

[今日课程大纲]

电商项目规格参数管理

[知识点详解]

一.规格参数管理介绍

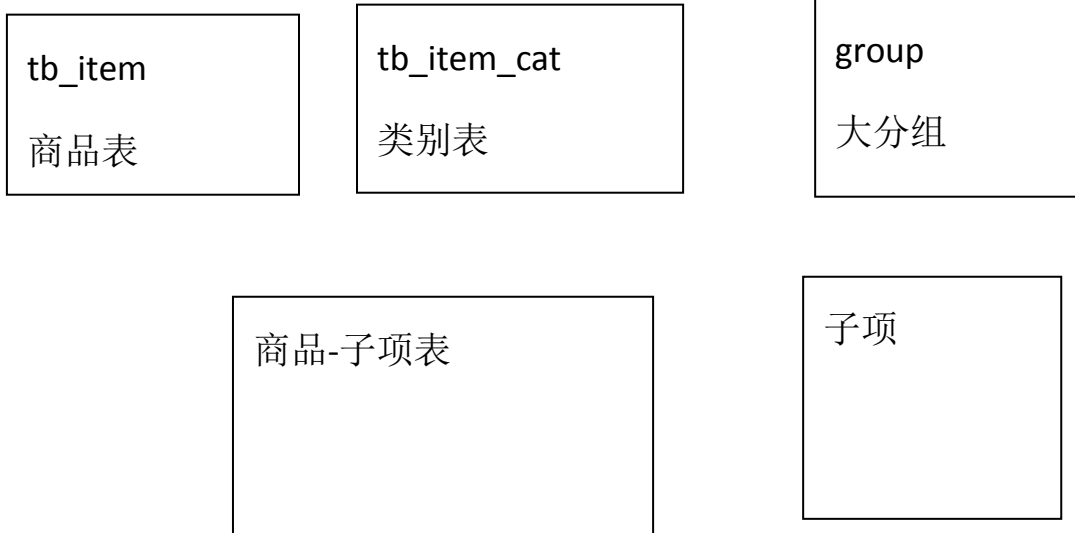
1.优化规格参数数据库设计

2.需求:

2.1 每个分类有不同的分类模板.

2.2 根据模板每个商品有具体信息

3.传统数据库设计



4.优化数据库设计

4.1 前提: group 和子项的对应关系是比较灵活

4.2 商品-子项表数据比较多.

4.3 优化后:

4.3.1 所有的规格参数模板设置到一个列中这个列中存储特定格式的数据.把数据存储为 json 数据格式.

4.3.2 设置一个表,设置某个列中存储 json 数据格式

4.3.2.1 目的:存储每个商品具体的规格信息.

二. 规格参数查询实现

1. 在 ego-service 中 TbItemCatDubboService 及实现类添加

```
/**
 * 根据类目 id 查询
 * @param id
 * @return
 */
TbItemCat selById(long id);

@Override
public TbItemCat selById(long id) {
    return tbItemCatMapper.selectByPrimaryKey(id);
}
```

2. 在 ego-service 新建 TbItemParamDubboService 及实现类

2.1 带有 withBlobs()的方法会查询 text 类型的列

```
public interface TbItemParamDubboService {

    /**
```

```
* 分页查询数据,
* @param page
* @param rows
* @return 包含:当前也显示数据和总条数
*/

EasyUIDataGrid showPage(int page,int rows);
}

public class TbItemParamDubboServiceImpl implements
TbItemParamDubboService{

    @Resource

    private TbItemParamMapper tbItemParamMapper;

    @Override

    public EasyUIDataGrid showPage(int page, int rows) {

        //先设置分页条件

        PageHelper.startPage(page, rows);

        //设置查询的 SQL 语句

        //XXXXExample() 设置了什么,相当于在 sql 中 where 从句
        中添加条件

        //如果表中有一个或一个以上的列是 text 类型. 生成的方法
        xxxxxxxxWithBlobs()

        //如果使用 xxxxWithBlobs() 查询结果中包含带有 text 类
```

的列,如果没有使用 `withblobs()` 方法不带有 `text` 类型。

```
List< TbItemParam> list =
tbItemParamMapper.selectByExampleWithBLOBs(new
TbItemParamExample());

//根据程序员自己编写的 SQL 语句结合分页插件产生最终结果,封装到 PageInfo

PageInfo< TbItemParam> pi = new PageInfo<>(list);

//设置方法返回结果

EasyUIDataGrid datagrid = new EasyUIDataGrid();

datagrid.setRows(pi.getList());

datagrid.setTotal(pi.getTotal());

return datagrid;

}

}
```

3. 在 ego-service-impl 的 applicationContext-dubbo.xml 注册服务接口

```
<!-- 规格参数 -->

<dubbo:service
interface="com.ego.dubbo.service.TbItemParamDubboService"
ref="tbItemParamDubboServiceImpl"></dubbo:service>

<bean id="tbItemParamDubboServiceImpl"
```

```
class="com.ego.dubbo.service.impl.TbItemParamDubboServiceImpl"></bean>
```

4. 在 ego-manage 中新建 com.ego.manage.pojo.TbItemParamChild,作用封装 jsp 中要的所有数据

```
public class TbItemParamChild extends TbItemParam {  
    private String itemCatName;
```

5. 在 ego-manage 中新建 TbItemParamService 及实现类

```
public interface TbItemParamService {  
  
    /**  
     * 分页显示商品规格参数  
     * @param page  
     * @param rows  
     * @return  
     */  
  
    EasyUIDataGrid showPage(int page,int rows);  
}
```

@Service

```
public class TbItemParamServiceImpl implements  
TbItemParamService {  
  
    @Reference  
  
    private TbItemParamDubboService
```

```
tbItemParamDubboServiceImpl;

@Reference

private TbItemCatDubboService
tbItemCatDubboServiceImpl;

@Override

public EasyUIDataGrid showPage(int page, int rows) {

    EasyUIDataGrid datagrid =
tbItemParamDubboServiceImpl.showPage(page, rows);

    List<TbItemParam> list = (List<TbItemParam>)
datagrid.getRows();

    List<TbItemParamChild> listChild = new
ArrayList<>();

    for (TbItemParam param : list) {

        TbItemParamChild child = new
TbItemParamChild();

        child.setCreated(param.getCreated());

        child.setId(param.getId());

        child.setItemCatId(param.getItemCatId());

        child.setParamData(param.getParamData());

        child.setUpdated(param.getUpdated());
```

```
        child.setItemCatName(tbItemCatDubboServiceImpl.selB  
yId(child.getItemCatId()).getName());  
  
        listChild.add(child);  
    }  
    datagrid.setRows(listChild);  
    return datagrid;  
}  
  
}
```

6. 在 ego-manage 中新建 TbItemParamController

```
@Controller  
  
public class TbItemParamController {  
    @Resource  
    private TbItemParamService tbItemParamServiceImpl;  
    /**  
     * 规格参数-分页显示  
     * @param page  
     * @param rows  
     * @return  
     */  
}
```

```
@RequestMapping("item/param/list")
@ResponseBody

public EasyUIDataGrid showPage(int page,int rows){

    return tbItemParamServiceImpl.showPage(page,
rows);
}
}
```

三. 规格参数批量删除

1. 在 ego-service 的 TbItemParamDubboService 及实现类添加

```
/**
 * 批量删除
 * @param ids
 * @return
 */
int delByIds(String ids) throws Exception;

@Override

public int delByIds(String ids) throws Exception {

    int index = 0 ;

    String[] idStr = ids.split(",");

    for (String id : idStr) {
```



```

        index

+=tbItemParamMapper.deleteByPrimaryKey(Long.parseLong
(id));

    }

    if(index==idStr.length){

        return 1;

    }else{

        throw new Exception("删除失败.可能原因:数据已经不存在");

    }

}

```

2. 在 ego-service-impl 的 applicationContext.xml 中添加 rollback-for

```

<!-- 声明式事务 -->
<tx:advice id="txAdvice" transaction-manager="txManage">
    <tx:attributes>
        <tx:method name="ins*" rollback-for="java.lang.Exception"/>
        <tx:method name="del*" rollback-for="java.lang.Exception"/>
        <tx:method name="upd*" rollback-for="java.lang.Exception"/>
        <tx:method name="*" read-only="true"/>
    </tx:attributes>
</tx:advice>

```

3. 在 ego-manage 的 TblItemParamService 及实现类添加

```

/**

    * 批量删除

    * @param ids

    * @return

    */

```

```
int delete(String ids) throws Exception;
```

```
@Override
```

```
public int delete(String ids) throws Exception {  
    return tbItemParamDubboServiceImpl.delByIds(ids);  
}
```

4. 在 ego-manage 的 TblItemParamController 添加

```
/**  
 * 批量删除规格参数  
 * @param ids  
 * @return  
 */  
  
@RequestMapping("item/param/delete")  
@ResponseBody  
public EgoResult delete(String ids){  
    EgoResult er = new EgoResult();  
  
    try {  
        int index = tbItemParamServiceImpl.delete(ids);  
        if(index==1){  
            er.setStatus(200);  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```

        er.setData(e.getMessage());
    }

    return er;
}

```

5. 在 ego-manage 的 item-param-list.jsp 中修改 js 代码

```

$.messager.confirm('确认','确定删除ID为 '+ids+ ' 的商品规格吗?',function(r){
    if (r){
        var params = {"ids":ids};
        $.post("/item/param/delete",params, function(data){
            if(data.status == 200){
                $.messager.alert('提示','删除商品规格成功!',undefined,function(){
                    $("#itemParamList").datagrid("reload");
                });
            }else{
                $.messager.alert('提示','删除商品规格失败!<br/>原因:' +data.data,undefined,function(){
                    $("#itemParamList").datagrid("reload");
                });
            }
        });
    }
});

```

四.规格参数新增

1. 在 ego-service 的 TbItemParamDubboService 及实现类添加

```

/**
 * 根据类目 id 查询参数模板
 * @param catId
 * @return
 */
TbItemParam selByCatid(long catId);

@Override
public TbItemParam selByCatid(long catId) {
    TbItemParamExample example = new
    TbItemParamExample();
}

```

```
example.createCriteria().andItemCatIdEqualTo(catId)
;

List< TbItemParam > list =
tbItemParamMapper.selectByExampleWithBLOBs(example);

if(list!=null&&list.size()>0){

    //要求每个类目只能有一个模板

    return list.get(0);

}

return null;

}
```

2. 在 ego-manage 的 TbItemParamService 及实现类添加

```
/**
 * 根据类目 id 查询模板信息
 * @param catId
 * @return
 */
EgoResult showParam(long catId);

@Override
public EgoResult showParam(long catId) {

    EgoResult er = new EgoResult();

    TbItemParam param =
```

```
tbItemParamDubboServiceImpl.selByCatid(catId);

    if(param!=null){

        er.setStatus(200);

        er.setData(param);

    }

    return er;

}
```

3. 在 ego-manage 的 TblItemParamController 添加

```
/**
 * 点击商品类目按钮显示添加分组按钮
 * 判断类目是否已经添加模板
 * @param catId
 * @return
 */
@RequestMapping("item/param/query/itemcatid/{catId}")
@ResponseBody
public EgoResult show(@PathVariable long catId){

    return tbItemParamServiceImpl.showParam(catId);

}
```

4. 在 ego-service 中 TblItemParamDuboService 及实现类添加

```
/**
```

```
* 新增,支持主键自增  
* @param param  
* @return  
*/  
  
int insParam(TbItemParam param);
```

```
@Override  
  
public int insParam(TbItemParam param) {  
    return tbItemParamMapper.insertSelective(param);  
}
```

5. 在 ego-manage 的 TbItemParamService 及实现类添加

```
/**  
 * 新增模板信息  
 * @param param  
 * @return  
 */  
  
EgoResult save(TbItemParam param);
```

```
@Override  
  
public EgoResult save(TbItemParam param) {  
    Date date = new Date();  
    param.setCreated(date);  
    param.setUpdated(date);  
    int index =
```

```
tbItemParamDubboServiceImpl.insParam(param);

    EgoResult er = new EgoResult();

    if(index>0){

        er.setStatus(200);

    }

    return er;

}
```

6. 在 ego-manage 的 TbItemParamController 中添加

```
/**
 * 商品类目新增
 * @param param
 * @param catId
 * @return
 */

@RequestMapping("item/param/save/{catId}")
@ResponseBody
public EgoResult save(TbItemParam
param,@PathVariable long catId){

    param.setItemCatId(catId);

    return tbItemParamServiceImpl.save(param);

}
```

四. 商品新增时同时新增商品规格参数信息

1. 在 ego-service 的 TbItemDubboService 及实现类修改如下

```
/**
 * 新增包含商品表和商品描述表
 * @param tbItem
 * @param desc
 * @return
 */
int insTbItemDesc(TbItem tbItem, TbItemDesc desc, TbItemParamItem paramItem) throws Exception;

@Override
public int insTbItemDesc(TbItem tbItem, TbItemDesc desc, TbItemParamItem paramItem) throws Exception {
    int index = 0;
    try {
        index = tbItemMapper.insertSelective(tbItem);
        index +=
        tbItemDescMapper.insertSelective(desc);

        index += tbItemParamItemMapper.insertSelective(paramItem);
    }
}
```



```
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    if(index==3){  
        return 1;  
    }else{  
        throw new Exception("新增失败,数据还原");  
    }  
}
```

2. 在 ego-manage 中 TbItemService 及实现类修改如下

```
/**  
 * 商品新增  
 * @param item  
 * @param desc  
 * @return  
 */  
int save(TbItem item ,String desc,String itemParams)  
throws Exception;
```

```
@Override  
public int save(TbItem item, String desc,String  
itemParams) throws Exception {  
    long id = IDUtils.genItemId();
```

```
item.setId(id);

Date date = new Date();

item.setCreated(date);

item.setUpdated(date);

item.setStatus((byte) 1);


TbItemDesc itemDesc = new TbItemDesc();

itemDesc.setItemDesc(desc);

itemDesc.setItemId(id);

itemDesc.setCreated(date);

itemDesc.setUpdated(date);


TbItemParamItem paramItem = new TbItemParamItem();

paramItem.setCreated(date);

paramItem.setUpdated(date);

paramItem.setItemId(id);

paramItem.setParamData(itemParams);


int index = 0;
```

```
        index =  
        tbItemDubboServiceImpl.insTbItemDesc(item,  
        itemDesc,paramItem);  
        System.out.println("index:" + index);  
        return index;  
    }
```

3. 在 ego-manage 的 TbItemController 中修改如下

```
/**  
 * 商品新增  
 * @param item  
 * @param desc  
 * @return  
 */  
@RequestMapping("item/save")  
@ResponseBody  
public EgoResult insert(TbItem item,String  
desc,String itemParams){  
    EgoResult er = new EgoResult();  
    int index;  
    try {  
        index = tbItemServiceImpl.save(item,  
desc,itemParams);
```

```
        System.out.println("controller:index:"+index);  
        if(index==1){  
            er.setStatus(200);  
        }  
    } catch (Exception e) {  
//        e.printStackTrace();  
        er.setData(e.getMessage());  
    }  
    return er;  
}
```