后台程序流程

1. 输入请求 index，进入 后台管理页面
2. IndexController 控制进行映射 “/index”和控制返回的文件“index”
3. 利用restful风格的路径请求来处理，查询和新增列表显示的统一方法处理， "/page/{moduleName}"
4. 查询所有商品信息 /item/query 在前端找到请求路径 并传递过来当前页和每页的行数
5. 在controller在定义查找的方法 itemService.findItemByPage();
6. 定义ItemService 和ItemServiceImpl
7. 定义itemmapper 继承通用mapper sysmapper 复杂的sql语句需要自己定义
8. 商品表中的商品类目id是商品类目表的外键 ，还需要根据此id查询商品分类
9. 根据前端显示的信息 //localhost:8091/item/cat/queryItemName 要定义queryItemName方法 因为使用的是ajax的post请求所以在请求路径中不包括？参数，但实际是传递了一个id
10. 定义ItemCatController, 自动注解 ItemCatService , 定义方法queryItemCatNameById，返回一个字符串 ，如果要加@responsebody注解 如果是对象那么是utf-8 如果是字符串那么是iso-8859-1 需要在requestmapper注解中添加属性produce=“text/html;charset=utf-8”即可
11. 至此 商品的查询已经全部结束
12. 定义
13. ---------------------------------
14. 查询完商品之后需要新增商品，新增商品第一步需要查询商品的分类 然后返回商品分类列表。
15. 商品分类列表的逻辑是 子节点的加载依赖于父节点的状态。当展开一个封闭的节点，如果节点没有加载子节点，它将会把节点id的值作为**http请求参数并命名为'id'**，通过URL发送到服务器上面检索子节点 ，首先初始parentid为0，根据Parentid为0查到id为1，因为一级菜单还有子菜单，所以再,1作为parentid传入，查到id为2 ，直到查到最后一级，没有子菜单，根据前端程序分析得出后端需要传回一个easyUITree的对象，参数包括当前目录的id ，名称text 以及是否有下级的状态 state
16. 创建vo对象 EasyUITree 参数 id ，text，state
17. 在controller中调用itemCatService.queryNodeByParentId(parentId);方法，返回包含对象为easyUITree的list集合 ，注意前端传值不一致，需要注解，并添加默认值为0；
18. 在itemCatService接口中定义相关方法。
19. 在itemCatServiceImpl中具体实现queryNodeByParentId(parentId)方法，逻辑是通过通用mapper查询，那么需要传递一个itemcat对象，先新建一个itemcat对象，然后返回一个包含itemcat对象的list集合。通过for循环，对每一个itemcat对象get取值，并通过set赋值给新建的每一个easyuitree对象，赋值完添加到新建的arraylist集合中，最后返回。
20. 商品列表树已经显示完成，接下来就是进行商品的添加和修改。
21. --------------------------------------------------
22. 点击提交按钮 请求路径为$.post("/item/save"） 所以需要在itemcontroller中定义save方法，并且需要向后台传递商品的信息 ，传过来的是json串，类似[{k:v},{K:v}]
23. 分析前端代码，因为在使用ajax发起异步请求时，传递的参数使用了$("#itemAddForm").serialize() 使用在后台接收时已经自动封装为了item对象 ，另外还需要接收商品的描述内容 格式为string。可以直接接收使用 ，定义itemService.addItem(item，itemDesc);方法，向前端只需要传递一个封装好的sysresult对象，设置好状态码和信息即可。
24. 在itemServiceImpl 中接收到了item对象和itemDesc，在前端界面没有设置创建时间和修改时间，也没有设置商品状态，因此在itemserviceImpl中要对这三个参数设值，并使用通用mapper插入。