

## [今日课程大纲]

### 门户导航菜单功能需求分析

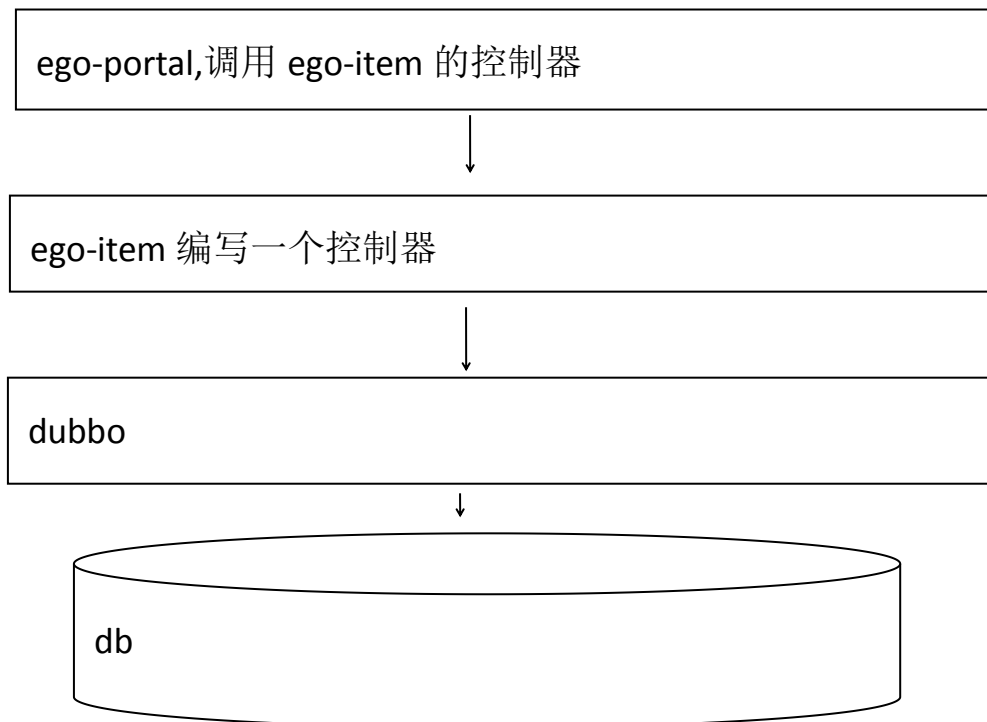
#### jsonp

完成导航菜单功能.

## [知识点详解]

### 一. 门户导航菜单功能需求分析

1. 在 ego-portal 中显示 tb\_item\_cat 商品类目的数据,让 ego-portal 调用 ego-item 中的数据,由 ego-item 调用 dubbo



#### 2. 跨域

- 2.1 一个 Servlet 容器 (tomcat) 中项目调用另一个 servlet 容器 (tomcat) 中项目,称为跨域

3. ajax 在研发时处于安全性考虑不允许跨域请求的.

3.1 解决办法:使用 jsonp

3.2 解决办法:由 ego-portal 访问自己的控制器,自己的控制器访问自己的 service,在自己的 service 中使用 httpClient 调用 ego-item 的控制器方法.

## 二. jsonp

1. jsonp:跨域 ajax 数据请求的解决方案

2. 发展由来

2.1 ajax 不能进行跨域请求(如果 ajax 请求的控制器返回的就是字符串或 json 数据,不能访问)

2.2 发现可以在一个项目直接访问另一个项目的 js 文件.(标签引用还是通过 ajax 访问)

2.3 使用 ajax 访问另一个项目的控制器,但是控制器返回的结果伪装成 js 文件.

3. 使用 jsonp 时服务器端返回的数据格式:

函数名(返回的数据);

4. 在客户端编写代码

4.1 dataType: 一定要设置 jsonp

4.2 jsonp: 传递给服务器的参数名.省略的默认 callback

4.3 jsonCallback: 参数名对应的值,表示最终回调的函数名.省略的

默认值 jquery:一堆数字

4.4 如果直接使用 success:function()对 jsonpCallback 值没有要求.

4.5 如果单独编写了一个 function,必须要求 jsonpCallback 和 function 的名称相同

```
$(function(){  
    $("button").click(function(){  
        $.ajax({  
            url:'http://localhost:9002/demo2',  
            type:'post',  
            dataType:'jsonp',  
            jsonpCallback:'ab12312321c',  
            jsonp:'callback',  
            success:function(data){  
                alert(data+"success");  
            }  
        });  
    })  
})
```

5. 在服务器端添加代码

5.1 使用 spring 对 jackson 封装类 MappingJacksonValue

```
@Controller  
  
public class DemoController {
```

```
@RequestMapping("demo3")
@ResponseBody
public MappingJacksonValue demo(String callback){
    People p = new People();
    p.setId(1);
    p.setName("张三");
    //把构造方法参数转换为 json 字符串并当作最终返回值函数
    的参数
    MappingJacksonValue mjv = new
MappingJacksonValue(p);
    //最终返回结果中函数名
    mjv.setJsonFunction(callback);
    return mjv;
}
}
```

### 三.搭建 ego-portal 和 ego-item

1. 环境和 ego-manage 相同
2. ego-portal 的端口 8082 , ego-item 的端口是 8081
3. 修改 applicationContext-dubbo.xml 注解扫描

## 四. 编写 ego-item 代码

1. 在 ego-item 中新建 com.ego.item.pojo.PortalMenu

1.1 最终 ego-item 返回给 ego-portal 的 jsonp 数据中函数的参数值

```
public class PortalMenu {  
  
    private List<Object> data;
```

2. 在 ego-item 中新建 com.ego.item.pojo.PortalMenuNode

2.1 如果菜单是父菜单,当前菜单的数据都封装到这个类中

```
public class PortalMenuNode {  
  
    private String u;  
  
    private String n;  
  
    private List<Object> i;
```

3. 在 ego-item 中新建 com.ego.item.service.TbItemCatService 及实现类

```
public interface TbItemCatService {  
  
    /**  
     * 查询出所有分类类目并转换为特定类型。  
     * @return  
     */  
  
    PortalMenu showCatMenu();  
}
```

@Service

```
public class TbItemCatServiceImpl implements
```

```
TbItemCatService {

    @Reference

    private TbItemCatDubboService

tbItemCatDubboServiceImpl;

    @Override

    public PortalMenu showCatMenu() {

        // 查询出所有一级菜单

        List<TbItemCat> list =

tbItemCatDubboServiceImpl.show(0);

        PortalMenu pm =new PortalMenu();

        pm.setData(selAllMenu(list));

        return pm;

    }

    /**

     * 最终返回结果所有查询到的结果.

     */

    public List<Object> selAllMenu(List<TbItemCat>

list){

        List<Object> listNode = new ArrayList<>();

        for (TbItemCat tbItemCat : list) {

            if(tbItemCat.getIsParent()){

                PortalMenuNode pmd = new PortalMenuNode();
```

```
pmd.setU("/products/"+tbItemCat.getId()+".html");

pmd.setN("<a
href='/products/"+tbItemCat.getId()+".html'>" + tbItemC
at.getName()+"</a>");

pmd.setI(selAllMenu(tbItemCatDubboServiceImpl.show(
tbItemCat.getId())));

listNode.add(pmd);

}else{

listNode.add("/products/"+tbItemCat.getId()+".html |
"+tbItemCat.getName());

}

}

return listNode;

}

}
```

3. 在 ego-item 中 新建 控制器

com.ego.item.controller.TbItemCatController

@Controller

```
public class TbItemCatController {

    @Resource

    private TbItemCatService tbItemCatServiceImpl;

    /**
     * 返回 jsonp 数据格式包含所有菜单信息
     * @param callback
     * @return
     */

    @RequestMapping("rest/itemcat/all")

    @ResponseBody

    public MappingJacksonValue showMenu(String

callback){

        MappingJacksonValue mjv = new

MappingJacksonValue(tbItemCatServiceImpl.showCatMenu(

));

        mjv.setJsonpFunction(callback);

        return mjv;

    }

}
```