSingleValue() {
520     return singleValue;
521 }
522
523 public boolean isBetweenValue() {
524     return betweenValue;
```

```

525     }
526
527     public boolean isListValue() {
528         return listValue;
529     }
530
531     public String getTypeHandler() {
532         return typeHandler;
533     }
534
535     protected Criterion(String condition) {
536         super();
537         this.condition = condition;
538         this.typeHandler = null;
539         this.noValue = true;
540     }
541
542     protected Criterion(String condition, Object value, String typeHandler) {
543         super();
544         this.condition = condition;
545         this.value = value;
546         this.typeHandler = typeHandler;
547         if (value instanceof List<?>) {
548             this.listValue = true;
549         } else {
550             this.singleValue = true;
551         }
552     }
553
554     protected Criterion(String condition, Object value) {
555         this(condition, value, null);
556     }
557
558     protected Criterion(String condition, Object value, Object secondValue, String typeHandler) {

```

```

559     super();
560     this.condition = condition;
561     this.value = value;
562     this.secondValue = secondValue;
563     this.typeHandler = typeHandler;
564     this.betweenValue = true;
565 }
566
567     protected Criterion(String condition, Object value, Object secondValue) {
568         this(condition, value, secondValue, null);
569     }
570 }
571 }

```

控制器：com.yuntu.curd.controller

ApplicantController.java

```

1  package com.yuntu.curd.controller;
2
3  import java.util.List;
4
5  import org.springframework.beans.factory.annotation.Autowired;
6  import org.springframework.stereotype.Controller;
7  import org.springframework.web.bind.annotation.RequestMapping;
8  import org.springframework.web.bind.annotation.ResponseBody;
9
10 import com.yuntu.curd.bean.Applicant;
11 import com.yuntu.curd.bean.Msg;
12 import com.yuntu.curd.service.ApplicantService;
13
14 /**
15  * 申报人控制器
16  * @author Administrator
17  *
18  */

```

```

19 @Controller
20 public class ApplicantController {
21     @Autowired
22     private ApplicantService applicantService;
23
24     /**
25      * 获取所有申报人的信息，并返回JSON数据
26      */
27     @RequestMapping("/applicants")
28     @ResponseBody //响应的数据是一个JSON数据
29     public Msg getApplicants() {
30         List<Applicant> list = applicantService.getApplicants();
31         return Msg.success().add("applicants", list);
32     }
33 }
34

```

ProjectInfoController.java

```

1 package com.yuntu.curd.controller;
2
3 import java.util.ArrayList;
4 import java.util.HashMap;
5 import java.util.List;
6 import java.util.Map;
7
8 import javax.validation.Valid;
9
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.stereotype.Controller;
12 import org.springframework.ui.Model;
13 import org.springframework.validation.BindingResult;
14 import org.springframework.validation.FieldError;
15 import org.springframework.web.bind.annotation.PathVariable;
16 import org.springframework.web.bind.annotation.RequestBody;
17 import org.springframework.web.bind.annotation.RequestMapping;

```

```
18 import org.springframework.web.bind.annotation.RequestMethod;
19 import org.springframework.web.bind.annotation.RequestParam;
20 import org.springframework.web.bind.annotation.ResponseBody;
21
22 import com.github.pagehelper.PageHelper;
23 import com.github.pagehelper.PageInfo;
24 import com.yuntu.curd.bean.Msg;
25 import com.yuntu.curd.bean.Projectinfo;
26 import com.yuntu.curd.service.ProjectInfoService;
27 /**
28  * 项目信息控制器
29  * @author Administrator
30  *
31  */
32 @Controller
33 public class ProjectInfoController {
34     @Autowired
35     private ProjectInfoService projectInfoService;
36
37     /**
38      * 将单个删除变成既能单个删除，又能多个删除
39      * 传递的参数为ids
40      *
41      * 如果是单个值 ids:1
42      * 如果是多个值 ids: 1-2-3
43      * @param ids
44      * @return
45      */
46     @RequestMapping(value="/projectInfo/{ids}",method=RequestMethod.DELETE)
47     @ResponseBody
48     public Msg deleteProjectInfoById(@PathVariable("ids") String ids) {
49         //判断是否包含-
50         if (ids.contains("-")) { //多个删除
51             List<Integer> int_ids = new ArrayList<Integer>();
```

```

52 String[] str_ids=ids.split("-");
53 //循环将字符串数组中的数据转换为Integer，并放到集合中
54 for (String str_id : str_ids) {
55     int_ids.add(Integer.valueOf(str_id));
56 }
57 projectInfoService.deletePrijectInfos(int_ids);;
58 } else { //单个删除
59     Integer id=Integer.valueOf(ids);
60     projectInfoService.deleteProjectInfo(id);
61 }
62 return Msg.success();
63 }
64
65 /**
66  * 更改数据
67  * @return
68  */
69 //如果要把piId传入对象，piId必须是Projectinfo对象中的属性且名字必须一致
70 @RequestMapping(value="/projectInfo/{piId}", method=RequestMethod.PUT)
71 @ResponseBody
72 public Msg saveProjectInfo(Projectinfo projectInfo) {
73     /**
74      * PUT方式说明
75      * Tomcat在进行应用处理时，值判断是否包含POST请求，如果不是POST直接结束数据获取
76      * 所以PUT请求时无法拿到想要的的数据，默认获得的全部是空
77      * 解决方案：
78      * 必须在web.xml文件添加HttpPutFormContentFilter过滤器
79      * 将请求中的数据解析包装成集合，同request被重新包装
80      * 这样的话，request.getParameter("")被重写，这样讲就能通过PUT请求那数据
81      *
82      */
83     projectInfoService.updateProjectInfo(projectInfo);

```

```

84     return Msg.success();
85 }
86
87
88 /**
89  * 根据pi_id查询项目的详细信息
90  * @param pi_Id
91  * @return
92  */
93
94 @RequestMapping(value="/projectInfo/{id}",method=RequestMethod.GET)
95 @ResponseBody
96 public Msg getProjectInfo(@PathVariable("id") Integer pi_Id) {
97     // @PathVariable("id"): 代表从路径中获取id
98     Projectinfo projectInfo = projectInfoService.getProjectInfo(pi_Id);
99     return Msg.success().add("projectInfo",projectInfo);
100 }
101
102 /**
103  * 验证项目名是否存在
104  * @param projectName
105  * @return
106  */
107
108 @RequestMapping("/checkProjectInfo")
109 @ResponseBody
110 public Msg checkProjectInfo (@RequestParam("projectName") String projectName) {
111     //判断项目名是否合法
112     String regProjectName="(^[a-zA-Z0-9_-]{6,64}$)|(^[\u2E80-\u9FFF]{2,32}$)";
113     if (regProjectName.matches(regProjectName)) {
114         return Msg.fail().add("validate_msg", "重复验证：论文名为2~32为中文，或者6~64为大小写字母、数字、下划线、横杠的组合");
115     } else {
116         //执行查询

```



```

115     boolean flag =
        projectInfoService.checkProjectInfo(projectName);
116     if(flag) {
117         return Msg.success();
118     } else {
119         return Msg.fail().add("validate_msg","x 不可用");
120     }
121 }
122 }
123
124 /**
125  * 保存，返回json数据
126  * 请求方式（规定）：
127  * URI:
128  * /projectInfo/{id} GET方式 查询
129  * /projectInfo/ POST方式 添加
130  * /projectInfo/{id} PUT方式 修改
131  * /projectInfo/{id} DELETE方式 删除
132  *
133  * @param projectInfo
134  * @param result
135  * @return
136  */
137 @RequestMapping(value="/projectInfo",method=RequestMethod.POST)
138 @ResponseBody
139 public Msg saveProjectInfo(@Valid ProjectInfo projectInfo, BindingResult result) {
140     //传递参数前添加@Valid 进行JSR303数据校验；BindingResult: 绑定的
    验证结果
141     //判断验证结果是否有错误信息
142     if (result.hasErrors()) { //如果有错误信息
143         //创建用于存放错误的map集合
144         Map<String, Object> map = new HashMap<String, Object>();
145         //返回所有的字段错误
146         List<FieldError> errors = result.getFieldErrors();

```

```
147 //将所有得到的错误信息，，都放在map集合中
148 for (FieldError fieldError : errors) {
149     System.out.println("错误的字段: "+fieldError.getField());
150     System.out.println("错误的提示信息: "+fieldError.getDefaultMessage());
151     map.put(fieldError.getField(),
152             fieldError.getDefaultMessage());
153 }
154 //返回到页面处理上
155 return Msg.fail().add("errorFields", map);
156 } else {
157     projectInfoService.saveProjectInfo(projectInfo);
158     return Msg.success();
159 }
160
161
162
163 /**
164  * ajax请求,分页查询页面数据，返回JSON格式
165  * @param pn
166  * @return
167  */
168 @RequestMapping("/projectInfos")
169 @ResponseBody
170 public Msg getProjectInfosWithJson(@RequestParam(value="pn", defaultValue="1") Integer pn) {
171     //查询所有员工
172     //引入PageHelper分页插件
173     //查询之前值需要调用，传入页码及第一页容量
174     PageHelper.startPage(pn, 5);
175     List<ProjectInfo> infos = projectInfoService.getAll();
176     //使用PageInfo进行包装查询后的结果，只需要将PageInfo交给页面
177     //PageInfo(封装了详细的分页信息，包括查询出来的信息，和连续查询显示的页数)
178     PageInfo page = new PageInfo(infos,5);
179
```

```

180 //直接将pageInfo对象所有的信息进行返回（JSON格式）
181 return Msg.success().add("pageInfo", page);
182 }
183
184
185 //RequestMapping("/projectInfos")
186 public String getProjectInfos(@RequestParam(value="pn",default
Value="1") Integer pn,
187 Model model) {
188 //查询所有员工
189 //引入PageHelper分页插件
190 //查询之前值需要调用，传入页码及第一页容量
191 PageHelper.startPage(pn, 5);
192 List<Projectinfo> infos = projectInfoService.getAll();
193 //使用Pageinfo进行包装查询后的结果，只需要将PageInfo交给页面
194 //PageInfo(封装了详细的分页信息，包括查询出来的信息，和连续查询显示
的页数)
195 PageInfo page = new PageInfo(infos,5);
196 model.addAttribute("PageInfo",page);
197 return "list";
198 }
199
200 }

```

数据访问层：com.yuntu.curd.dao

ApplicantMapper.java

```

1 package com.yuntu.curd.dao;
2
3 import com.yuntu.curd.bean.Applicant;
4 import com.yuntu.curd.bean.ApplicantExample;
5 import java.util.List;
6 import org.apache.ibatis.annotations.Param;
7
8 public interface ApplicantMapper {
9     long countByExample(ApplicantExample example);

```

```

10
11  int deleteByExample(ApplicantExample example);
12
13  int deleteByPrimaryKey(Integer acId);
14
15  int insert(Applicant record);
16
17  int insertSelective(Applicant record);
18
19  List<Applicant> selectByExample(ApplicantExample example);
20
21  Applicant selectByPrimaryKey(Integer acId);
22
23  int updateByExampleSelective(@Param("record") Applicant
    record, @Param("example") ApplicantExample example);
24
25  int updateByExample(@Param("record") Applicant record,
    @Param("example") ApplicantExample example);
26
27  int updateByPrimaryKeySelective(Applicant record);
28
29  int updateByPrimaryKey(Applicant record);
30 }

```

ProjectinfoMapper.java

```

1  package com.yuntu.curd.dao;
2
3  import com.yuntu.curd.bean.Projectinfo;
4  import com.yuntu.curd.bean.ProjectinfoExample;
5  import java.util.List;
6  import org.apache.ibatis.annotations.Param;
7
8  public interface ProjectinfoMapper {
9      long countByExample(ProjectinfoExample example);
10
11     int deleteByExample(ProjectinfoExample example);

```

```
12
13  int deleteByPrimaryKey(Integer piId);
14
15  int insert(Projectinfo record);
16
17  int insertSelective(Projectinfo record);
18
19  /**
20   * 根据条件查询所有，两表联查，同时查询出申报项目的项目和对应的申报人
21   * @param example
22   * @return
23   */
24  List<Projectinfo> selectByExampleWithApplicantion(ProjectinfoExample example);
25
26  /**
27   * 根据id查询所有，两表联查，同时查询出单个申报项目的项目和对应的申报人
28   * @param piId
29   * @return
30   */
31  Projectinfo selectByPrimaryKeyWithApplicantion(Integer piId);
32
33  List<Projectinfo> selectByExample(ProjectinfoExample example);
34
35  Projectinfo selectByPrimaryKey(Integer piId);
36
37  int updateByExampleSelective(@Param("record") Projectinfo record, @Param("example") ProjectinfoExample example);
38
39  int updateByExample(@Param("record") Projectinfo record, @Param("example") ProjectinfoExample example);
40
41  int updateByPrimaryKeySelective(Projectinfo record);
42
43  int updateByPrimaryKey(Projectinfo record);
```

业务逻辑层：com.yuntu.curd.service

ApplicantService.java

```
1 package com.yuntu.curd.service;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7
8 import com.yuntu.curd.bean.Applicant;
9 import com.yuntu.curd.dao.ApplicantMapper;
10
11 /**
12  * 申请人业务层
13  * @author 阿华
14  *
15  */
16 @Service
17 public class ApplicantService {
18
19     @Autowired
20     private ApplicantMapper applicantMapper;
21
22     /**
23      * 业务查询所有的申报人
24      * @return
25      */
26     public List<Applicant> getApplicants() {
27         return applicantMapper.selectByExample(null);
28     }
29 }
30
```

ProjectInfoService.java

```
1 package com.yuntu.curd.service;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7
8 import com.yuntu.curd.bean.Projectinfo;
9 import com.yuntu.curd.bean.ProjectinfoExample;
10 import com.yuntu.curd.bean.ProjectinfoExample.Criteria;
11 import com.yuntu.curd.dao.ProjectinfoMapper;
12
13 /**
14  * 项目表业务处理层
15  * @author 阿华
16  *
17  */
18 @Service
19 public class ProjectInfoService {
20     @Autowired
21     ProjectinfoMapper projectInfoMapper;
22
23     /**
24      * 查询所有项目信息
25      * @return
26      */
27     public List<Projectinfo> getAll() {
28         return
29         projectInfoMapper.selectByExampleWithApplicantion(null);
30     }
31
32     /**
33      * 保存项目信息
34      * @param projectInfo
35      * @return
36      */
37 }
```

```
35  */
36  public void saveProjectInfo(Projectinfo projectInfo) {
37  projectInfoMapper.insertSelective(projectInfo);
38  }
39
40
41  /**
42  *检查用户是否存在
43  */
44  public boolean checkProjectInfo(String projectName) {
45  //创建查询条件
46  ProjectinfoExample example = new ProjectinfoExample();
47  //得到条件对象
48  Criteria Criteria = example.createCriteria();
49  Criteria.andPiProjectnameEqualTo(projectName);
50  //根据条件查询数量
51  long count = projectInfoMapper.countByExample(example);
52  return count==0;
53  }
54
55  /**
56  * 修改时，根据pi_Id查询出单个项目信息
57  * @param pi_Id
58  * @return
59  */
60  public Projectinfo getProjectInfo(Integer pi_Id) {
61  return projectInfoMapper.selectByPrimaryKeyWithApplicantion(pi_Id);
62  }
63
64  /**
65  * 修改项目信息（根据id有选择的更新）
66  * @param projectInfo
67  */
68  public void updateProjectInfo(Projectinfo projectInfo) {
69  projectInfoMapper.updateByPrimaryKeySelective(projectInfo);
```



```

70     }
71
72     /**
73     * 根据单个id删除项目信息
74     * @param id
75     */
76     public void deleteProjectInfo(Integer id) {
77         projectInfoMapper.deleteByPrimaryKey(id);
78     }
79
80     /**
81     * 根据多个id删除项目信息
82     * @param ids
83     */
84     public void deletePrijectInfos(List<Integer> ids) {
85         ProjectinfoExample example = new ProjectinfoExample();
86         Criteria Criteria = example.createCriteria();
87         Criteria.andPiIdIn(ids);
88         projectInfoMapper.deleteByExample(example);
89     }
90 }
91

```

测试: com.yuntu.curd.test

MapperTest.java

```

1  package com.yuntu.curd.test;
2
3  import java.text.ParseException;
4  import java.text.SimpleDateFormat;
5  import java.util.Date;
6  import java.util.UUID;
7
8  import org.apache.ibatis.session.SqlSession;
9  import org.junit.Test;
10 import org.junit.runner.RunWith;

```

```
11 import org.springframework.beans.factory.annotation.Autowired;
12 import org.springframework.test.context.ContextConfiguration;
13 import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
14
15 import com.yuntu.curd.bean.Applicant;
16 import com.yuntu.curd.bean.Projectinfo;
17 import com.yuntu.curd.dao.ApplicantMapper;
18 import com.yuntu.curd.dao.ProjectinfoMapper;
19
20 /**
21  * 测试dao的任务
22  * @author 阿华
23  * 推荐spring项目就可以spring提供的单元测试，可以自动注入我们需要的组件
24  * 步骤：
25  * 1.导入spring测试模块
26  * 2.@ContextConfiguration指定spring配置文件位置
27  * 3.直接@Autowired自动使用组件
28  */
29 @RunWith(SpringJUnit4ClassRunner.class)
30 @ContextConfiguration(locations= {"classpath:applicationContext.xml"})
31 public class MapperTest {
32     @Autowired
33     ApplicantMapper applicantMapper;
34     @Autowired
35     ProjectinfoMapper projectinfoMapper;
36     //批量操作
37     @Autowired
38     SqlSession sqlSession;
39
40     @Test
41     public void testCRUD() throws ParseException {
42         /**
43          * 原始版本的测试版本
```

```

44  * //1、创建IOC容器
45  * ApplicationContext ioc = new
  ClassPathXmlApplicationContext();
46  * //2、从容器中获取mapper
47  * ApplicantExample bean = ioc.getBean(ApplicantExample.class);
48  */
49  System.out.println("-----" + applicantMapper);
50  // Date date = new SimpleDateFormat("yyyy-MM-dd").parse("2000-1
  1-11");
51  // //插入申报人
52  // applicantMapper.insertSelective(new Applicant(null,"andi",da
  te,1,5));
53
54  /*
55  * 批量插入多个；使用批量操作的sqlSession
56  * 在applicationContext.xml已经注入了SQLSession对象
57  */
58  //创建用于批量添加的mapper接口对象
59  ProjectinfoMapper mapper = sqlSession.getMapper(ProjectinfoMap
  per.class);
60  //循环操作
61  for (int i = 0; i < 50; i++) {
62  //自动生成字符串
63  String proname=UUID.randomUUID().toString().substring(0, 5)+"原
  子弹抛光"+i;
64  mapper.insertSelective(new Projectinfo(null, proname,
65  new SimpleDateFormat("yyyy-MM-dd").parse((1990+i)+"-11-11"),
66  new SimpleDateFormat("yyyy-MM-dd").parse((1999+i)+"-12-12"),
67  i%3, i%7+1, null));
68  }
69  System.out.println("批量添加完成");
70  }
71  }
72

```

```

1 package com.yuntu.curd.test;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.sql.SQLException;
6 import java.util.ArrayList;
7 import java.util.List;
8 import org.mybatis.generator.api.MyBatisGenerator;
9 import org.mybatis.generator.config.Configuration;
10 import org.mybatis.generator.config.xml.ConfigurationParser;
11 import org.mybatis.generator.exception.InvalidConfigurationException;
12 import org.mybatis.generator.exception.XMLParserException;
13 import org.mybatis.generator.internal.DefaultShellCallback;
14
15 public class MBGTest {
16     public static void main(String[] args) throws IOException, XML
17     ParserException, InvalidConfigurationException, SQLException, Inte
18     rruptedException {
19         List<String> warnings = new ArrayList<String>();
20         boolean overwrite = true;
21         File configFile = new File("mbg.xml");
22         ConfigurationParser cp = new ConfigurationParser(warnings);
23         Configuration config = cp.parseConfiguration(configFile);
24         DefaultShellCallback callback = new DefaultShellCallback(overw
25         rite);
26         MyBatisGenerator myBatisGenerator = new MyBatisGenerator(confi
27         g, callback, warnings);
28         myBatisGenerator.generate(null);
29     }
30 }

```

MVCTest.java

```

1 package com.yuntu.curd.test;
2

```

```
3 import java.util.List;
4
5 import org.junit.Before;
6 import org.junit.Test;
7 import org.junit.runner.RunWith;
8 import org.springframework.beans.factory.annotation.Autowired;
9 import org.springframework.mock.web.MockHttpServletRequest;
10 import org.springframework.test.context.ContextConfiguration;
11 import org.springframework.test.context.junit4.SpringJUnit4Class
    sRunner;
12 import
    org.springframework.test.context.web.WebAppConfiguration;
13 import org.springframework.test.web.servlet.MockMvc;
14 import org.springframework.test.web.servlet.MvcResult;
15 import org.springframework.test.web.servlet.request.MockMvcRequ
    estBuilders;
16 import org.springframework.test.web.servlet.setup.MockMvcBuilde
    rs;
17 import org.springframework.web.context.WebApplicationContext;
18
19 import com.github.pagehelper.PageInfo;
20 import com.yuntu.curd.bean.Projectinfo;
21
22 /**
23  * 使用spring提供的单元测试组件
24  * 步骤:
25  * 使用spring模块发送虚拟请求，测试页面的curd操作
26  * @author 阿华
27  *
28  */
29 @RunWith(SpringJUnit4ClassRunner.class)
30 @ContextConfiguration(locations = {"classpath:applicationContex
    t.xml",
31     "file:src/main/webapp/WEB-INF/dispatcherServlet-servlet.xml"})
32 @WebAppConfiguration
33 public class MVCTest {
```

```
34 // 传入springmvc的容器
35 @Autowired
36 WebApplicationContext context;
37 // 虚拟mvc请求,获取处理结果
38 MockMvc mockMvc;
39
40 // 每次请求前都要初始化构建mockMvc
41 @Before
42 public void initMockMvc() {
43     mockMvc = MockMvcBuilders.webAppContextSetup(context).build();
44 }
45
46 @Test
47 public void testPage() throws Exception {
48     /*
49      * MockMvcRequestBuilders.get("/projectInfos"):使用springmvc发送
      请求信息,而且是get请求
50      * param("pn", "7"): 携带一个参数 params(arg0): 携带多个参数
51      */
52     MvcResult result =
53         mockMvc.perform(MockMvcRequestBuilders.get("/projectInfos").param(
54             "n", "5")).andReturn();
55     // 请求成功之后,请求域就会有pageinfo,可以取出
56     MockHttpServletRequest request = result.getRequest();
57     // 从请求中拿到PageInfo信息
58     PageInfo pe = (PageInfo) request.getAttribute("PageInfo");
59     //输出验证
60     System.out.println("当前页" + pe.getPageNum());
61     System.out.println("页容量" + pe.getPageSize());
62     System.out.println("起始数" + pe.getStartRow());
63     System.out.println("结束数" + pe.getEndRow());
64     System.out.println("总条数" + pe.getTotal());
65     System.out.println("总页数" + pe.getPages());
66     System.out.println("当前应该连续展示的页码");
67     int[] nums = pe.getNavigatepageNums();
68     for (int i : nums) {
```

```

67 System.out.print(" " + i);
68 }
69 System.out.println();
70 // 获取当前页的项目信息
71 List<Projectinfo> list = pe.getList();
72 for (Projectinfo projectInfo : list) {
73 System.out.println("ID:" + projectInfo.getPiId() + "\t项目名" +
projectInfo.getPiProjectname() + "\t申报人"
74 + projectInfo.getApplicant().getAcName());
75 }
76 }
77 }
78

```

src/main/resources

ApplicantMapper.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "ht
tp://mybatis.org/dtd/mybatis-3-mapper.dtd">
3 <mapper namespace="com.yuntu.curd.dao.ApplicantMapper">
4 <resultMap id="BaseResultMap" type="com.yuntu.curd.bean.Applica
nt">
5 <id column="ac_id" jdbcType="INTEGER" property="acId" />
6 <result column="ac_name" jdbcType="VARCHAR" property="acName"
/>
7 <result column="ac_birthdate" jdbcType="TIMESTAMP" property="acB
irthday" />
8 <result column="ac_sex" jdbcType="INTEGER" property="acSex" />
9 <result column="workingLife" jdbcType="INTEGER" property="worki
nglife" />
10 </resultMap>
11 <sql id="Example_Where_Clause">
12 <where>
13 <foreach collection="oredCriteria" item="criteria"
separator="or">
14 <if test="criteria.valid">
15 <trim prefix="(" prefixOverrides="and" suffix=")">

```

```
16 <foreach collection="criteria.criteria" item="criterion">
17   <choose>
18     <when test="criterion.noValue">
19       and ${criterion.condition}
20     </when>
21     <when test="criterion.singleValue">
22       and ${criterion.condition} #{criterion.value}
23     </when>
24     <when test="criterion.betweenValue">
25       and ${criterion.condition} #{criterion.value} and #
        {criterion.secondValue}
26     </when>
27     <when test="criterion.listValue">
28       and ${criterion.condition}
29       <foreach close=")" collection="criterion.value" item="listItem"
        open="(" separator=",">
30         #{listItem}
31       </foreach>
32     </when>
33   </choose>
34 </foreach>
35 </trim>
36 </if>
37 </foreach>
38 </where>
39 </sql>
40 <sql id="Update_By_Example_Where_Clause">
41   <where>
42     <foreach collection="example.oredCriteria" item="criteria" separator="or">
43       <if test="criteria.valid">
44         <trim prefix="(" prefixOverrides="and" suffix=")">
45           <foreach collection="criteria.criteria" item="criterion">
46             <choose>
47               <when test="criterion.noValue">
48                 and ${criterion.condition}
```



```
49 </when>
50 <when test="criterion.singleValue">
51   and ${criterion.condition} #{criterion.value}
52 </when>
53 <when test="criterion.betweenValue">
54   and ${criterion.condition} #{criterion.value} and #
    {criterion.secondValue}
55 </when>
56 <when test="criterion.listValue">
57   and ${criterion.condition}
58   <foreach close=")" collection="criterion.value" item="listItem"
    open="(" separator=",">
59     #{listItem}
60   </foreach>
61 </when>
62 </choose>
63 </foreach>
64 </trim>
65 </if>
66 </foreach>
67 </where>
68 </sql>
69 <sql id="Base_Column_List">
70   ac_id, ac_name, ac_birthday, ac_sex, workingLife
71 </sql>
72 <select id="selectByExample" parameterType="com.yuntu.curd.bean.ApplicantExample" resultMap="BaseResultMap">
73   select
74   <if test="distinct">
75     distinct
76   </if>
77   <include refid="Base_Column_List" />
78   from applicant
79   <if test="_parameter != null">
80     <include refid="Example_Where_Clause" />
81   </if>
```

```

82  <if test="orderByClause != null">
83    order by ${orderByClause}
84  </if>
85  </select>
86  <select id="selectByPrimaryKey" parameterType="java.lang.Integer" resultMap="BaseResultMap">
87    select
88    <include refid="Base_Column_List" />
89    from applicant
90    where ac_id = #{acId,jdbcType=INTEGER}
91  </select>
92  <delete id="deleteByPrimaryKey" parameterType="java.lang.Integer">
93    delete from applicant
94    where ac_id = #{acId,jdbcType=INTEGER}
95  </delete>
96  <delete id="deleteByExample" parameterType="com.yuntu.curd.bean.ApplicantExample">
97    delete from applicant
98    <if test="_parameter != null">
99      <include refid="Example_Where_Clause" />
100    </if>
101  </delete>
102  <insert id="insert" parameterType="com.yuntu.curd.bean.Applicant">
103    insert into applicant (ac_id, ac_name, ac_birthday,
104      ac_sex, workingLife)
105    values (#{acId,jdbcType=INTEGER}, #{acName,jdbcType=VARCHAR},
106      #{acBirthday,jdbcType=TIMESTAMP},
107      #{acSex,jdbcType=INTEGER}, #{workinglife,jdbcType=INTEGER})
108  </insert>
109  <insert id="insertSelective" parameterType="com.yuntu.curd.bean.Applicant">
110    insert into applicant
111    <trim prefix="(" suffix=")" suffixOverrides=",">
112      <if test="acId != null">
113        ac_id,

```

```
113 </if>
114 <if test="acName != null">
115   ac_name,
116 </if>
117 <if test="acBirthday != null">
118   ac_birthday,
119 </if>
120 <if test="acSex != null">
121   ac_sex,
122 </if>
123 <if test="workinglife != null">
124   workingLife,
125 </if>
126 </trim>
127 <trim prefix="values (" suffix=")" suffixOverrides=",">
128   <if test="acId != null">
129     #{acId,jdbcType=INTEGER},
130   </if>
131   <if test="acName != null">
132     #{acName,jdbcType=VARCHAR},
133   </if>
134   <if test="acBirthday != null">
135     #{acBirthday,jdbcType=TIMESTAMP},
136   </if>
137   <if test="acSex != null">
138     #{acSex,jdbcType=INTEGER},
139   </if>
140   <if test="workinglife != null">
141     #{workinglife,jdbcType=INTEGER},
142   </if>
143 </trim>
144 </insert>
145 <select id="countByExample" parameterType="com.yuntu.curd.bea
n.ApplicantExample" resultType="java.lang.Long">
146   select count(*) from applicant
```

```
147 <if test="_parameter != null">
148 <include refid="Example_Where_Clause" />
149 </if>
150 </select>
151 <update id="updateByExampleSelective" parameterType="map">
152 update applicant
153 <set>
154 <if test="record.acId != null">
155 ac_id = #{record.acId,jdbcType=INTEGER},
156 </if>
157 <if test="record.acName != null">
158 ac_name = #{record.acName,jdbcType=VARCHAR},
159 </if>
160 <if test="record.acBirthday != null">
161 ac_birthday = #{record.acBirthday,jdbcType=TIMESTAMP},
162 </if>
163 <if test="record.acSex != null">
164 ac_sex = #{record.acSex,jdbcType=INTEGER},
165 </if>
166 <if test="record.workinglife != null">
167 workingLife = #{record.workinglife,jdbcType=INTEGER},
168 </if>
169 </set>
170 <if test="_parameter != null">
171 <include refid="Update_By_Example_Where_Clause" />
172 </if>
173 </update>
174 <update id="updateByExample" parameterType="map">
175 update applicant
176 set ac_id = #{record.acId,jdbcType=INTEGER},
177 ac_name = #{record.acName,jdbcType=VARCHAR},
178 ac_birthday = #{record.acBirthday,jdbcType=TIMESTAMP},
179 ac_sex = #{record.acSex,jdbcType=INTEGER},
180 workingLife = #{record.workinglife,jdbcType=INTEGER}
181 <if test="_parameter != null">
```

```

182 <include refid="Update_By_Example_Where_Clause" />
183 </if>
184 </update>
185 <update id="updateByPrimaryKeySelective" parameterType="com.yuntu.curd.bean.Applicant">
186     update applicant
187     <set>
188     <if test="acName != null">
189         ac_name = #{acName,jdbcType=VARCHAR},
190     </if>
191     <if test="acBirthday != null">
192         ac_birthday = #{acBirthday,jdbcType=TIMESTAMP},
193     </if>
194     <if test="acSex != null">
195         ac_sex = #{acSex,jdbcType=INTEGER},
196     </if>
197     <if test="workinglife != null">
198         workingLife = #{workinglife,jdbcType=INTEGER},
199     </if>
200     </set>
201     where ac_id = #{acId,jdbcType=INTEGER}
202 </update>
203 <update id="updateByPrimaryKey" parameterType="com.yuntu.curd.bean.Applicant">
204     update applicant
205     set ac_name = #{acName,jdbcType=VARCHAR},
206     ac_birthday = #{acBirthday,jdbcType=TIMESTAMP},
207     ac_sex = #{acSex,jdbcType=INTEGER},
208     workingLife = #{workinglife,jdbcType=INTEGER}
209     where ac_id = #{acId,jdbcType=INTEGER}
210 </update>
211 </mapper>

```

ProjectinfoMapper.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>

```

```
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "ht
tp://mybatis.org/dtd/mybatis-3-mapper.dtd">
3 <mapper namespace="com.yuntu.curd.dao.ProjectinfoMapper">
4   <resultMap id="BaseResultMap"
5     type="com.yuntu.curd.bean.Projectinfo">
6     <id column="pi_id" jdbcType="INTEGER" property="piId" />
7     <result column="pi_projectName" jdbcType="VARCHAR"
8       property="piProjectname" />
9     <result column="pi_startDate" jdbcType="TIMESTAMP"
10      property="piStartdate" />
11     <result column="pi_endDate" jdbcType="TIMESTAMP"
12      property="piEnddate" />
13     <result column="pi_status" jdbcType="INTEGER"
14      property="piStatus" />
15     <result column="acId" jdbcType="INTEGER" property="acid" />
16   </resultMap>
17   <sql id="Example_Where_Clause">
18     <where>
19       <foreach collection="oredCriteria" item="criteria"
20         separator="or">
21         <if test="criteria.valid">
22           <trim prefix="(" prefixOverrides="and" suffix=")">
23             <foreach collection="criteria.criteria" item="criterion">
24               <choose>
25                 <when test="criterion.noValue">
26                   and ${criterion.condition}
27                 </when>
28                 <when test="criterion.singleValue">
29                   and ${criterion.condition} #{criterion.value}
30                 </when>
31                 <when test="criterion.betweenValue">
32                   and ${criterion.condition} #{criterion.value}
33                   and
34                   #{criterion.secondValue}
35                 </when>
```

```
36 <when test="criterion.listValue">
37   and ${criterion.condition}
38   <foreach close=")" collection="criterion.value"
39   item="listItem" open="(" separator=",">
40     #{listItem}
41   </foreach>
42 </when>
43 </choose>
44 </foreach>
45 </trim>
46 </if>
47 </foreach>
48 </where>
49 </sql>
50 <sql id="Update_By_Example_Where_Clause">
51   <where>
52     <foreach collection="example.oredCriteria" item="criteria"
53     separator="or">
54       <if test="criteria.valid">
55         <trim prefix="(" prefixOverrides="and" suffix=")">
56           <foreach collection="criteria.criteria" item="criterion">
57             <choose>
58               <when test="criterion.noValue">
59                 and ${criterion.condition}
60               </when>
61               <when test="criterion.singleValue">
62                 and ${criterion.condition} #{criterion.value}
63               </when>
64               <when test="criterion.betweenValue">
65                 and ${criterion.condition} #{criterion.value}
66                 and
67                 #{criterion.secondValue}
68               </when>
69               <when test="criterion.listValue">
70                 and ${criterion.condition}
```

```
71 <foreach close=")" collection="criterion.value"
72 item="listItem" open="(" separator=",">
73   #{listItem}
74 </foreach>
75 </when>
76 </choose>
77 </foreach>
78 </trim>
79 </if>
80 </foreach>
81 </where>
82 </sql>
83 <sql id="Base_Column_List">
84   pi_id, pi_projectName, pi_startDate, pi_endDate, pi_status, acI
85   d
86 </sql>
87
88
89 <!-- *****自定义的sql查询开始***** -->
90 <!-- sql执行返回的结果对象 -->
91 <resultMap id="WithApplicantResultMap" type="com.yuntu.curd.bea
92   an.Projectinfo">
93   <id column="pi_id" jdbcType="INTEGER" property="piId" />
94   <result column="pi_projectName" jdbcType="VARCHAR" property="p
95     iProjectname" />
96   <result column="pi_startDate" jdbcType="TIMESTAMP" property="p
97     iStartdate" />
98   <result column="pi_endDate" jdbcType="TIMESTAMP" property="piE
99     nddate" />
100  <result column="pi_status" jdbcType="INTEGER" property="piStat
101    us" />
102  <result column="acId" jdbcType="INTEGER" property="acid" />
103  <!-- 关联申报人 -->
104  <association property="applicant" javaType="com.yuntu.curd.bea
105    n.Applicant">
106    <id column="ac_id" jdbcType="INTEGER" property="acId" />
```



```

101 <result column="ac_name" jdbcType="VARCHAR" property="acName"
    />
102 <result column="ac_birthday" jdbcType="TIMESTAMP" property="ac
    Birthday" />
103 <result column="ac_sex" jdbcType="INTEGER" property="acSex" />
104 <result column="workingLife" jdbcType="INTEGER" property="work
    inglife" />
105 </association>
106 </resultMap>
107 <!-- 联合查询所有的两表所有信息 -->
108 <sql id="WithApplicant_Column_List">
109     p.pi_id, p.pi_projectName, p.pi_startDate, p.pi_endDate, p.pi
    status, p.acId , a.ac_id , a.ac_name , a.ac_birthday , a.ac_sex ,
    a.workingLife
110 </sql>
111 <!-- 联合查询两张表 -->
112 <select id="selectByExampleWithApplicantion" parameterType="co
    m.yuntu.curd.bean.ProjectinfoExample" resultMap="WithApplicantResu
    ltMap">
113     select
114     <if test="distinct">
115         distinct
116     </if>
117     <include refid="WithApplicant_Column_List" />
118     from projectinfo p
119     left join applicant a on p.acId=a.ac_id
120     <if test="_parameter != null">
121         <include refid="Example_Where_Clause" />
122     </if>
123     <if test="orderByClause != null">
124         order by ${orderByClause}
125     </if>
126 </select>
127
128 <select id="selectByPrimaryKeyWithApplicantion"
    parameterType="java.lang.Integer" resultMap="WithApplicantResultMa
    p">
129     select

```

```
130 <include refid="WithApplicant_Column_List" />
131 from projectinfo p
132 left join applicant a on p.acId=a.ac_id
133 where pi_id = #{piId,jdbcType=INTEGER}
134 </select>
135 <!-- *****自定义的sql查询结束***** -->
136
137
138
139 <select id="selectByExample"
140 parameterType="com.yuntu.curd.bean.ProjectinfoExample"
141 resultMap="BaseResultMap">
142 select
143 <if test="distinct">
144 distinct
145 </if>
146 <include refid="Base_Column_List" />
147 from projectinfo
148 <if test="_parameter != null">
149 <include refid="Example_Where_Clause" />
150 </if>
151 <if test="orderByClause != null">
152 order by ${orderByClause}
153 </if>
154 </select>
155 <select id="selectByPrimaryKey"
156 parameterType="java.lang.Integer" resultMap="BaseResultMap">
157 select
158 <include refid="Base_Column_List" />
159 from projectinfo
160 where pi_id = #{piId,jdbcType=INTEGER}
161 </select>
162 <delete id="deleteByPrimaryKey"
163 parameterType="java.lang.Integer">
164 delete from projectinfo
```

```
165 where pi_id =
166 #{piId,jdbcType=INTEGER}
167 </delete>
168 <delete id="deleteByExample"
169 parameterType="com.yuntu.curd.bean.ProjectinfoExample">
170 delete from projectinfo
171 <if test="_parameter != null">
172 <include refid="Example_Where_Clause" />
173 </if>
174 </delete>
175 <insert id="insert"
176 parameterType="com.yuntu.curd.bean.Projectinfo">
177 insert into projectinfo (pi_id, pi_projectName,
178 pi_startDate,
179 pi_endDate, pi_status, acId
180 )
181 values
182 (#{piId,jdbcType=INTEGER}, #{piProjectname,jdbcType=VARCHAR},
183 #{piStartdate,jdbcType=TIMESTAMP},
184 #{piEnddate,jdbcType=TIMESTAMP},
185 #{piStatus,jdbcType=INTEGER}, #{acid,jdbcType=INTEGER}
186 )
187 </insert>
188 <insert id="insertSelective"
189 parameterType="com.yuntu.curd.bean.Projectinfo">
190 insert into projectinfo
191 <trim prefix="(" suffix=")" suffixOverrides=",">
192 <if test="piId != null">
193 pi_id,
194 </if>
195 <if test="piProjectname != null">
196 pi_projectName,
197 </if>
198 <if test="piStartdate != null">
199 pi_startDate,
```

```
200 </if>
201 <if test="piEnddate != null">
202     pi_endDate,
203 </if>
204 <if test="piStatus != null">
205     pi_status,
206 </if>
207 <if test="acid != null">
208     acId,
209 </if>
210 </trim>
211 <trim prefix="values (" suffix=")" suffixOverrides=",">
212     <if test="piId != null">
213         #{piId,jdbcType=INTEGER},
214     </if>
215     <if test="piProjectname != null">
216         #{piProjectname,jdbcType=VARCHAR},
217     </if>
218     <if test="piStartdate != null">
219         #{piStartdate,jdbcType=TIMESTAMP},
220     </if>
221     <if test="piEnddate != null">
222         #{piEnddate,jdbcType=TIMESTAMP},
223     </if>
224     <if test="piStatus != null">
225         #{piStatus,jdbcType=INTEGER},
226     </if>
227     <if test="acid != null">
228         #{acid,jdbcType=INTEGER},
229     </if>
230 </trim>
231 </insert>
232 <select id="countByExample"
233     parameterType="com.yuntu.curd.bean.ProjectinfoExample"
234     resultType="java.lang.Long">
```

```
235 select count(*) from projectinfo
236 <if test="_parameter != null">
237 <include refid="Example_Where_Clause" />
238 </if>
239 </select>
240 <update id="updateByExampleSelective" parameterType="map">
241 update projectinfo
242 <set>
243 <if test="record.piId != null">
244 pi_id = #{record.piId,jdbcType=INTEGER},
245 </if>
246 <if test="record.piProjectname != null">
247 pi_projectName =
248 #{record.piProjectname,jdbcType=VARCHAR},
249 </if>
250 <if test="record.piStartdate != null">
251 pi_startDate = #{record.piStartdate,jdbcType=TIMESTAMP},
252 </if>
253 <if test="record.piEnddate != null">
254 pi_endDate = #{record.piEnddate,jdbcType=TIMESTAMP},
255 </if>
256 <if test="record.piStatus != null">
257 pi_status = #{record.piStatus,jdbcType=INTEGER},
258 </if>
259 <if test="record.acid != null">
260 acId = #{record.acid,jdbcType=INTEGER},
261 </if>
262 </set>
263 <if test="_parameter != null">
264 <include refid="Update_By_Example_Where_Clause" />
265 </if>
266 </update>
267 <update id="updateByExample" parameterType="map">
268 update projectinfo
269 set pi_id = #{record.piId,jdbcType=INTEGER},
```

```
270 pi_projectName = #{record.piProjectname,jdbcType=VARCHAR},
271 pi_startDate = #{record.piStartdate,jdbcType=TIMESTAMP},
272 pi_endDate =
273 #{record.piEnddate,jdbcType=TIMESTAMP},
274 pi_status =
275 #{record.piStatus,jdbcType=INTEGER},
276 acId =
277 #{record.acid,jdbcType=INTEGER}
278 <if test="_parameter != null">
279 <include refid="Update_By_Example_Where_Clause" />
280 </if>
281 </update>
282 <update id="updateByPrimaryKeySelective"
283 parameterType="com.yuntu.curd.bean.Projectinfo">
284 update projectinfo
285 <set>
286 <if test="piProjectname != null">
287 pi_projectName = #{piProjectname,jdbcType=VARCHAR},
288 </if>
289 <if test="piStartdate != null">
290 pi_startDate = #{piStartdate,jdbcType=TIMESTAMP},
291 </if>
292 <if test="piEnddate != null">
293 pi_endDate = #{piEnddate,jdbcType=TIMESTAMP},
294 </if>
295 <if test="piStatus != null">
296 pi_status = #{piStatus,jdbcType=INTEGER},
297 </if>
298 <if test="acid != null">
299 acId = #{acid,jdbcType=INTEGER},
300 </if>
301 </set>
302 where pi_id = #{piId,jdbcType=INTEGER}
303 </update>
304 <update id="updateByPrimaryKey"
```

```

305 parameterType="com.yuntu.curd.bean.Projectinfo">
306 update projectinfo
307 set pi_projectName =
308 #{piProjectname,jdbcType=VARCHAR},
309 pi_startDate =
310 #{piStartdate,jdbcType=TIMESTAMP},
311 pi_endDate =
312 #{piEnddate,jdbcType=TIMESTAMP},
313 pi_status =
314 #{piStatus,jdbcType=INTEGER},
315 acId = #{acid,jdbcType=INTEGER}
316 where
317 pi_id = #{piId,jdbcType=INTEGER}
318 </update>
319 </mapper>

```

applicationContext.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:p="http://www.springframework.org/schema/p"
5   xmlns:context="http://www.springframework.org/schema/context"
6   xmlns:mvc="http://www.springframework.org/schema/mvc"
7   xmlns:aop="http://www.springframework.org/schema/aop"
8   xmlns:tx="http://www.springframework.org/schema/tx"
9   xmlns:task="http://www.springframework.org/schema/task"
10  xsi:schemaLocation="
11    http://www.springframework.org/schema/beans
12    http://www.springframework.org/schema/beans/spring-beans-4.0.x
13    sd
14    http://www.springframework.org/schema/context
15    http://www.springframework.org/schema/context/spring-context-
16    4.0.xsd
17    http://www.springframework.org/schema/mvc
18    http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd
19    http://www.springframework.org/schema/aop

```

```
18 http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
19 http://www.springframework.org/schema/tx
20 http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
21 http://www.springframework.org/schema/task
22 http://www.springframework.org/schema/task/spring-task-4.0.xsd"
23
24 <!-- Spring的配置文件,主要配置业务逻辑相关的内容 -->
25 <context:component-scan base-package="com.yuntu">
26 <!-- 除了控制器,其他都扫描 -->
27 <context:exclude-filter type="annotation"
28 expression="org.springframework.stereotype.Controller" />
29 </context:component-scan>
30
31 <!-- *****数据源***** -->
32 <!-- 数据源(默认直接配置在applicationContext.xml文件中;放在database.properties文件中; JNDI) -->
33 <context:property-placeholder location="classpath:dbconfig.properties"/>
34 <bean id="pooledDataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">
35 <!-- URL -->
36 <property name="jdbcUrl" value="${jdbc.jdbcUrl}"></property>
37 <!-- 驱动 -->
38 <property name="driverClass" value="${jdbc.driverClass}"></property>
39 <!-- 账号 -->
40 <property name="user" value="${jdbc.user}"></property>
41 <!-- 密码 -->
42 <property name="password" value="${jdbc.password}"></property>
43 </bean>
44 <!-- *****数据源***** -->
45
46
47 <!-- =====配置mybatis的整合===== -->
48 <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
```



```

49  <!-- 指定mybatis全局配置文件的位置 -->
50  <property name="configLocation" value="classpath:mybatis-config.xml"></property>
51  <!-- 指定数据源的使用 -->
52  <property name="dataSource" ref="pooledDataSource"></property>
53  <!-- 指定mybatis, mapper文件所在的位置(类路径下的所有的映射文件) -->
54  <property name="mapperLocations" value="classpath:mapper/*.xml"></property>
55  </bean>
56
57  <!-- 配置扫描器,将mybatis接口的实现加入到Ioc容器中(接口实现类) -->
58  <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
59  <property name="basePackage" value="com.yuntu.curd.dao"></property>
60  </bean>
61
62  <!-- 配置一个可以批量执行的sqlSession -->
63  <bean id="sSqlSession" class="org.mybatis.spring.SqlSessionTemplate">
64  <!-- 指定sqlSession会话工厂 -->
65  <constructor-arg name="sqlSessionFactory" ref="sqlSessionFactory"></constructor-arg>
66  <!-- mybatis默认使用单个更新（executorType默认为SIMPLE而不是BATCH）
67  SIMPLE：单个模式
68  BATCH：批量模式
69  -->
70  <constructor-arg name="executorType" value="BATCH"></constructor-arg>
71  </bean>
72  <!-- =====配置mybatis的整合=====* -->
73
74  <!-- *****事务控制配置***** -->
75  <bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
76  <!-- 添加数据源 -->

```

```

77 <property name="dataSource" ref="pooledDataSource"></property>
78 </bean>
79
80 <!-- 配置事务增强 -->
81 <tx:advice id="txAdvice" transaction-manager="transactionManager"
82 <tx:attributes>
83 <!-- 所有的方法都是事务方法 -->
84 <tx:method name="*" />
85 <!-- eg:以get开头的方法 -->
86 <tx:method name="get*" read-only="true" />
87 </tx:attributes>
88 </tx:advice>
89
90 <!-- 1.开启基于注解的事务 2.使用xml配置的形式的事务(一般情况下, 比较重要的都是使用配置的事务) -->
91 <aop:config>
92 <!-- 配置切入点 -->
93 <aop:pointcut expression="execution(* com.yuntu.curd.service.*(..))" id="txPoint" />
94 <!-- 事务增强 -->
95 <aop:advisor advice-ref="txAdvice" pointcut-ref="txPoint" />
96 </aop:config>
97
98 <!-- *****事务控制配置***** -->
99
100 </beans>

```

dbconfig.properties

```

1 jdbc.url=jdbc:mysql://127.0.0.1/dpm?useUnicode=true&characterEncoding=utf-8
2 jdbc.driverClass=com.mysql.jdbc.Driver
3 jdbc.user=root
4 jdbc.password=123456

```

mybatis-config.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>

```

```

2 <!DOCTYPE configuration
3   PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4   "http://mybatis.org/dtd/mybatis-3-config.dtd">
5
6 <!-- mybatis配置文件 -->
7 <configuration>
8   <!-- 设置（驼峰命名规则） -->
9   <settings>
10    <setting name="mapUnderscoreToCamelCase" value="true"/>
11  </settings>
12
13  <!-- 类型别名（实体类包） -->
14  <typeAliases>
15    <package name="com.yuntu.curd.bean"/>
16  </typeAliases>
17
18  <!-- 引入分页插件 -->
19  <plugins>
20    <plugin interceptor="com.github.pagehelper.PageInterceptor">
21    </plugin>
22  </plugins>
23 </configuration>

```

dispatcherServlet-servlet.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:p="http://www.springframework.org/schema/p"
5   xmlns:context="http://www.springframework.org/schema/context"
6   xmlns:mvc="http://www.springframework.org/schema/mvc"
7   xmlns:aop="http://www.springframework.org/schema/aop"
8   xmlns:tx="http://www.springframework.org/schema/tx"
9   xmlns:task="http://www.springframework.org/schema/task"
10  xsi:schemaLocation="
11    http://www.springframework.org/schema/beans

```

```
12 http://www.springframework.org/schema/beans/spring-beans-4.0.x
sd
13 http://www.springframework.org/schema/context
14 http://www.springframework.org/schema/context/spring-context-
4.0.xsd
15 http://www.springframework.org/schema/mvc
16 http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd
17 http://www.springframework.org/schema/aop
18 http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
19 http://www.springframework.org/schema/tx
20 http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
21 http://www.springframework.org/schema/task
22 http://www.springframework.org/schema/task/spring-task-4.0.xs
d">
23
24 <!-- SpringMVC的配置文件,包含网站的跳转逻辑的控制以及配置 -->
25 <!-- 注解扫描 -->
26 <context:component-scan base-package="com.yuntu"
27 use-default-filters="false">
28 <!-- 不用全部都扫描,只需要扫描控制器组件,所以要禁用掉默认的扫描规
则 通过注解扫描Controller组件,只扫描控制器 -->
29 <context:include-filter type="annotation"
30 expression="org.springframework.stereotype.Controller" />
31 </context:component-scan>
32
33 <!-- 配置视图解析器,方便控制器处理结果之后进行页面的返回 -->
34 <bean class="org.springframework.web.servlet.view.InternalReso
urceViewResolver">
35 <property name="prefix" value="/WEB-INF/views/"></property>
36 <property name="suffix" value=".jsp"></property>
37 </bean>
38
39 <!-- 进行辅助配置-->
40 <!-- 将SpringMVC不能处理的请求交给Tomcat处理 -->
41 <mvc:default-servlet-handler/>
42 <!-- 能支持springmvc更高级的注解功能(JSR303,快捷的AJAX,映射动态功
能) -->
```

```
43 <mvc:annotation-driven></mvc:annotation-driven>
44
45 </beans>
```

web.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
3   <display-name>yuntu-dpm-curd</display-name>
4   <welcome-file-list>
5     <welcome-file>index.jsp</welcome-file>
6   </welcome-file-list>
7
8   <!-- 1.启动spring容器，加载spring的配置文件 -->
9   <context-param>
10     <param-name>contextConfigLocation</param-name>
11     <param-value>classpath:applicationContext.xml</param-value>
12   </context-param>
13   <listener>
14     <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
15   </listener>
16
17   <!-- 2.配置SpringMVC 前端控制器，拦截所有请求进行处理 -->
18   <servlet>
19     <servlet-name>dispatcherServlet</servlet-name>
20     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
21     <!-- 指明SpringMVC的配置文件路径
22     也可以不指定，需要跟web.xml文件处于同级目录，创建名为xxx-servlet.xml文件
23     -->
24     <!-- <init-param>
25     <param-name></param-name>
26     <param-value></param-value>
```

```
27 </init-param> -->
28 <load-on-startup>1</load-on-startup>
29 </servlet>
30 <servlet-mapping>
31 <servlet-name>dispatcherServlet</servlet-name>
32 <url-pattern>/</url-pattern>
33 </servlet-mapping>
34
35 <!-- 3. 字符编码过滤器（必须放在所有的过滤器之前） -->
36 <filter>
37 <filter-name>characterEncodingFilter</filter-name>
38 <filter-class>org.springframework.web.filter.CharacterEncoding
  Filter</filter-class>
39 <!-- 设置编码格式 -->
40 <init-param>
41 <param-name>encoding</param-name>
42 <param-value>utf-8</param-value>
43 </init-param>
44 <!-- 设置强制请求是否使用 -->
45 <init-param>
46 <param-name>forceRequestEncoding</param-name>
47 <param-value>true</param-value>
48 </init-param>
49 <!-- 设置强制响应是否使用 -->
50 <init-param>
51 <param-name>forceResponseEncoding</param-name>
52 <param-value>true</param-value>
53 </init-param>
54 </filter>
55 <filter-mapping>
56 <filter-name>characterEncodingFilter</filter-name>
57 <url-pattern>/*</url-pattern>
58 </filter-mapping>
59
60 <!-- 4. 使用REST风格的URI（将页面普通的post请求转为指定的delete或者
  put请求） -->
```

```

61 <filter>
62 <filter-name>HiddenHttpMethodFilter</filter-name>
63 <filter-class>org.springframework.web.filter.HiddenHttpMethodF
  ilter</filter-class>
64 </filter>
65 <filter-mapping>
66 <filter-name>HiddenHttpMethodFilter</filter-name>
67 <url-pattern>/*</url-pattern>
68 </filter-mapping>
69
70 <!-- 添加PUT请求的过滤器 -->
71 <filter>
72 <filter-name>HttpPutFormContentFilter</filter-name>
73 <filter-class>org.springframework.web.filter.HttpPutFormConten
  tFilter</filter-class>
74 </filter>
75 <filter-mapping>
76 <filter-name>HttpPutFormContentFilter</filter-name>
77 <url-pattern>/*</url-pattern>
78 </filter-mapping>
79
80 </web-app>

```

index.jsp

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
4 <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
5 <%
6   String path=request.getContextPath();
7   String basePath=request.getScheme()+"/"+request.getServerName()
  +":"+
8   request.getServerPort()+path+"/";
9   //将path放到页作用域，方便使用el表达式调用
10  pageContext.setAttribute("app_path", path);
11
12 %>

```

```
13
14 <!DOCTYPE html>
15 <html>
16 <head>
17 <meta charset="UTF-8">
18 <title>YUNTU_DPM_CURD</title>
19 <!-- 导入前端框架 (jQuery的引入在bootstrap之前) -->
20 <link rel="stylesheet" type="text/css" href="{app_path}/static/bootstrap-3.3.7-dist/css/bootstrap.min.css">
21 <script type="text/javascript" src="{app_path}/static/js/jquery-1.12.3.js"></script>
22 <script type="text/javascript" src="{app_path}/static/bootstrap-3.3.7-dist/js/bootstrap.min.js"></script>
23 </head>
24 <body>
25
26 <!-- 用于修改的模态框 -->
27 <div class="modal fade" id="projectInfoUpdateModal" tabindex="-1"
    role="dialog" aria-labelledby="myModalLabel">
28   <div class="modal-dialog" role="document">
29     <div class="modal-content">
30       <div class="modal-header">
31         <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">×</span></button>
32         <h4 class="modal-title" id="myModalLabel">项目编辑</h4>
33       </div>
34       <div class="modal-body">
35
36         <!-- 模态框中的主体, 表单构成 -->
37         <form class="form-horizontal">
38           <!-- 项目名称 -->
39           <div class="form-group">
40             <label for="ProjectName_add_input" class="col-sm-2 control-label">ProjectName</label>
41             <div class="col-sm-10">
42               <p class="form-control-static" id="projectName_update_static">
43               </p>
44             </div>
45           <!-- 起始时间 -->
```



```
46 <div class="form-group">
47 <label for="startDate_update_input" class="col-sm-2 control-
label">StartDate</label>
48 <div class="col-sm-10">
49 <input type="date" name="piStartdate" class="form-control" id="sta
rtDate_update_input" placeholder="StartDate">
50 </div>
51 </div>
52 <!-- 结束时间 -->
53 <div class="form-group">
54 <label for="endDate_update_input" class="col-sm-2 control-label">E
ndDate</label>
55 <div class="col-sm-10">
56 <input type="date" name="piEnddate" class="form-control" id="endDa
te_update_input" placeholder="EndDate">
57 </div>
58 </div>
59 <!-- 申报状态 -->
60 <div class="form-group">
61 <label for="status_update_input" class="col-sm-2 control-label">St
atus</label>
62 <div class="col-sm-6">
63 <!-- 状态的显示和提交保存的值是不一样的 -->
64 <select class="form-control" name="piStatus">
65 <option value="0">已申报</option>
66 <option value="1">申报中</option>
67 <option value="2">已审核</option>
68 </select>
69 </div>
70 </div>
71 <!-- 申报人列表 -->
72 <div class="form-group">
73 <label for="acid_update_input" class="col-sm-2 control-label">Appi
licant</label>
74 <div class="col-sm-6">
75 <!-- 内容时查询出来名字，提交的值是申报人的acid编号 -->
76 <select class="form-control" name="acid">
77
78 </select>
```

```

79 </div>
80 </div>
81 </form>
82 </div>
83 <div class="modal-footer">
84 <button type="button" class="btn btn-default" data-
dismiss="modal">关闭</button>
85 <button type="button" class="btn btn-primary" id="projectInfo_upd:
te_btn">保存</button>
86 </div>
87 </div>
88 </div>
89 </div>
90
91
92
93
94
95 <!-- 用于添加的模态框 -->
96 <div class="modal fade" id="projectInfoAddModal" tabindex="-1" rol
e="dialog" aria-labelledby="myModallLabel">
97 <div class="modal-dialog" role="document">
98 <div class="modal-content">
99 <div class="modal-header">
100 <button type="button" class="close" data-dismiss="modal" aria-lab
el="Close"><span aria-hidden="true">×</span></button>
101 <h4 class="modal-title" id="myModallLabel">项目申报</h4>
102 </div>
103 <div class="modal-body">
104
105 <!-- 模态框中的主体，表单构成 -->
106 <form class="form-horizontal">
107 <!-- 项目名称 -->
108 <div class="form-group">
109 <label for="ProjectName_add_input" class="col-sm-2 control-
label">ProjectName</label>
110 <div class="col-sm-10">
111 <input type="text" name="piProjectname" class="form-control"
id="projectName_add_input" placeholder="ProjectName">
112 <!-- 显示错误提示信息 -->

```

```
113 <span class="help-block"></span>
114 </div>
115 </div>
116 <!-- 起始时间 -->
117 <div class="form-group">
118 <label for="startDate_add_input" class="col-sm-2 control-label">S
tartDate</label>
119 <div class="col-sm-10">
120 <input type="date" name="piStartdate" class="form-control" id="st
artDate_add_input" placeholder="StartDate">
121 </div>
122 </div>
123 <!-- 结束时间 -->
124 <div class="form-group">
125 <label for="endDate_add_input" class="col-sm-2 control-label">Enc
Date</label>
126 <div class="col-sm-10">
127 <input type="date" name="piEnddate" class="form-control" id="endE
ate_add_input" placeholder="EndDate">
128 </div>
129 </div>
130 <!-- 申报状态 -->
131 <div class="form-group">
132 <label for="status_add_input" class="col-sm-2 control-label">Stat
us</label>
133 <div class="col-sm-6">
134 <!-- 状态的显示和提交保存的值是不一样的 -->
135 <select class="form-control" name="piStatus">
136 <option value="0">已申报</option>
137 <option value="1">申报中</option>
138 <option value="2">已审核</option>
139 </select>
140 </div>
141 </div>
142 <!-- 申报人列表 -->
143 <div class="form-group">
144 <label for="applicant_add_input" class="col-sm-2 control-label">A
pplicant</label>
145 <div class="col-sm-6">
```

```
146 <!-- 内容时查询出来名字，提交的值是申报人的acid编号 -->
147 <select class="form-control" name="acid">
148
149 </select>
150 </div>
151 </div>
152 </form>
153 </div>
154 <div class="modal-footer">
155 <button type="button" class="btn btn-default" data-
dismiss="modal">关闭</button>
156 <button type="button" class="btn btn-primary" id="projectInfo_sav
e_btn">保存</button>
157 </div>
158 </div>
159 </div>
160 </div>
161
162
163 <!-- 主页内容 -->
164 <div class="container-fluid">
165 <!-- 标题行 -->
166 <div class="row">
167 <div class="col-md-12">
168 <h1>YUNTU_DPM_CURD</h1>
169 </div>
170 </div>
171 <!-- 按钮行 -->
172 <div class="row">
173 <!-- col-md-2: 占两格
174 col-md-offset-10: 向右偏移是个格子
175 -->
176 <div class="col-md-2 col-md-offset-10" style="margin-bottom:
10px;">
177 <button type="button" class="btn btn-success" id="projectInfo_adc
_btn">新增</button>
178 <button type="button" class="btn btn-danger" id="projectInfo_dele
te_all_btn">删除</button>
179 </div>
```

```
180 </div>
181 <!-- 显示表格数据 -->
182 <div class="row">
183 <div class="col-md-12">
184 <!-- 表格展示出数据 -->
185 <table class="table table-striped table-hover" id="projectInfos_t
able">
186 <!-- 列标题 -->
187 <thead>
188 <tr>
189 <td>
190 <input type="checkbox" id="check_all">
191 </td>
192 <td>项目编号</td>
193 <td>项目名称（#）</td>
194 <td>开始日期</td>
195 <td>结束日期</td>
196 <td>申报状态</td>
197 <td>申报人</td>
198 <td>性别</td>
199 <td>工作年限</td>
200 <td>操作</td>
201 </tr>
202 </thead>
203 <tbody>
204
205 </tbody>
206 </table>
207 </div>
208 </div>
209 <!-- 显示分页信息 -->
210 <div class="row">
211 <!-- 分页文字信息 -->
212 <div class="col-md-4 col-md-offset-2" id="page_info_area"></div>
213 <!-- 分页导航信息 -->
214 <div class="col-md-6" id="page_nav_area"></div>
215 </div>
216 </div>
217
```

```
218
219 </body>
220 <script type="text/javascript">
221 //全局变量
222 var totalCount,currentPage;
223 /*1.页面加载完成后直接发送ajax请求，拿到分页数据*/
224 $(function() {
225 //页面进入直接访问第一页数据
226 top_page(1);
227 })
228
229 function top_page(pn) {
230 $.ajax({
231 url:"${app_path}/projectInfos",
232 data:"pn="+pn,
233 type:"get",
234 success:function(result){
235 //console.log(result);
236 //1.解析JSON数据并显示员工信息
237 build_projectInfos_table(result);
238 //2.解析并显示分页文字分页信息
239 build_page_info(result);
240 //3.解析并显示分页文字导航条
241 build_page_nav(result);
242 }
243 });
244 }
245
246 function build_projectInfos_table(result) {
247 //0.构建表格之前，将表格中数据全部清空，防止出现多个表格数据
248 $("#projectInfos_table tbody").empty();
249
250 //1.解析json数据，拿到其中的extend值中的PageInfo中的list
251 var projectInfos=result.extend.pageInfo.list;
252 //遍历
253 $.each(projectInfos,function(index,item){
254 //console.log(item.piProjectname,index)
255 //复选框
```

```

256 var checkBoxTd=$(("<td><input type='checkbox' class='check_item' /
    ></td>"));
257 //获取item中的每一个属性信息
258 var piIDTD=$(("<td></td>")).append(item.piId);
259 var piProjectnameTd=$(("<td></td>")).append(item.piProjectname);
260 var piStartdateTd=$(("<td></td>")).append(dateFarmat("yyyy-MM-dd",r
    ew Date(item.piStartdate)));
261 var piEnddateTd=$(("<td></td>")).append(dateFarmat("yyyy-MM-dd",new
    Date(item.piEnddate)));
262 var piStatusTd=$(("<td></td>")).append(item.piStatus);
263 var acNameTd=$(("<td></td>")).append(item.applicant.acName);
264 var acSexTd=$(("<td></td>")).append(item.applicant.acSex==
    1?'男':'女');
265 var workinglifeTd=$(("<td></td>")).append(item.applicant.workinglif
    e);
266
267 //编辑按钮
268 var editBtn=$(("<button></button>")).addClass("btn btn-primary btn-
    sm edit_btn");
269 append($("<span></span>")).addClass("glyphicon glyphicon-penci
    l").append("编辑");
270 //为编辑按钮添加自定义属性，表示当前操作项的id
271 editBtn.attr("edit-id",item.piId);
272
273 //删除按钮
274 var delBtn=$(("<button></button>")).addClass("btn btn-danger btn-sm
    delete_btn");
275 append($("<span></span>")).addClass("glyphicon glyphicon-trash").
    append("删除");
276 //为编辑按钮添加自定义属性，表示当前操作项的id
277 delBtn.attr("del-id",item.piId);
278
279 //将两个按钮拼接在一起
280 var btnTd=$(("<td></td>")).append(editBtn).append(" ").append(delBt
    n);
281
282 //组合
283 $("<tr></tr>").append(checkBoxTd).append(piIDTD).append(piProject
    nameTd).append(piStartdateTd).

```

```

284     append(piEnddateTd).append(piStatusTd).append(acNameTd).append(ac
SexTd).
285     append(workinglifeTd).append(btnTd).appendTo("#projectInfos_table
tbody");
286     });
287 }
288
289 //日期格式化操作
290 //dateFarmat("yyyy-MM-dd hh:mm:ss",new Date(item.piStartdate))
291 function dateFarmat(fmt,date)
292 {
293     var o = {
294         "M+" : date.getMonth()+1, //月份
295         "d+" : date.getDate(), //日
296         "h+" : date.getHours(), //小时
297         "m+" : date.getMinutes(), //分
298         "s+" : date.getSeconds(), //秒
299         //季度
300         "q+" : Math.floor((date.getMonth()+3)/3),
301         "S" : date.getMilliseconds() //毫秒
302     };
303     if(/(y+)/.test(fmt))
304         fmt=fmt.replace(RegExp.$1, (date.getFullYear()+"").substr(4 - Reg
Exp.$1.length));
305     for(var k in o)
306         if(new RegExp("(" + k + ")").test(fmt))
307             fmt = fmt.replace(RegExp.$1, (RegExp.$1.length==1) ? (o[k]) :
(("00"+ o[k]).substr((""+ o[k]).length)));
308     return fmt;
309 }
310
311 //2.
312 function build_page_info(result) {
313     //构建分页信息之前进行分页元素的清空
314     $("#page_info_area").empty();
315     //构建分页信息数据
316     $("#page_info_area").append(
317         "当前第"+result.extend.pageInfo.pageNum+
318         "页， 总共"+result.extend.pageInfo.pages+

```



```
319 "页，总共"+result.extend.pageInfo.total+"条记录"
320 );
321 //全局变量totalCount和currentPage
322 totalCount=result.extend.pageInfo.pages;
323 currentPage=result.extend.pageInfo.pageNum;
324 }
325
326 //3.
327 function build_page_nav(result) {
328     //构建分页导航之前进行分页导航元素的清空
329     $("#page_nav_area").empty();
330
331     //构建ul元素
332     var ul=$("<ul></ul>").addClass("pagination");
333
334     //首页元素
335     var firstPageLi=$("<li></li>").append("<a></a>").append("首
    页").attr("href","#");
336     //上一页元素
337     var prePageLi=$("<li></li>").append("<a></a>").append("<«>").attr
    ("href","#");
338     //判断是否还有上一页数据
339     if (result.extend.pageInfo.hasPreviousPage == false) {//没有上一页
340         //禁用上一页和首页按钮，给首页元素添加类样式
341         firstPageLi.addClass("disabled");
342         prePageLi.addClass("disabled");
343     } else {
344         //为元素添加点击翻页事件
345         firstPageLi.click(function() {
346             top_page(1);
347         });
348         //为元素添加点击翻页事件
349         prePageLi.click(function() {
350             top_page(result.extend.pageInfo.pageNum-1);
351         });
352     }
353
354     //下一页元素
```

```
355 var nextPageLi=$("<li></li>").append($("<a></a>").append("»").attr("href","#"));
356 //末页元素
357 var lastPageLi=$("<li></li>").append($("<a></a>").append("末页").attr("href","#"));
358 //判断是否还有下一页数据
359 if (result.extend.pageInfo.hasNextPage == false) {//没有上一页
360 //禁用下一页和末页按钮，给首页元素添加类样式
361 nextPageLi.addClass("disabled");
362 lastPageLi.addClass("disabled");
363 } else {
364 //为下一页添加点击翻页事件
365 nextPageLi.click(function() {
366 top_page(result.extend.pageInfo.pageNum+1);
367 });
368 //为末页点击翻页事件
369 lastPageLi.click(function() {
370 top_page(result.extend.pageInfo.pages);
371 });
372 }
373
374 //添加首页和上一页
375 ul.append(firstPageLi).append(prePageLi);
376 //遍历中间的数字
377 $.each(result.extend.pageInfo.navigatepageNums,function(index, item) {
378 //构建li的元素的对象
379 var numLi=$("<li></li>").append($("<a></a>").append(item).attr("href","#"));
380 //判断数字是否是当前显示的页码，添加bootstrap选中样式处理
381 if (result.extend.pageInfo.pageNum == item) {
382 numLi.addClass("active");
383 }
384 //每一个li的点击事件都是通过top_page函数进行ajax访问
385 numLi.click(function() {
386 top_page(item);
387 });
388
389 //循环将每个li添加ul中
```

```
390     ul.append(numLi);
391   });
392   //添加下一页和末页
393   ul.append(nextPageLi).append(lastPageLi);
394   //将ul加到nav导航条中
395   var navEle=$("<nav></nav>").append(ul);
396   //显示
397   navEle.appendTo("#page_nav_area");
398
399 }
400
401
402 //添加按钮的单击事件，弹出模态框
403 $("#projectInfo_add_btn").click(function() {
404   //进行表单内容的重置（包括表单数据，表单样式）
405   clear_add_from("#projectInfoAddModal form");
406   //发送ajax请求，查询申报人信息显示在项目申报得下拉列表
407   getApplicants("#projectInfoAddModal select[name=acid]");
408
409   //模态框弹出
410   $('#projectInfoAddModal').modal({
411     backdrop: 'static'
412   });
413
414
415 });
416
417 //表单清空
418 function clear_add_from(ele) {
419   //重置
420   $(ele)[0].reset();
421   //清空样式
422   $(ele).find("*").removeClass("has-error has-success");
423   //清空表单附加提示信息
424   $(ele).find(".help-block").text("");
425 }
426
427 //获取申报人列表信息
428 function getApplicants(ele) {
```

```
429 //先清除下拉列表中的原本的元素对象
430 $(ele).empty();
431
432 //发送ajax请求获取所有申报人
433 $.ajax({
434     url:"${app_path}/applicants",
435     type:"GET",
436     success:function(result){
437         //console.log(result);
438         //显示申报人信息在下拉列表中
439         $.each(result.extend.applicants,function(){
440             //console.log(result.extend.applicants);
441             //构建列表项内容
442             var optionEle=$("<option></option>").append(this.acName).attr("value",this.acId);
443             //将构建的option列表添加在select里面
444             optionEle.appendTo(ele);
445         });
446     }
447
448 });
449 }
450
451 //显示验证之后的提示信息
452 function show_validate_msg(ele, status, msg) { //选择器，状态信息，提示信息
453     //每次显示验证信息时，都将之前类样式清除
454     $(ele).parent().removeClass("has-success has-error");
455     //判断状态，确定状态对应的样式
456     if ("success"==status) {
457         //正确，添加正确样式
458         $(ele).parent().addClass("has-success");
459     } else if ("error"==status) {
460         //失败，添加失败样式
461         $(ele).parent().addClass("has-error");
462     }
463     //展示正确或者错误的文字信息
464     $(ele).next("span").text(msg);
465 }
```

```
466
467
468 //数据校验（添加表单）
469 function validate_add_form() {
470     //主要校验项目名称格式
471     //1.拿到校验数据，使用正则表达式
472     var projectName=$("#projectName_add_input").val();
473     var regName=/^[a-zA-Z0-9_-]{6,64}$|^[u2E80-u9FFF]{2,32}$/;
474     //验证
475     if (!regName.test(projectName)) {
476         //调用验证信息的显示函数
477         show_validate_msg("#projectName_add_input","error","论文名为2~32为
        中文，或者6~64为大小写字母、数字、下划线、横杠的组合");
478         return false;
479     } else {
480         //调用验证信息的显示函数
481         show_validate_msg("#projectName_add_input","success","");
482     }
483     return true;
484 }
485
486 //点击模态框中的保存，动态添加项目申报数据
487 $("#projectInfo_save_btn").click(function() {
488     //0.先要对提交给服务器的数据校验
489     if (!validate_add_form()) { //如果校验不成功，点击保存的事件就不再
        执行
490         return false;
491     }
492     //1.通过ajax请求用户名校验是否可用；如果没有才能继续保存
493     if ($(this).attr("ajax_validata_projectName")==="error") {
494         return false;
495     }
496
497     //将模态框中的数据交给应用服务器，进行数据库服务器保存
498     //发送ajax请求，保存员工
499     /*
500     通过jQuery提供的表单序列化进行表单数据提交（这用就不用通过id选择一个一
        个的获取一个一个值
501     而是用serialize()得到信息）
```

```
502  */
503  $.ajax({
504    url:"${app_path}/projectInfo",
505    type:"POST",
506    data:$("#projectInfoAddModal form").serialize(),
507    success:function(result){
508      //console.log(result);
509      if (result.code == 200) {//成功
510        //项目保存
511        //1.关闭模态框
512        $('#projectInfoAddModal').modal('toggle');
513        //2.来到最后一页显示刚才数据
514        top_page(totalCount);
515      } else {//失败
516        //判断时哪一个字段失败（假如失败中的值为undefined，证明代码有错误信息）
517        if (result.extend.errorFields.piProjectname!=undefined) {
518          show_validate_msg("#projectName_add_input","success",result.extend.errorFields.piProjectname);
519        }
520      }
521    }
522  });
523
524 });
525
526 //进行项目名重复验证，ajax异步请求
527 $("#projectName_add_input").change(function() {
528   //获取输入框的值
529   var projectName=this.value;
530   //如果校验的方法返回不成功，用户名重复验证就不再执行
531   if (!validate_add_form()) {//如果校验校验不成功，点击保存的事件就不再执行
532     return false;
533   }
534   //发送ajax请求校验用户是否可用
535   $.ajax({
536     url:"${app_path}/checkProjectInfo",
537     data:"projectName="+projectName,
```

```
538     type:"POST",
539     success:function(result){
540         //判断返回的Msg对象中的状态码
541         if (result.code == 200) {//成功
542             show_validate_msg("#projectName_add_input","success","√ 可用");
543             //通过jq对按钮进行携带信息（保证验证项目名重复时点击保存不能提交模态框）
544             $("#projectInfo_save_btn").attr("ajax_validata_projectName","success");
545         }else{
546             show_validate_msg("#projectName_add_input","error",result.extend.validate_msg);
547             $("#projectInfo_save_btn").attr("ajax_validata_projectName","error");
548         }
549     }
550 });
551 });
552
553
554 //编辑按钮的单击事件
555 /*
556  * 加载dom元素时，此时还没有编辑按钮直接绑定click函数无法绑定、
557  解决方案：
558  动态绑定：live()函数（已过去） on()函数（）
559  */
560 //on()函数绑定，绑定整个文档中
561 $(document).on("click",".edit_btn",function(){
562     //1.查询项目信息
563     getProjectInfo($(this).attr("edit-id"));
564     //2. 查询申报人的信息
565     getApplicants("#projectInfoUpdateModal select[name=acid]");
566     //3.将需要更新的员工id从编辑按钮上携带给模态框，用来修改时候使用；然后再弹出模态框
567     $("#projectInfo_update_btn").attr("edit-id", $(this).attr("edit-id"));
568
569
570     //然后模态框弹出
```

```
571 $('#projectInfoUpdateModal').modal({
572   backdrop: 'static'
573 });
574
575 });
576 //1.查询单个项目信息
577 function getProjectInfo(id) {
578   $.ajax({
579     url: "${app_path}/projectInfo/"+id,
580     type: "GET",
581     success: function(result){
582       console.log(result);
583       //获取员工对象
584       var projectInfo=result.extend.projectInfo;
585       //将员工对象放在模态框中对应的位置
586       $("#projectName_update_static").text(projectInfo.piProjectname);
587       $("#startDate_update_input").val(dateFarmat("yyyy-MM-dd",new Date(
       projectInfo.piStartdate)));
588       $("#endDate_update_input").val(dateFarmat("yyyy-MM-dd",new Date(p
       rojectInfo.piEnddate)));
589       $("#projectInfoUpdateModal select[name=piStatus]").val([parseInt
       (projectInfo.piStatus)]);
590       $("#projectInfoUpdateModal select[name=acid]").val([parseInt(proj
       ectInfo.acid)]);
591     }
592   });
593 }
594 }
595
596 //修改模态框中保存按钮的点击事件
597 $("#projectInfo_update_btn").click(function() {
598   //1.验证数据合法性，数据校验（根据需求）
599   //2.发送ajax请求，进行员工的数据更新（PUT方式请求）
600   $.ajax({
601     url: "${app_path}/projectInfo/"+$(this).attr("edit-id"),
602     type: "PUT",
603     data: $("#projectInfoUpdateModal form").serialize(),
604     success: function(result){
605       //隐藏模态框
```



```
606 $("#projectInfoUpdateModal").modal('hide');
607 //跳转到当前页面
608 top_page(currentPage);
609 }
610 });
611
612 });
613
614
615 //全选，全不选
616 $("#check_all").click(function() {
617     //将当前操作复选框的checked的属性值，作为参数
618     //设置为所有子项的复选框的checked的属性值
619     $(".check_item").prop("checked",$(this).prop("checked"));
620 });
621
622 //每个选中，全选按钮也选中
623 $(document).on("click",".check_item",function(){
624     //获取选中状态（选中状态是否等于复选框的总数量）
625     var flag=$(".check_item:checked").length==$(".check_item").length;
626     $("#check_all").prop("checked", flag);
627 });
628
629 //单个删除按钮的动态绑定点击事件
630 $(document).on("click",".delete_btn",function(){
631     //1.获取删除项的名字
632     var projectInfoName=$(this).parents("tr").find("td:eq(2)").text();
633     //2.获取按钮上的del-id
634     var delId=$(this).attr("del-id");
635     //询问判断
636     if (confirm("您确定要删除 “”+ projectInfoName +”” 吗? ")) {
637         //发送ajax请求
638         $.ajax({
639             url:"${app_path}/projectInfo/"+delId,
640             type:"DELETE",
641             success:function(result){
642                 //console.log(result);
```

```

643 //跳转到当前页
644 top_page(currentPage);
645 }
646 });
647 }
648 });
649
650 //点击删除所选项
651 $("#projectInfo_delete_all_btn").click(function() {
652 //创建用于存放删除的名字
653 var projectInfoNames="";
654 //创建用于存放项目id的字符串
655 var del_ids_str="";
656 //循环遍历
657 $.each($(".check_item:checked"),function(){
658 //将遍历的项目名称拼接到projectInfoNames中，并且用","隔开
659 projectInfoNames+=$(this).parents("tr").find("td:eq(2)").text()+
660 ("，");
661 //将遍历的id拼接到del_ids_str中，用"-"隔开
662 del_ids_str+=$(this).parents("tr").find("td:eq(1)").text()+"-";
663 });
664 //去掉一个","
665 projectInfoNames += projectInfoNames.substring(0, projectInfoName
666 s.length-1);
667 //去掉一个"-"
668 del_ids_str += del_ids_str.substring(0, del_ids_str.length-1);
669 if (confirm("您确定要删除 “”+ projectInfoNames +”” 吗? ")) {
670 //发送ajax请求
671 $.ajax({
672 url:"${app_path}/projectInfo/"+del_ids_str,
673 type:"DELETE",
674 success:function(result){
675 console.log(result);
676 top_page(currentPage);
677 }
678 });
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

mbg.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE generatorConfiguration
3   PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.
4     0//EN"
5     "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
6 <generatorConfiguration>
7   <context id="DB2Tables" targetRuntime="MyBatis3">
8     <!-- 注释生成配置 -->
9     <commentGenerator>
10       <!-- 去掉所有自动生成的注释 -->
11       <property name="suppressAllComments" value="true"/>
12     </commentGenerator>
13
14     <!-- 配置数据库连接信息 -->
15     <jdbcConnection driverClass="com.mysql.jdbc.Driver"
16       connectionURL="jdbc:mysql://127.0.0.1/dpm"
17       userId="root"
18       password="123456">
19     </jdbcConnection>
20
21     <jjavaTypeResolver >
22       <property name="forceBigDecimals" value="false" />
23     </jjavaTypeResolver>
24
25     <!-- 指定JavaBean生成配置，位置
26     targetPackage: 目标包位置
27     targetProject: 所在项目位置
28     -->
29     <jjavaModelGenerator targetPackage="com.yuntu.curd.bean" target
30       Project=".\\src\\main\\java">
31       <property name="enableSubPackages" value="true" />
32       <property name="trimStrings" value="true" />
```

```
32 </javaModelGenerator>
33
34 <!-- 指定sql映射文件生成配置，位置
35 targetPackage: 目标包位置
36 targetProject: 所在项目位置
37 -->
38 <sqlMapGenerator targetPackage="mapper" targetProject=".\\src\\main\\resources">
39 <property name="enableSubPackages" value="true" />
40 </sqlMapGenerator>
41
42 <!-- 指定Dao接口生成的位置，mapper接口 -->
43 <javaClientGenerator type="XMLMAPPER" targetPackage="com.yuntu.curd.dao" targetProject=".\\src\\main\\java">
44 <property name="enableSubPackages" value="true" />
45 </javaClientGenerator>
46
47 <!-- 每张表生成策略
48 tableName: 表名
49 domainObjectName: 实体类名
50 -->
51 <table tableName="applicant" domainObjectName="Applicant"></table>
52 <table tableName="projectinfo" domainObjectName="Projectinfo">
53 <!-- <table schema="DB2ADMIN" tableName="ALLTYPES" domainObjectName="Customer" >
54 <property name="useActualColumnNames" value="true"/>
55 <generatedKey column="ID" sqlStatement="DB2" identity="true" />
56 <columnOverride column="DATE_FIELD" property="startDate" />
57 <ignoreColumn column="FRED" />
58 <columnOverride column="LONG_VARCHAR_FIELD" jdbcType="VARCHAR" />
59 </table> -->
60
61 </context>
```

Copyright © 曹帅华 All Rights Reserved