京淘总结

通用mapper为什么不用写sql

SELECT id,parent\_id,NAME,STATUS FROM tb\_item\_cat

拼串

通用Mapper基于Mybatis提供注解来开发

@SelectProvider 知道这是一个查询操作 SELECT FROM

表：根据泛型设置类，找到POJO，同POJO类上的注解@Table(name="tb\_item\_cat")表名：tb\_item\_cat

字段：找到POJO，找属性，如果没找到，去看mybatis有没有配置全局的驼峰规则，如果mapUnderscoreToCamelCase=false，就拿属性作为字段名称；如果mapUnderscoreToCamelCase=true，把当前的属性反驼峰规则（按数据库字段规则：sortOrder 先把大写字母前加下划线，然后把单词变小写sort\_order）；如果找到注解@Column，把注解中设置的值作为字段。

总结：通用mapper的方法上都代用mybatis提供的相关注解，就知道是什么语句了，然后根据泛型找到相对应得表找属性，在看看mybatis中的驼峰映射规则，和column等进行拼写sql语句呀，pojo中有用jpa来实现

1. 人员配置 项目组50人

项目经理2人：任务分配，任务监督，项目进度的跟进

产品经理5人： 确定需求，给出系统原形

前端8人：负责系统的前台页面，静态页面等，美工3人

后端20人：实现商品，订单，设计等开发

测试5人：对各系统进行测试，抗压等

运维5人：负责项目的部署发布，更新维护等

2.



网络拓扑图（22台服务器集群）

3.整个工程分为8个工程，细分为26个子工程。

|  |  |  |
| --- | --- | --- |
| 序号 | 项目名称 | 各子系统 |
| 业务子系统 | | |
| *1* | jt-web | 前台商城系统：用户可以访问商城首界面，查看不同分类下的商品，浏览商品的详细信息，并可以查询商品，将商品加入购物车，最终提交订单，还包括用户注册和登录。 |
| *2* | jt-manage | 后台管理系统：商品分类管理、商品信息管理、商品规格属性、注册用户管理以及CMS内容发布管理等功能。  包括：jt-manage-mapper/pojo/service/web四个子项目 |
| *3* | jt-cart | 购物车系统：未登录商品选择，登录商品选择，修改商品数量，计算支付金额，下单提交到订单系统。 |
| *4* | jt-order | 订单系统：提供下单、查询订单、修改订单状态、定时处理订单。 |
| *5* | jt-search | 搜索系统：提供商品的搜索功能。 |
| 支撑子系统 | | |
| *6* | jt-parent | jar包依赖管理 |
| *7* | jt-common | 公用工具类 |
| *8* | jt-sso | 单点登录系统：为多个系统之间提供用户登录凭证以及查询登录用户的信息。 |

1. 开发环境

### 开发工具和环境

Windows7/CentOS 5.6

Power Designer 15.6

Eclipse Mars Release 4.5.0

JDK 1.7 必须。一些高级应用需要高版本支持

Maven 3.3.9

Tomcat 7.0.59

Nginx 1.9.0

Redis 2.8.9

MySQL 5.5.27（注意5.0导入jtdb.sql会报错必须升级） + Amoeba

整个系统采用大型项目开发方式，使用maven继承和聚合，形成多个子项目，方便团队开发、调试和一键部署。使用主流大型项目框架SpringMVC+Spring+Mybatis，采用MySQL数据库集群。基于通用Mapper技术，使用SysMapper实现单表CRUD操作完全封装，用户无需写单表的“任何”SQL语句。各层采用包扫描机制，自动发现新的类。利用注解方式，实现各层之间类的调用。页面通过ajax get/post方式提交数据，通过springMVC从页面直接获得参数。采用jackson技术，将对象转换为json字符串，实现各子系统之间的数据的无缝传递。返回对象为获取更丰富的信息，采用SysResult对象封装页面所需要的数据和提示信息，页面通过JSTL进行解析。后台系统采用easyUI，除了SysResult还采用EasyUIResult对象。

# 1.系统后台

1. 根据业务纵向拆分，后台系统属于业务之一， 在根据模块水平分拆
2. 添加toncat插件，添加jetty插件支持实现热部署（就是在应用正在运行的时候升级软件，却不需要重新启动应用）

1.环境商品分类（分类树）

基础表BasePojo

Created(创建时间) updated(修改时间)

实现序列化

商品分类表ItemCat

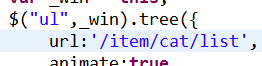
Id（商品分类的id） parentId(父类目的id) name名称

staatus(状态1正常 2删除 ) sortOrder(排序号)

isPartent(是否是父节点一级目录)



点击选着类目是EasyUI的一个button按钮

发送ajax请求

通过EasyUI.Tree异步加载

富客户端方式，div+css+ajax局部属性；ajax异步提交+访问后台获取数据，数据格式：json，调用回调进行对json处理。高度封装，好处：开发者把注意力都放在业务的处理，代码量就少。坏处：初学者不易理解。（学习成本高）

1. 商品分类弹出树，和ztree非常相似，ajax+json。共性都先要组织一个json串，查询手册或者demo，查询json格式要求，然后按它的要求的格式构造vo对象。在后台组织数据，利用springmvc的注解将返回的java的对象转换成json字符串，前台EasyUI.tree的js进行回调，解析json把响应数据放入动态插入的div树结构中（渲染）。
2. 异步tree组成：第一次调用（只加载根节点，数据库表设计根的父的id，设置0）和之后调用（点击树，打开某个分支时，js判断，如果这个分支没有打开过，把当前节点的id传递到后台查询这个节点下的所有子节点。然后返回展现；如果这个分支已经打开过，不会再出向后台发起请求，数据不会再次加载，而直接从浏览器缓存中获取）。
3. Vo对象的格式，要有text和state属性 text代表名字

State代表状态是打开还是关闭 isparent? Closed:open

因为简单所以就不创建vo对象了，放在了po对象中

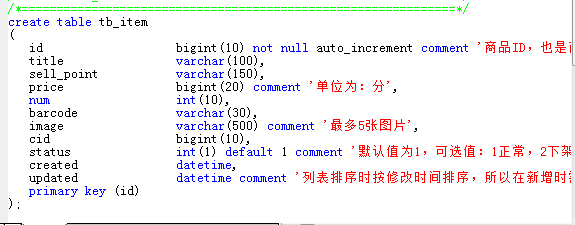
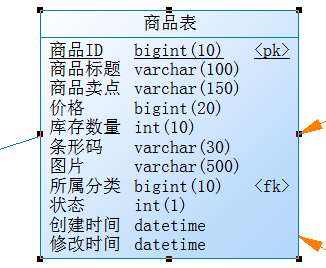
商品分类树总结：利用了EasyUI.tree组件，完成了加载树，要求返回json的格式要带有text和states属性，text指名字，state指状态，打开还是关闭，叶子节点就打开，其他关闭，点击EasyUI.button按钮，发送ajax请求，根据根节点id=0查询，返回json串，回调，解析json

进行页面的渲染，

之后点击树，判断状态是打开的还是关闭的，是关闭的再次带着当前id发起请求，查询此id下的所有子节点，然后回显，如果打开过的，就不会发起请求直接到浏览器缓存中获取

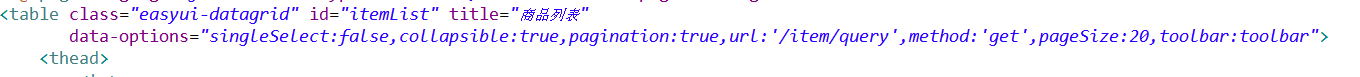
1. 商品列表（按照修改时间排序查出所有列表分页）

当页面自动加载，发起请求





2.1 EasyUI-datagrid列表组件，首页在页面的table，只写标题，然后会自动发起ajax请求



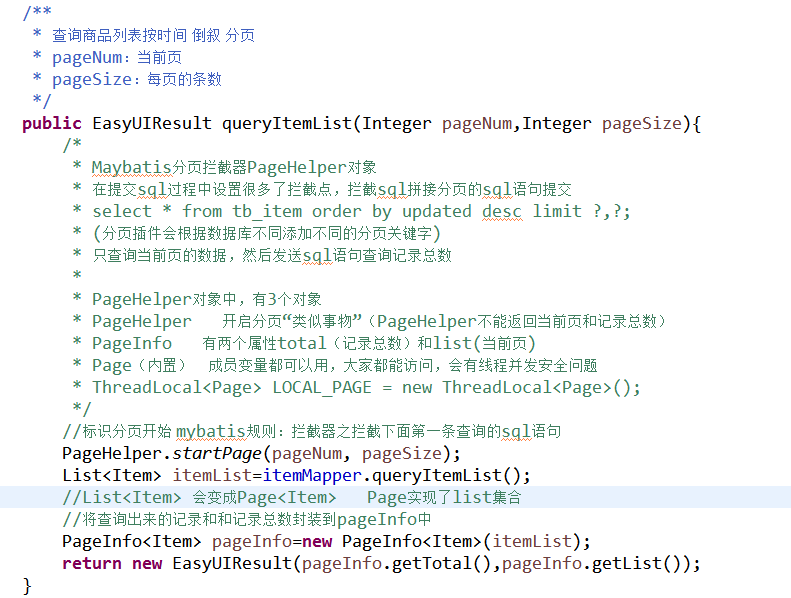
Pagination：true分页，分页条自动产生需要后台分页插件的支持

[http://localhost:8081/item/query?Page=1当前页&rows=20每页的条数](http://localhost:8081/item/query?Page=1当前&rows=20每页的条数)

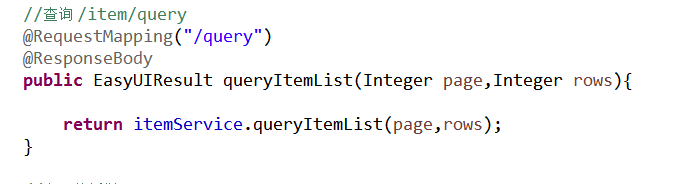
2.2返回符合格式的json 当前页，记录总数 构造符合json串的vo对象(EasyUIResult)，记录总数total 当前页 rows

开发步骤：

1. 通用Mapper不支持排序，要手写sql
2. Service层

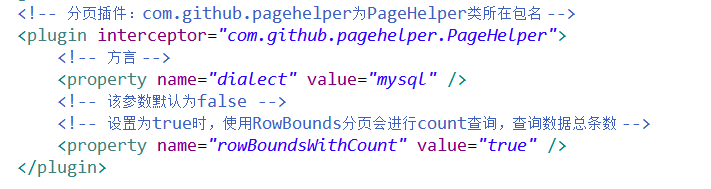


3.在Controller层中，要返回EasyUIResult对象，里面封装了当前页和记录总数，EasyUI 规定的返回格式



1. EasyUI.datagrid自动解析json串

分页插件



逻辑分页：先把所有数据查出来，挑去用户要求的某页（1次查询，数据量少使用）

物理分页：查询数据库只查询当页数据，顺便把记录总数返回（2次查询，数据量大时）

2.按照修改时间分页加载商品列表总结：

利用EasyUI.datagrid组件（列表组件），点击商品列表带着当前页和每页的条数发起ajax请求，使用 data-options 来初始化属性，里面有个支持分页的属性需要后台分页配合，EasyUI.datagrid要求返回的json格式必须带有当前页内容和记录总数，构造vo对象easyuiResult。（里面有total(当前页)和rows（记录总数））因为通用mapper不支持排序要手写sql。

Mysql的分页插件有三个对象，pagehelper,pageinfo,page（内置对象封装了threadlocal）

pageHelperl拦截下面第一条查询语句，根据数据库拼接不同的分页查询

然后提交，同时也发起查询记录总数的sql.（在mybatis文件中有配置该属性）然后将查询到的内容保存在pageinfo中,从pageinfo中取出list和total(当前页和记录总数)设置到vo对象中并返回，EasyUI。Datagrid解析并渲染页面。

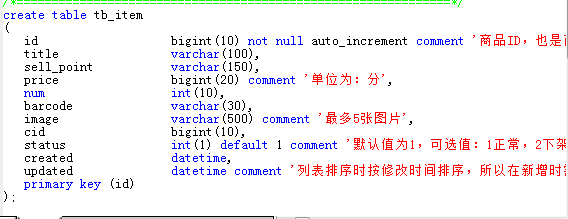
然后就是商品的删改，使用通用mapper搞定，其中增加商品时，因为商品表中有个商品描述的大字段，所以进行拆表了。所以增加商品就没有使用通用mapper了

如果数据库表中有大字段，索引失效，全表遍历，但数据量大时，即使满足条件的记录在最前面，依然要遍历所有，性能下降非常多。

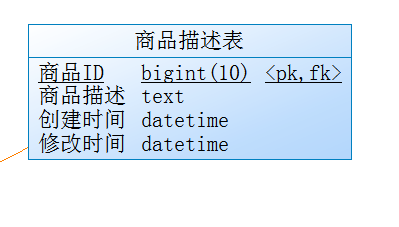
拆表，本来字段属于商品信息的，应该放在商品表中，但是因为它是大字段，把它单独拆成一张表。关联关系：一对一。特征：主键既外键。（修改和删除代码就非常方便）

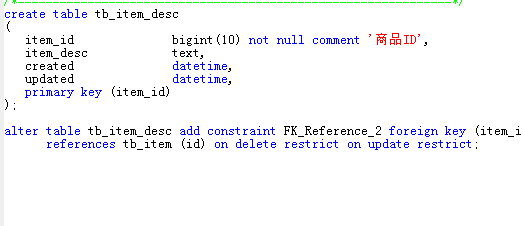
1. 商品增删改（商品描述信息）





因为商品描叙是个大字段，所以进行拆表





3.1点击商品提交



3.2发送ajax请求

.serialize序列化类似get请求参数串？name1=s&age=18

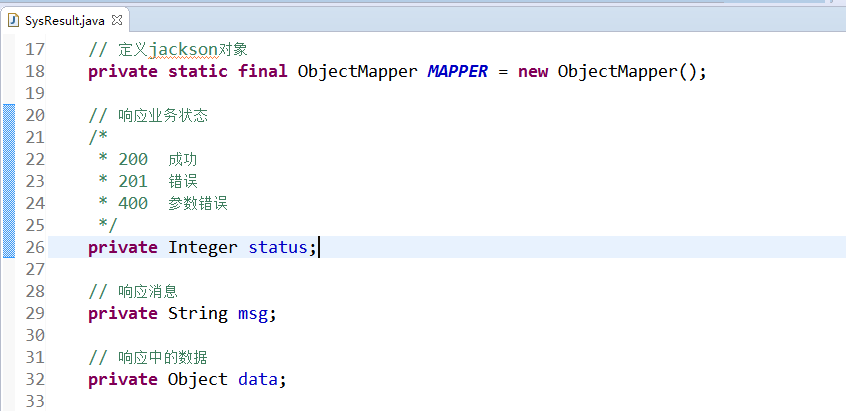
会自动将form表单所有的表单控件name跟value进行拼串



3.3返回对象SysResult 有属性status响应状态值200的代表成功 201出错 400内部错误

Msg 错误异常的异常信息，

Data操作的对象（返回更丰富的信息）



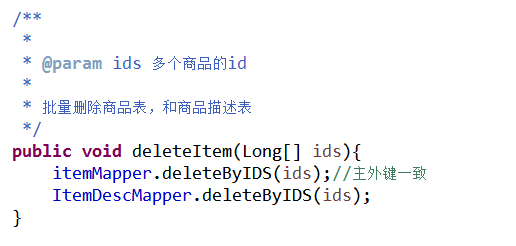
商品的新增



商品的修改



商品的批量删除根据id



根据ids删除对象，根据不为空的字段修改

商品的状态修改根据id批量修改 因为还有数星星status

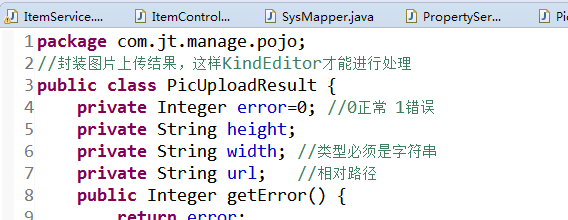
所以通用Mapper不支持摇自己写sql。



1. 图片上传

5.1 KindEditor组件实现图片上传

5.2KindEditor回显图片时有json格式要求，下面的对象就是按这个格式要求实现



图片上传

1.KindEditor组件实现图片上传

2.js组件能否操作服务器资源，例如：上传图片，js弱语言，在设计之初，约 定：你为了安全性，不能访问服务器资源，不能操作服务器目录， 写 文件

3.KindEditor需要服务器别的开发语言配合（比如php），controller

4.springmvc通过文件上传解析器，在controller写一个方法，方法写一个参数 MultipartFile

5.利用flash可以操作本地资源，批量上传

图片上传步骤：

1.获取原始文件名，开展名字

2.判断是否合法，判断文件后缀是否是图片类型 .jpg .gif .png

3.判断是否为木马规则：判断上传的文件是否是文件，如果是文件，可以获取 到height(高 度)和width(宽度)

4.KindEditor构造一个json。有4个属性 url+height+width(字符类型)+error（null, 必 须设置0是正常 1是异常）

5.创建目录C：\jt-upload\images\2017\05\06

6.修改图片的名称：当前的时间+3位的随机数+扩展名

7.保存

