

[今日课程大纲]

电商项目商品新增

[知识点详解]

一.商品类目显示

1.在 ego-service 中创建 TbItemCatDubboService 接口及实现类

```
public interface TbItemCatDubboService {  
  
    /**  
     * 根据父类目 id 查询所有子类目  
     * @param pid  
     * @return  
     */  
  
    List<TbItemCat> show(long pid);  
}
```

```
public class TbItemCatDubboServiceImpl implements  
TbItemCatDubboService{  
  
    @Resource  
  
    private TbItemCatMapper tbItemCatMapper;  
  
    @Override  
  
    public List<TbItemCat> show(long pid) {  
  
        TbItemCatExample example =new TbItemCatExample();
```

```
example.createCriteria().andParentIdEqualTo(pid);

return tbItemCatMapper.selectByExample(example);
}

}
```

2.在 ego-service-impl 的 applicationContext-dubbo.xml 注册发布服务接口



```
<!-- 商品类目服务 -->

<dubbo:service
interface="com.ego.dubbo.service.TbItemCatDubboService" ref="tbItemCatDubboServiceImpl"></dubbo:service>

<bean id="tbItemCatDubboServiceImpl"
class="com.ego.dubbo.service.impl.TbItemCatDubboServiceImpl"></bean>
```

3.在 ego-commons 中新建一个 EasyUITree 类,专门封装 EasyUI 中 Tree 组件每个节点对象的数据

3.1 state 是 EasyUI 中 node 节点的固定属性之一.可取值只有"closed"和"open",如果取值为"closed"显示节点为文件夹

 Folder ,如果取值为"open"显示节点为文件  File 2

```
public class EasyUITree {

private long id;
```

```
private String text;  
  
private String state;
```

4.在 ego-manage 中新建 TbItemCatService 接口及实现类

```
public interface TbItemCatService {  
  
    /**  
     * 根据父菜单 id 显示所有子菜单  
     * @param pid  
     * @return  
     */  
  
    List<EasyUiTree> show(long pid);  
}
```

```
@Service  
  
public class TbItemCatServiceImpl implements  
TbItemCatService {  
  
    @Reference  
  
    private TbItemCatDubboService  
tbItemCatDubboServiceImpl;  
  
    @Override  
  
    public List<EasyUiTree> show(long pid) {  
  
        List<TbItemCat> list =  
tbItemCatDubboServiceImpl.show(pid);  
  
        List<EasyUiTree> listTree = new ArrayList<>();
```

```
        for (TbItemCat cat : list) {  
            EasyUiTree tree = new EasyUiTree();  
            tree.setId(cat.getId());  
            tree.setText(cat.getName());  
  
            tree.setState(cat.getIsParent()?"closed":"open");  
            listTree.add(tree);  
        }  
        return listTree;  
    }  
}
```

5.在 ego-manage 中新建 TbItemCatController

5.1 在控制器方法中设置参数 id 默认值为 0,第一请求时没有参数,希望查询出所有一级菜单.一级菜单的父 id 为 0

```
@Controller  
  
public class TbItemCatController {  
    @Resource  
    private TbItemCatService tbItemCatServiceImpl;  
    /**  
     * 显示商品类目  
     * @param id  
     * @return
```

```

    */

    @RequestMapping("item/cat/list")

    @ResponseBody

    public List<EasyUiTree>

showCat(@RequestParam(defaultValue="0") long id){

        return tbItemCatServiceImpl.show(id);

    }

}

```

二. 实现商品新增-图片上传

1. 需要在 linux 中安装 vsftpd,安装后实现使用 ftpclient 完成图片上传功能.
2. 在 vsftpd 所在服务器安装 nginx,实现图片回显
3. 在 ego-parent 中 pom.xml 引入 ftpclient 的 jar

```
<ftpclient-version>3.3</ftpclient-version>
```

```
<!-- ftpclient -->
```

```
    <dependency>
```

```
        <groupId>commons-net</groupId>
```

```
        <artifactId>commons-net</artifactId>
```

```
        <version>${ftpclient-version}</version>
```

```
    </dependency>
```

4. 在 ego-commons 中 pom.xml 中引入 ftpclient 的 jar

```
<dependencies>

    <dependency>

        <groupId>commons-net</groupId>

        <artifactId>commons-net</artifactId>

    </dependency>

</dependencies>
```

5. 在 ego-commons 中 com.ego.common.utils 添加两个工具类

```
com.ego.common.utils
├── FtpUtil.java
└── IDUtils.java
```

6. 在 ego-commons 中 src/main/resources 下新建 commons.properties

```
ftpclient.host = 192.168.249.129

ftpclient.port = 21

ftpclient.username = ftpuser

ftpclient.password = ftpuser

ftpclient.basepath = /home/ftpuser

ftpclient.filepath = /
```

7. 在 ego-manage 的 applicationContext-spring 中添加属性文件扫描

```
<!-- 扫描属性文件 -->

    <context:property-placeholder

location="classpath:commons.properties"/>
```

8. 在 ego-manage 中 PicService 接口及实现类

8.1 方法返回值是 Map 原因

8.1.1 jsp 中使用 kindeditor 的多文件上传插件,要求返回值为固定格式,成功和失败返回值包含内容不一样,不建议使用新建一个类方式.

```
public interface PicService {  
  
    /**  
  
     * 文件上传  
  
     * @param file  
  
     * @return  
  
     */  
  
    Map<String,Object> upload(MultipartFile file) throws  
IOException;  
}
```

```
@Service  
  
public class PicServiceImpl implements PicService {  
  
    @Value("${ftpclient.host}")  
  
    private String host;  
  
    @Value("${ftpclient.port}")  
  
    private int port;  
  
    @Value("${ftpclient.username}")  
  
    private String username;
```

```
@Value("${ftpclient.password}")

private String password;

@Value("${ftpclient.basepath}")

private String basePath;

@Value("${ftpclient.filepath}")

private String filePath;


@Override

public Map<String,Object> upload(MultipartFile file)
throws IOException {

    String genImageName =
IDUtils.genImageName()+file.getOriginalFilename().sub
string(file.getOriginalFilename().lastIndexOf("."));

    boolean result = FtpUtil.uploadFile(host, port,
username, password, basePath, filePath, genImageName,
file.getInputStream());

    Map<String,Object> map = new HashMap<>();

    if(result){

        map.put("error", 0);

        map.put("url","http://" +
host+"/"+genImageName);

    }else{
```



```
        map.put("error", 1);

        map.put("message", "图片上传失败");

    }

    return map;

}

}
```

9. 在 ego-manage 中新建 PicController

```
@Controller

public class PicController {

    @Resource

    private PicService picServiceImpl;

    /**
     * 图片上传
     * @param uploadFile
     * @return
     */

    @RequestMapping("pic/upload")

    @ResponseBody

    public Map<String, Object> upload(MultipartFile

uploadFile){
```

```
Map<String,Object> map = new HashMap<>();

try {

    map= picServiceImpl.upload(uploadFile);

} catch (IOException e) {

    e.printStackTrace();

    map.put("error", 1);

    map.put("message", "上传图片时服务器异常");

}

return map;

}

}
```

三.实现商品新增

1. 在 ego-service 中 TbItemDubboService 接口及实现类添加

1.1 当出错时继续运行

1.2 通过手动抛出异常可以使用 dubbo 的 provider 和 consumer 交互的问题.

```
/**

 * 新增包含商品表和商品描述表

 * @param tbItem

 * @param desc

 * @return
```

```

        */

        int insTbItemDesc(TbItem tbItem, TbItemDesc desc)
        throws Exception;

        @Override

        public int insTbItemDesc(TbItem tbItem, TbItemDesc
        desc) throws Exception {

            int index =0;

            try {

                index= tbItemMapper.insertSelective(tbItem);

                index+=

                tbItemDescMapper.insertSelective(desc);

            } catch (Exception e) {

                e.printStackTrace();

            }

            if(index==2){

                return 1;

            }else{

                throw new Exception("新增失败,数据还原");

            }

        }
    }

```

2. 在 ego-service-impl 的 applicationContext.xml 中添加回滚

```
<!-- 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*" />
        <tx:method name="upd*" />
        <tx:method name="*" read-only="true"/>
    </tx:attributes>
</tx:advice>
```

3. 在 ego-manage 中 TblItemService 及实现类添加方法

3.1 方法没有处理异常,因为希望在控制器捕获到异常信息

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

@Override
public int save(TbItem item, String desc) 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);

int index = 0;

index =
tbItemDubboServiceImpl.insTbItemDesc(item, itemDesc);

System.out.println("index:" + index);

return index;

}
```

4. 在 ego-commons 中 EgoResult 添加 Object data;

```
public class EgoResult {

    private int status;

    private Object data;
```

5. 在 ego-manage 的 TbItemController 添加方法

```
@RequestMapping("item/save")

@ResponseBody
```

```

public EgoResult insert(TbItem item,String desc){
    EgoResult er = new EgoResult();

    int index;

    try {

        index = tbItemServiceImpl.save(item, desc);

        System.out.println("controller:index:"+index);

        if(index==1){

            er.setStatus(200);

        }

    } catch (Exception e) {
//        e.printStackTrace();

        er.setData(e.getMessage());

    }

    return er;

}

```

6. 在 item-add.jsp 中修改了,添加错误信息提示

```

114 $.post("/item/save",$("#itemAddForm").serialize(),function(data){
115     if(data.status == 200){
116         $.messager.alert('提示','新增商品成功!');
117     }else{
118         $.messager.alert('提示','新增商品失败!<br/> 原因:'+data.data);
119     }
120 });

```