Jsp And Servlet

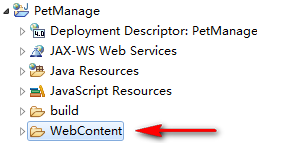
Jsp And Servlet 是java 的web 访问技术。

我们要开发web 应用，与之前建 立的项目所不同的是：此处我们建 立web项目

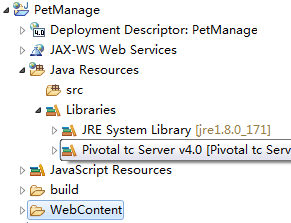
1. 建 立一个web 项目------Dynamic web project

Web.xml --- 网站配置文件,又称部署表述符,也就是说这个xml 文件中，配置了我们这个网站项目的一些特性.

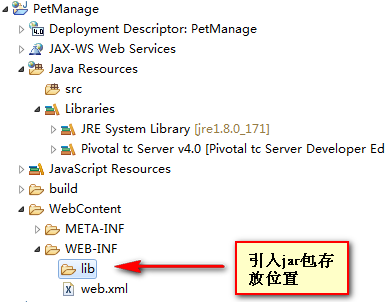
由于我们这个是web 项目，简单地说也就是网页项，肯定得有html,css,js, bootstrap,jquery ,这些资源在站点中放哪儿呢？就放到 webcontent 这个文件夹下面去

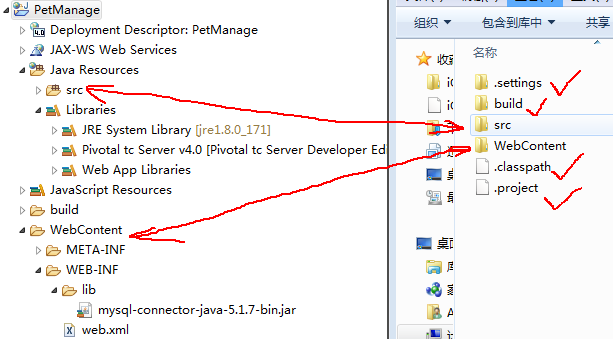


作为一个java项目，必然要有java 的包，java类，或者更专业点说----我们的entity,dao,biz,helper ,就放在 src 文件夹下边



另外， 我们要引入jar包， 比如, MySQL 的 jdbc 驱动包，日志的log4j.jar ,这些引入的包又放哪儿？

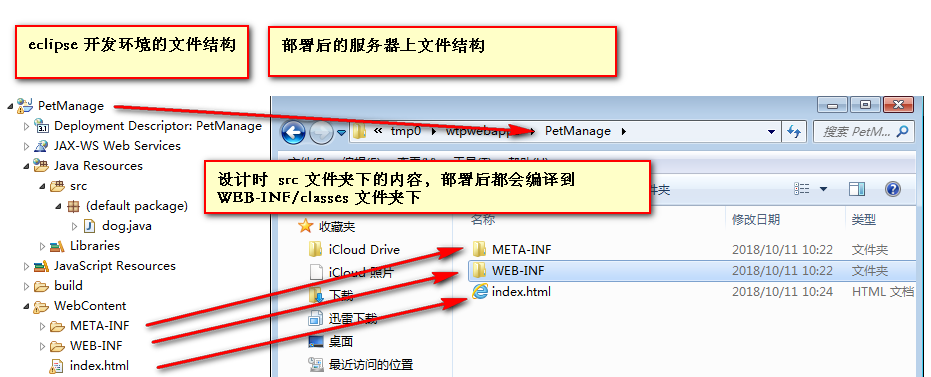




2 如何运行一个java web 网站 ?

我们运行一个 java 网站，需要一个web 服务器的支持，一般我们可以使用 tomcat web 服务器，我们又称之为容器.因此，如果我们机器中还没有tomcat,我们就要先行安装tomcat .安装好tomcat 后， 我们还要在开发环境配置web server， 我们刚安装的tomcat 服务器

1. 将我们的站点部署到tomcat服务器



.jsp ------ java server page java 服务器面页, java 动态页面，可以在页面上跟数据库进行增、删、改、查的交互

JSP 页面上出现的元素:

<%@ page %> ------ **指令元素**，此处是一个page 指令，该指令用途是:指明网页编码，网页内容类型

<% %> 脚 本元素 这个标签 对中可以直接写java代码

<%= %> ------ 表达式元素,可以在页面上直接显示一个表达式的结果

编码技巧:

我们让添加和修改共用一个页面，修改的时候，是要传递一个要修改的宠物的编号，要不然， 系统怎么知道 我们要修改哪一条记录.客户端向服务端传递数据，除了在表单中以表单数据的方式向服务端提交，另外， 还有在对应url 后面跟“查询字符串”的方式，比如:petadd\_edit.jsp?id=xxx, 查询字符串 除了可以在超链接中给出，还可以在地址栏给出，还可以在js 的location 中给出.

Delete 的写法：经过javascript 确认删除，则调删除功能，不过，这种写法是有隐患的， 我们立即演示这个**隐患**

**登录：**一个是要做一个明确的登录页面，能够输入用户名密码。 实施登录。另外，如果有人未经登录，直接从地址栏去请求相应页面， 又如何控制? 网站默认去index.html 页面，但是我们如何判断用户是否登录过呢？如果我们能够顺利判断用户是否登录过，那么若登录过， 我们可以放行让他访问他要访问的内容，否则，就导航去登录页面，让他登录

进入网站 ，默认会去index.jsp 页面， 我们可以在此页面写一个小脚本，在小脚 本中判断用户是否登录过

WEB-INF 是客户端禁止访问的，存储在WEB-INF 下的页面，只能通过服务端跳转进入.

Session 的存活周期是浏览器一关闭，session 立既消毁.

Cookie --- 与其它的内置对象不同，cookie 是需要自行实例化的内置对象，它里面只能存字符串，（session可以存对象,session存在服务端内存，cookie存在于客户端内存），l

1 图片上传分为两步骤：

1. 准备我们的上传表单
2. 服务端对上传数据的处理

实施步骤：

1. 在站点根下建 立upload 文件，作为上传内容存储的位置
2. 建 立上传表单页面

分页基于这样一种事实：数据表中有大量记录，比如10000条，而我们不可能一次性取出这10000条在页面展示，一方面用户体验不好，二方面大量数据在线路上传输，很占带宽，所以我们需要对要浏览展示的数据做分页处理.比如，每页只显示十条.

涉及到分页， 我们有几个关键词：每页显示条数，当前页数，总记录数,总页数

页面上要能正确显示当前是第几页，以及翻页按钮

要用下拉列表做翻页导航，考虑以下几点：

1. 动态绑定出下拉列表option 项
2. 以js 或 jQuery 写出 option 响 应翻页的点击动作，可以写在onchange 或 click

由于各页面都会共用到一些css,js 的脚本，我们专门写一个头文件，这个文件中包含所有我们要用到的css,js ,jquery,让每个页面包含。

在实际开发中如何去解决这样一种路径差异：

两种办法：

第1种： 让所有页面都处于相同深度层次的文件夹下,这样的话，你所有页面跟静态资源（css,js）之间的相对位置都是一致的.

第二种： 在页面上用<base >标签的path 属性，指定站点的根，让页面上所有的路径定位，都 变成以根为参照的相对路径.

1. 根据传过来的要删除的id,从数据库取出这条宠物
2. 依据这条记录的photo 字段的值，去删掉磁盘上的图片文件
3. 删除数据表中这条记录

编码常用的一个技巧：当逻辑特别复杂，不好共用代码片段的时候 ， 我们往往要设法对代码做分支处理，将不同的代码逻辑分到不同的分支里边去，分流了我们编写的复杂性.

大部分EL 表达式中报：找不到对应属性的时候 ， 一般都是因为设计实体类属性的时候，没有采用驼峰命名法，使得EL 表达式解析器解析你的表达式时，无法按驼峰命名法反向组装出对应属性的get 访问器，所以这种时候，你去改你的属性名称就能解决问题。

页面上使用EL表达式，与使用JSP表达式（小脚本表达式），有一个小小不同，相应变量要求要放到某个可见范围------isp页面的四个作用范围:page,request,session,application.

所以换种说法，某个变量，你要在页面上用EL 表达式访问，必须把它放到某个范围.

放入某个范围 都是用 XXX.setAttribute(xxx) 从范围中人为取出:XXX.getAttribute(xxx)

EL 表达式中判断非空的语法

"${not empty sessionScope.userId}"

不要写成如下形式:

${ sessionScope.userId != null}

布尔变量值取反，不是用”!” ,而是用 NOT 关键字

**<c:if test=”${条件表达式}”>**

**</c:if>**

**<c:forEach items=”你要迭代的变量” var=”你在循环体当中可以引用的循环变量名” varStatus=”varStatus” begin=begin end=end step=step>**

**</c:forEach>**

EL 表达式除了可以使用core库当中的常规标签 ，还可以引入函数库，然后使用EL 的函数标签，简化编程。EL 的函数标签 ， 除了使用系统标签 ，还可以根据自已的需求，自定义标签，比如，我们的项目中，在petlist页面上， 我要将owner\_id 转化为 owner\_Name,我就可以自定义一个函数标签 去完成这个任务.

声明一个servlet 的步骤：

1 我们选 中servlet,添加一个servlet,虽然也可以直接建 一个普通java类，但是建 普通java类要麻烦一点，何以见得会麻烦一点，表现在以下两点：1）要去web.xml 当中声明你所建立的servlet, 2) 自动为我们继承该继承的父类

凡是修改了web.xml 的操作，都 得重新部署

Annotation ------ 就一种特殊的类，写在类或方法的上面，标识该 类或方法的特殊属性

包名是以点为分隔符

在系统中声明一个servlet 有一，两种方式，都 可以使用：1 annotation 方式 2 web.xml 方式（传统方式）

在servlet 当中，不用再指明ContextPath,Servlet 当中会自计算ContextPath,但是，在客户端页面中写超链，必须加 contextPath; JSTL 的链接的跳转路径， 也不要加 contextPath

要在我们的项目中使用自动分页小型框架，我们要做以下准备工作:

1. 引入jar包 ------ pager-taglib.jar
2. 拷贝页脚的分页按钮文件进入项目 ------ pager.jsp
3. 将分页类拷贝到项目 ------ pager.java

实施分页操作设计:

1. 保证数据访问层写得有取分页数据的方法------也就是按跳过条数和每页条数取记录的方法; 保证数据访问层有取得总条数的方法
2. 在准备数据的Servlet 中，从查询参数中取offset 值 ，pageSize 值 ;

利用以上两个参数:

取当前页数据记录

取总条数;

实例化一个pager.java 的分页类的对象,将当前页记录(List<Pet>),和总条数设置到这个对象当中,然后将该 对象放入request范围（一定以”pager”为键名） ,转发回jsp页面.

3 在页面上合适的地方，引入分页页脚文件 。

页面实现ajax 交互，虽然多如牛毛的各种javascript框架，都提供了ajax 功能的封装版本，但万变不离其中的是javascript的原生ajax 对象的封装：XMLHttpRequest ------ 但这个对象的跨浏览器兼容性不太好，没有框架中封装的ajax功能使用方便.

原生 XMLHTTPRequest 对象实现Ajax的步骤：

**1、创建XMLHttpRequest对象**

|  |
| --- |
| var xmlhttp;  function doAjax(url) {  try {  xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");  } catch (e) {  try {  xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  } catch (e) {  try {  xmhttp = new XMLHttpRequest();  if (xmlhttp.overrideMimeType) {  xmlhttp.overrideMimeType("text/xml");  }  } catch (e) { }  }  }  } |

2、 指定异步任务完成后，系统要执行的回调函数:

|  |
| --- |
| Xmlhttp.onreadstatechange=回调函数名称 (千万不要加上函数的小括号) |

3、设置请求信息:

|  |
| --- |
| Xmlhttp.Open(“传递方式---POST get”,”URL”,boolean--- true false); |

4、 发送请求

|  |
| --- |
| Xmlhttp.send(null); |

我们在实际编码应用中，会很多场景使用Ajax技术，能够巧妙构造我们的应用逻辑。不过。 我们更多使用 前端框架jQuery(Vue.js Angenar.js)里边，为我们集成封装好的Ajax 框架，有很好的跨浏览器兼容性，而且使用简便.

jQuery 的 Ajax 函数提供给我们三个使用: $.get $.post $.Ajax

/\* $.get(url,{},function(){回调}) \*/

传参的话，可以使用查询字符串，更推荐JSON 格式的参数传递 ---写在{}中，用JSON 对象格式表示 .

$.get 与 $.post 书写方式差不多，只是传递 方式不一样;

$.ajax 可以做更细粒度的控制.

我们传统的渲染模式，是服务端渲染---就是在服务端已经定死这些数据是显示为table 还是 ul 呢，已经在服务端写死。随着移动端设备的不断丰富，这种模式显得力不从心，于是，我们选 择客户端渲染---服务端只负责送回JSON数据，这些数据具体怎么渲染，怎么排列，收客户端自主解析.我们本周主要学习、练习内容，就是服务端、客户端以JSON 数据为主要交互.

JSON------ Javascript Object notation:js 对象表示法，

{}--- 表示对象

[]--- 表示数组

宠物店的列表上有id,我们现在进一步丰富我们的软件交互功能---将id做超链封装，点击id的时候，我们弹层显示该店下所有的宠物，要求：弹层当中也要分页.我们数据交互采用json数据，客户端渲染。

我们要取出某个宠物店的所有宠物，为此我们要设计一个servlet;随着项目功能的扩展和丰富， 我们会发现，我们的项目中会有多得难以忍受的servlet,因为servlet 里边，我们可以操作的只有两个方法：一个doGet,一个doPost,所以， 我们要完成一个功能，就得为这个功能专门编写一个servlet,我们有没有什么技术方案能够改善这种状况呢?后一步， 我们通过反射技术，扩展servlet 的功能就能解决这个问题。

如果我们采用客户端渲染，则就告别了EL,JSTL这些Java 的语法元素，完全用js,jQuery的语法来做数据绑定---也就是将服务端送来的Json数据显示到页面.

我们在客户端解析JSON 数据的时候 ，有时会对于JSON 当中封装的数据对象的数据结构不太了解，其实，我们可以先设法将这个JSON字符串打印出来， 然后比照着去编码，不要通过“猜”的方式。

BaseServlet 框架就是spring mvc 框架的雏形，是一个独立的，可胜任于大型企业级项目开发的框架，有运行性能的基本保障。而要使用这个框架，在任何一个项目中，BaseServlet.java 这个类的写法都是一样的，可以直接拷贝过去使用。另外，在BaseServlet 框 架中， 我们编写我们的功能servlet 的时候，有以下几点刚性要求，必须遵守:

1. 我们的功能servlet 不再继承于httpServlet,改而继承于BaseServlet.
2. **我们的功能servlet 当中的方法，都必须是公有的，并且返回一个字符串的方法,给它传递 一个request,一个response 这两个参数，而返回的这个字符串就代表方法执行完成后，要跳转的路径，如果是做重定向跳转，则路径值前面得加上:”RIDIRECT”前缀，以此告诉baseServlet 要做重定向跳转（也就是不做转发跳转）,否则就是做转发跳转.**
3. **在客户端调用一个功能servlet 的时候 ，与以前的调用相比，末尾都必须跟一个查询字符串，以此告诉baseServlet ，是要调用功能servlet 当中的哪一个方法. Servlet\_partern?method=XXX,**

**正常进度：ssm ------ spring, spring MVC, mybatis**

**Hibernate**

**maven**

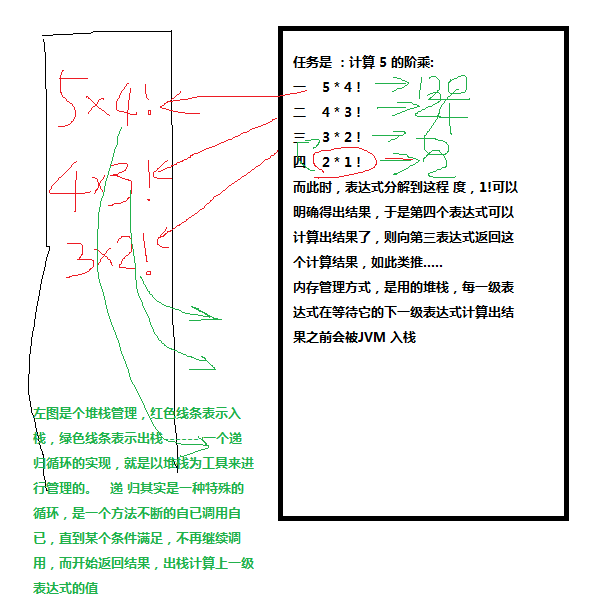
**VUE.js + Spring Boot**

**补充资 料：递归代码的编码**

**Oracle 数据库**

**多线程**

**递归的编码过程：对一个无法明确计算得到结果的表达式，进行分解，得到另一个表达式，这另一个表达式能得到明确结果吗？能：则返回明确结果；不能，则将该 另一个表达式又进行分解，直至得到的表达式可以得到明确果---就返回明确结果，不再进行分解。 而一旦有某一级的分解表达式返回了明确结果，则它的上一级表达式就可以计算出明确结果了，反推回来，一级表达式也就能计算出结果了.**

****

**只要是层次数据， 我们肯定是使用递归技术来处理**

**递归定义：我们在定义menu 实体类的时候 ，里面有个属性是子菜单的集合，List<menu>,而这个list 中装的又是menu,于是就形成，我们在定义menu 的定义体当中，又用到了menu这个概念，这就是递归定义。我们要用递 归方式扫描出所有层级（无限级）的菜单，首先就要有这个递 归定义的实体类.**

RBAC 讲解

1. 名词解释:什么是rbac?作为一个企业级项目，必须有严谨的安全访问机制，作为应用程 序后台，其功能不仅是要求要登录过才能访问，另外，不同身份的登录者，他们在系统中拥有的操作权限也是不一样，解决访问权限控制的技术，就是所谓rbac技术。
2. RBAC ------ Role Base Access Control:以角色为基础的访问控制，意译方式：以角色为基础的权限访问控制== >以角色为基础的资源访问控制, 而后者的控制粒度，比前者更为精细.

以角色为基础，是相对于以用户为基础，以往时代，我们曾经以用户为基础，也就是为每个用户赋权，导致系统每添加一个用户，就要实施一次复杂的赋权过程，效率低且过程繁复；改而为将相应权限赋值给角色（此处角色相当于一个权限组），新增加用户时，我们只要指明他是属于什么角色（也就是属于什么权限组）就可以了，使得赋要过程 大为轻松。

1. 但凡我们在自已的企业级项目中，要编写自已的RBAC控制，我们必须确保数据库中，有以下几个表:User 表,Role 表, resource 表,以及用来表达这些表之间的主外键关系的中间关联表，这些表不用限定用何表名，只要实施相应功能的表存在就行，叫什么名字可以依据项目的命名约定而定义。系统中要有完成这些功能的表，但是表名不一定要叫这些名字。表达这些表之间的主外键关系：如果数据量不是很大，直接用关联字段，反之，则用中间关联表. 为角色-资源添加一个角色资源管理表 role\_resource(roleid,resourceid) 用户 \_角色表 user\_role(userid,roleid)

RBAC 要完成的具体控制

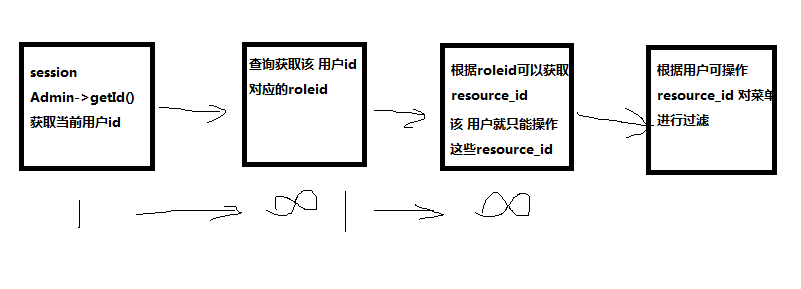
1. **菜单过滤**---过滤掉当前登录者所不具备权限的菜单
2. **请求检查**---对请求相应资源的request进行检查，当前登录者是否具备该资源的操作权限，比如操作者熟悉你的站点文件结构，他从浏览器地址栏直接对你站点的某项资源发起请求，而如果他对这项资源没有操作权限，这个请求也应该被拒绝.

**控**制一个servlet或servlet 中的某个方法要具备某种权限才可访问，我们一般采用在servlet 或 对应的 servlet的方法上添加Annotation 来加以控制.

**Annotation------是加于一个类或类的方法或方法的参数上面的一种特殊的类，**[**它标明了类或类的方法或方法的参数的某项特性.以@符号打头**](mailto:它标明了类或类的方法或方法的参数的某项特性.以@符号打头)**.**

**应用步骤：1 声明annotation**

1. **对于页面上出现的功能按钮**，屏蔽无权操作者的操作，比如某登录者对某资源只有查看的权限，并无增删改的权限，则应控制他不能点击增删改铵钮，或增删改铵钮根本不对他显示.



编码过程当中，一旦遇上某种场景的排重操作，需要以一个集合来记录排重的选择结果的，则最佳选择是用HashMap<Object,Object> 集合 == > HashMap<String,Integer>

1. 取出所有菜单后， 进行过滤
2. **边取的时候，就用它的权限同步过滤**

**我们在已完成的项目中添加模块功能的步骤:**

1. **设计表**
2. **实体类**
3. **Dao Biz 及 接口们**
4. **表现层逻辑**

**现在我们给大家示范增加一个新模块------部门模块**

**现在我E盘根目录下边有一个发布过的站 点的备份，我不想将之拷贝至tomcat 的标准发布路径下（/webapps）,就在原地(e:\PetManage)发布， 可以吗？可以，这种发布，我们只要配置tomcat 的虚拟站点:**

**如果我们测试中， 要测试我的一个申请好的域名，我们要设置本地hosts()文件，让127.0.0.1 解析到我们的域名(比如：**[**www.ys.com**](http://www.ys.com)**), 然后在tomcat 配置文件中增加该 本机的配置 .**

**前**端页面上有时会用到一些需要后端计算的功能，这个时候， 我们可以考虑使用JSTL 表达式的自定义函数功能，

1. 我们要实现的功能，先在后端的一个java 类中以一个静态方法的格式完成功能逻辑的编写，
2. 在functions.tld 文件中，以既定格式声明这个静态方法为一个可以在前端调用的JSTL的自定义函数，
3. 在前端要使用的页面上，引入第2步中声明的taglib 文件，就可以在页面上以既定格式调用这个静态方法了。

**以上三步，每一步在编写过程 中，都 有相应的语法格式要求，参照案例中的格式。**

**在TomCat 服务器上配置多主机头支持:**

1. **配置 c:\windows\system32\drivers\etc\hosts 文件，要求以管理员身份打开，否则无法保存修改结果**
2. **修改tomcat安装路径/conf/server.xml 配置文件,去增加一个<host>节点声明，一个host 节点就对应的一个主机头**
3. **手动启动tomcat , 不要在eclipse 中启动tomcat**

|  |
| --- |
| //更新部分字段的SQL编码技巧  admin(id,name,password,nickname,age,address,roleids)  name,nickname,roleids  String sql\_head="update admin ";  String condition=" where id=?";  if(admin.getName!=null){  sql\_head+=" set name=? , ";  }  if(admin.getPassword()!=null){  sql\_head+=" set password=? , ";  }  if(admin.getNickname()!=null){  sql\_head+=" set nickname=? , ";  }  if(admin.getAge()!=0){  sql\_head+=" set age=? , ";  }  if(admin.getAddress()!=null){  sql\_head+=" set address=? , ";  }  if(admin.getRoleids()!=null){  sql\_head+=" set nickname=? , ";  }  sql\_head.substring(0,sql\_head.lastIndexOf(","));  String sql=sql\_head+" "+condition; |

**MyBatis ------**

1. **是什么？是一个数据持久层的框架，使我们能摆脱手写SQL 代码，是一个ORM 框架**

ORM **------ Object relation Mapper:对象 关系 映射,在java 中，典型的ORM框架有：MyBatis, Hibernate ------ MyBatis 是半自动化的， 需要我们手写sql 代码，Hibernate 是全自动的，持久化所需sql代码，全自动生成.**

**项目中启用MyBatis 的步骤:**

1. 拷贝mybatis应用jar包到项目当中,并添加到构建路径.
2. 在系统构建路径（src文件夹）下建立MyBatis 的配置文件（MyBatis-config.xml）

其中 ：<properties>---指定数据库连接配置的属性文件

属性文件的名字自定义， 内容是数据库连接的特性值:

<typeAliases>: 其中指定我们的实体类所在的包，或实体类的非完全限定名

<environments> :配置mybatis 的运行环境，主要配置数据库连接池、事务管理方式等

|  |
| --- |
| <properties resource=*"connection.properties"*></properties>  <typeAliases>  <package name=*"com.hxzy.entity"*/>  </typeAliases>  <environments default=*"development"*>  <environment id=*"development"*>  <transactionManager type=*"JDBC"*></transactionManager>  <dataSource type=*"POOLED"*>  <property name=*"driver"* value=*"${driver}"*/>  <property name=*"url"* value=*"${url}"*/>  <property name=*"username"* value=*"${username}"*/>  <property name=*"password"* value=*"${password}"*/>  </dataSource>  </environment>  </environments> |

<mappers> ------ 在mybatis 配置文件中注册我们的实体映射文件， 我们的每一个实体类，要有一个实体映射文件与之对应，是一个xml文件,其中的内容是:对这个实体类进行CURD 的sql 语句

Mappers 有一个属性是：namespace: 其中就是书写我们的实体类的全限定名

1. 编写实体类，以及实体类的映射文件（CURD 的语句文件）,并在mybatis的配置文件中注册，让mybatis能找到这个映射文件。
2. 编写测试类测试mybatis的功能

在应用程 序中使用mybatis 的步骤:

1. 从一个所谓资源件(Resources)，通过传入mybatis 的配置文件为参数，获取一个InputStream 输入流文件;
2. 利用 InputStream 为参数，获取 session 工厂。

使用myBatis 就是两个重要的xml 文件，一个是mybatis 配置文件：mybatis-config.xml; 另一个是实体映文件----- 每一个实体类都要对应一个映射文件.

Mapper 节点的 namespace 属性， 你要正确的写为 对应实体类的全限定名 ；mapper 节点中就是书写对应实体类的CURD语句

我们以映射文件中的 命名空间+语句id 的方式来调用sql语句，并不符合我们java人员的使用习惯，我们更习惯 对象名称.对象方法(参数) 这样的格式，那么，mybatis 好提供了这种调用方式给我们，但是，用这种方式，要改动一个小地方：就是：映射文件中的命名空间，要指向接口的全限定名,然后，声明一个接口文件，接口中的方法，就一一对应映射文件中的语句.

接口中的方法名=语句的id

方法的参数=parameterType

方法的返回值 = resultType

Generater ------ 这是一个mybatis 的辅助工具， 帮助我们经过简单的配置步骤，自动生成项目中要用到的代码------包含数据库表的entity,Mapper,dao，将生成的代码， 拷入项目，就可直接使用.

步骤:

新建 eShop 项目了:

映射文件所放的包名:com.eshop.mapper

实体类：com.eshop.model

dao 接口:com.eshop.dao

Spring

从诞生时候起，两大核心功能：1 **DI**--- dependency injection :依赖注入

**2 AOP**--- aspect Orientation(面向) program : 面向方面编程

Spring 核 心工作原理是:动态代理的实现

Spring 当前java开发市场上，有几个并行的都 在使用的spring 的版本： 2.5 3.0 3.1 4.0

步骤：

1. 引入jar 包: 版本讲究. 2.5--- 整合包，spring.jar 3.0，3.1 --- 分包的，各个不同的模块功能写在不同的jar包中，要用什么功能， 就只引入对应的jar包
   1. ----- 支持annotation 声明式配置，支持自动包扫描技术

3.0，3.1 ------ 支持 Java 源文件来编写配置

4.0 ----- spring boot spring cloud spring data

2 编写一个功能类

3 编写配置文件，在配置文件中声明第二步的功能类

1. 写测试类

未写代码，先引jar包------军马未动，粮草先行,,我们要用 dbcp 数据源连接池， 先要把dbcp的jar 拷入项目.

流行的数据库连接池管理方式有以下中种:1 jndi 中的context.xml 中配置的连接池 2 dbcp 3 cp30 4 druid------阿里巴巴集团的科技公司的 昆少团队 研发.

我们向eshop 项目中加入spring 功能，表现层暂时还用servlet,让spring 和我们 数据层的 ORM------MyBatis 先整合起来,等明天， 我们讲解了Spring MVC ,再把个现层的servlet 换做Spring MVC 来实现，至此，三大框架的整合（ssm--- spring spring MVC MyBatis）就宣告成功.

将spring 功能加入eshop

1. 导入 jar 包
2. 编写applicationContrext.xml 文件的骨架
3. 由于我们的是动态web项目，我们要将 applicationContext.xml 文件在 系统的 web.xml 文件中注册,并在web.xml 文件中注册spring 的 侦听器.

我们首先完成在一个控制台应用程序中整合myBatis 与 Spring 框架

1. 导包---   
2. applicationContext.xml 文件编写
3. myBatis 与spring 整合后，我们的dao 可以有以下写法:
4. 就还是原先的 Mapper 接口映射的写法:dao 接口中的方法名称跟映射文件 中的语句的id 相同，并且参数与返回值都与其相同------这种写法有个称谓叫做:mapper 代理方式，这种代理写法在 整合 以后，applicationContext.xml 文件中可以有两种声明方式:

一) 就是通过在sqlsessionFactory 的声明中，指定 映射文件在项目中的扫描位置 --- <property name="mapperLocations" value="classpath\*:com/hxzy/mapper/\*.xml"></property>

二) 直接把dao 接口（mapper代理）,作为一个bean 在applicationContext.xml 中声明

用 MapperScannerConfigurer 指定接口所在的包位置 （便捷）

用 MapperFactoryBean 将每一个接口用为一个bean 在 applicationContext 中声明（麻烦一点）

以上是使用接口代理类的方式，还有一种方式：就是使用我们所熟知的 dao 的声明方式

整合开始:

1. 拷3.1 jar 包, common\_logging, log4j,common\_pool,common\_dbcp,common\_io, mybais\_spring 桥梁包
2. 编写 spring 配置文件------applicationContext.xml 文件

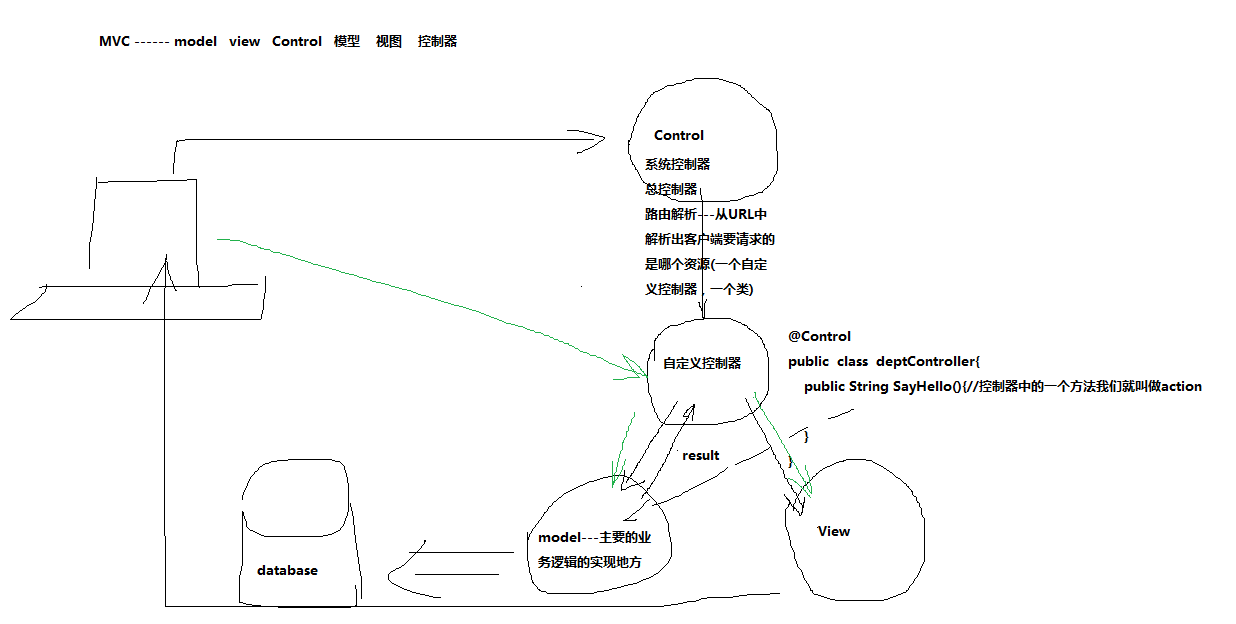
只要是在spring容器(容器的具体体现：applicationcontext)注册了的java类---JavaBean,都 能被容器所管理------它的生命周期(从实例化、初始化到销毁)的全过程，都由spring控制.

依赖注入(DI) ------ 分两种形式进行注入：1） set 注入---是set访问器注入 2） 构造注入---从构造器注入

其实还有一种是叫静态注入

一个类要交给spring 管理，要么写applicationContext 中写<bean>去声明；另一种就是写annotation。

Spring MVC ------



工作于数据层的框架------ORM------myBatis Hibernate Spring data JPA(Java Persistence API)

工作于表现层的框架------ MVC------ Spring MVC, struts1 Struts2

区分---MVC 架构和 三层架构概念的区别

在项目中启用 spring MVC 框架功能的步骤:

1. 导包
2. 在web.xml 文件中配置spring MVC 的总控制器----DispatcherServlet
3. 编写spring MVC 的配置文件-------文件名约定: DispatcherServlet的名称 加上 –servlet.xml -------- springapp-servlet.xml

学习任何语言的任何框架，把握两点则可：1） 路由解析方式 2）参数传递方式(参数从客户端到服务端，数据是如何从控制器到视图)

<http://localhost:6060/myframework/dept/fetail?id=5>

上下文/Controller/Action?queryString

路由重写: <http://localhost:6060/myframework/dept/fetail/id/5>

<http://localhost:6060/myframework/dept/5/fetail>

返回值: ”dispatcher:”------转发去默认视图

“redirect:”------重定向去默认视图

“ajax:”------什么地方也不跳转，因为该请求来自一个ajax异步调用

“same:”------调用同一控制器中另一方法

“other:”------调用另一控制器中的方法

“jsp\_ dispatcher:”------直接转发去一个指定的jsp

“jsp\_ redirect:”------直接重定向去一个指定的jsp

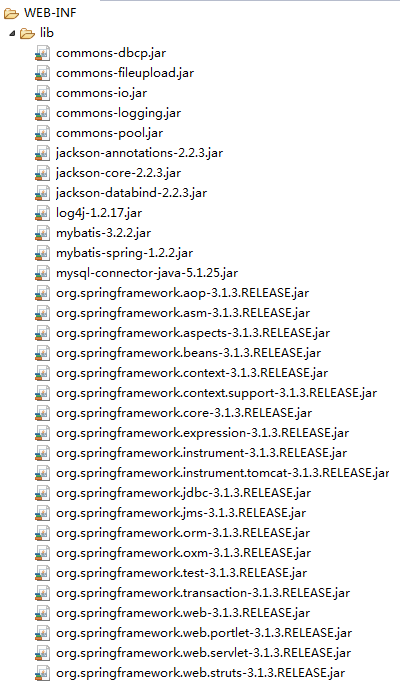
Springmvc-servlet.xml 中要配置的项:

1. 要配置静态资源的放置路径，这样静态资源就不会被总控制中心拦截
2. 要配置一个视图解析器，在其中指明视图定位的前缀、后缀
3. 涉及到文件上传，要配置 一个节点，声明有关上传文件控制项

Ssm 整合步骤

1. 导包------ 导入spring 3.1 的所有jar包，mybatis 包，mybatis 跟spring 之间的桥梁包，log4j.jar, common-logging.jar, common-fileupload.jar, common-dbcp.jar,common-io.jar,common-pool.jar

所需导入的jar 包 如下图:



1. 将 spring-mvc 的 dispatcherServlet (这是spring-mvc 的总控制中心)，在web.xml 里边注册，另外，要通过一个contextParam（上下文参数） 节点，指定 spring 的配置文件所在位置及名称 , 还要注册一个spring 的侦听器。

Web.xml 文件中的配置如下图:



1. 编写相关功能的配置文件:
   1. myBatis.xml------ 配置文件的名称，自定义，到开发文档API中拷贝相应的文档大纲



实体映射文件的案例图如下:



* 1. spring MVC 的配置文件------ 配置文件的名称，要依据如下命名约定：总控制器的那个servlet 所取的名字 加上 “-servlet.xml”,如下图:

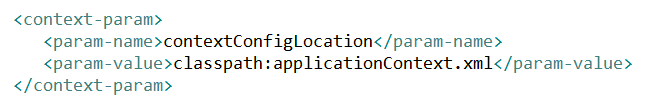
,

图中我们看到，servlet 的名称是:springmvc,于是配置文件名称 就是:springmvc-servlet.xml,而且该 配置文件要放在WEB-INF 下面

配置文件内容如下:

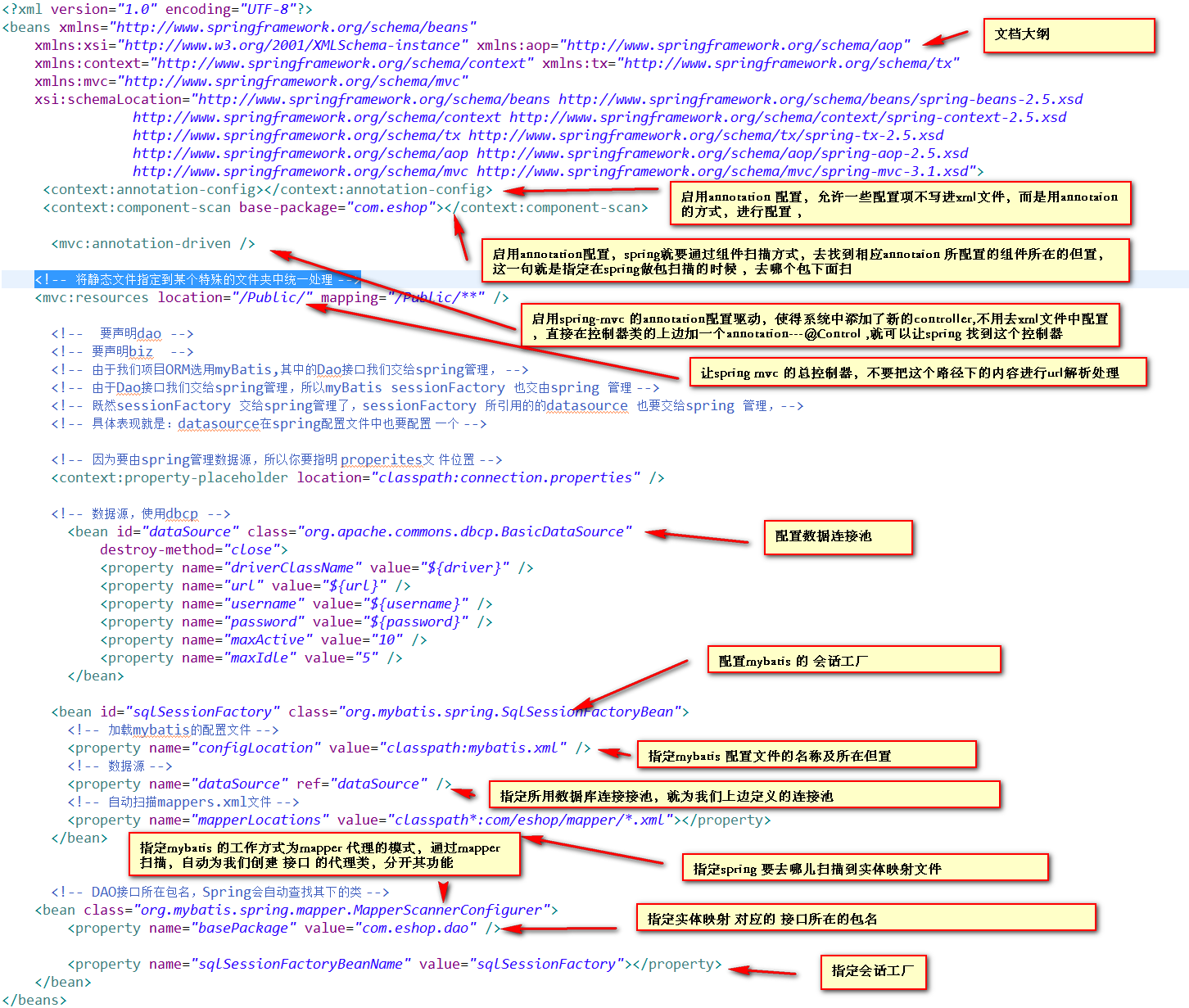


* 1. spring 的配置文件 ------ 配置文件的名称及存放位置，我们都 可以自定义，因为这个配置文件的名称及位置，我们是在web.xml文件中用上下文参数指定了的，web.xml 中的情形如下图:



图中的写法就表明， 我们的spring 配置文件，名叫applicationContext.xml,位置在系统构建路径下(classpath ，设计是就是src 目录，部署后，是 WEB-INF/classes 目录)-------这种命名及在放位置，是业界一个默认的约定。

applicationContext.xml 文件的配置内容如下图例 :



4 配置文件都编写完成，框架的整合搭建过程就完成了。可以进行日常常规的开发了------ 日常常规就是去编写我们的业务逻辑及功能实现的 Controller 及其中的方法，以及 视图（jsp）,Model (实体类，dao ,biz )