# 

模版培训文档 1

一、 概述 1

二、 标签创建及使用详解 2

1、创建 2

1.1基础知识 2

1.2参数传值 3

2、使用 9

三、 API的调用 9

1使用 9

2、api开发指南 10

四：url模版映射 11

# 模版培训文档

## 概述

javashop模版制作需使用到标签，api调用，复杂的模版中还会用到url映射，以购物车为例：



我们首先用标签输出上述列表数据，对于更改商品数量的功能性操作，我们通过异步调用核心api来完成，我们以更新购物车数量为例，如更新购物车操作api地址为：/api/shop/cart/update-num.do?cartid=5&num=3&productid=305

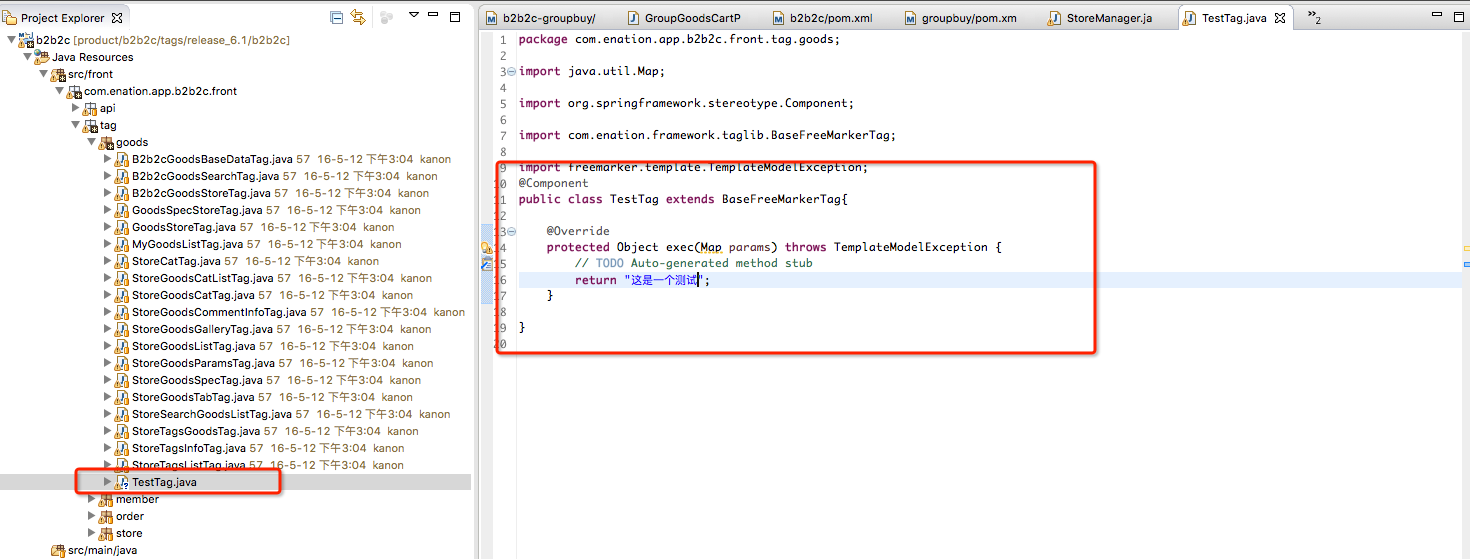
调用上述地址会返回一个json串：{"result":1,"message":null,"data":{"store":171}}

## 标签创建及使用详解

### 1、创建

#### 1.1基础知识

创建Tag类,继承com.enation.framework.taglib.BaseFreeMarkerTag然后实现这个类的exec方法，如：



@Component是spring的注解，这样此类就会自动被spring扫描,并在spring容器中创建此类的实例，此实例的beanid为testTag，就是类名首字母小写。也可以指定beanid,@Component("myTestTag")

前端调用

<!-- 创建tag -->

<#assign testTag= newTag("testTag")>

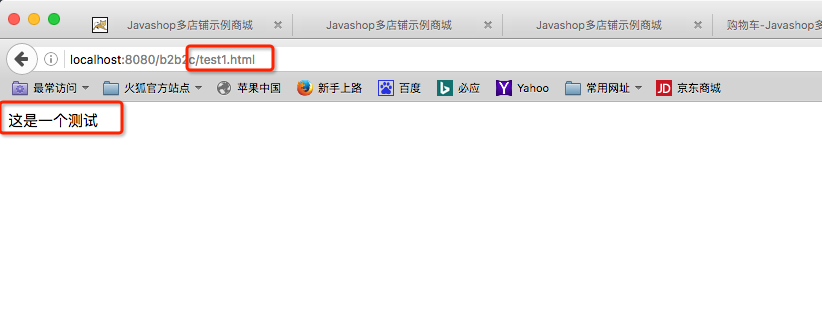
<!-- 标签的exec方法会在标签调用的时候被调用，也是就在模板中 -->

<#assign demo = testTag() >

<!-- 调用显示 -->

${demo}

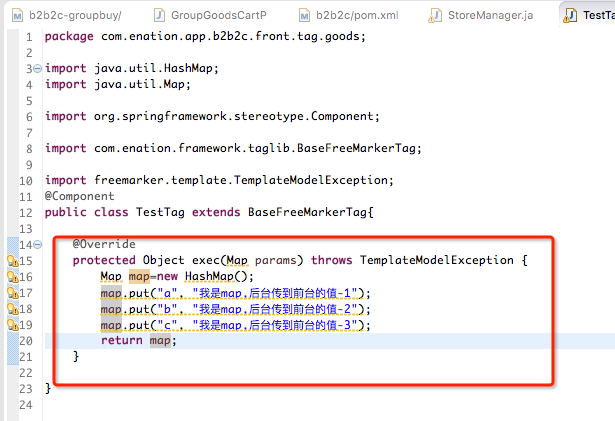
效果如图



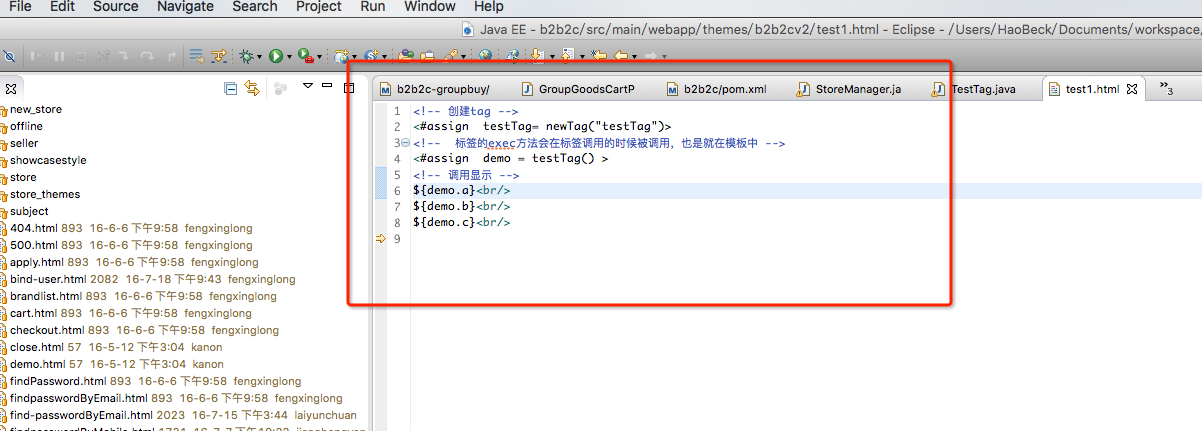
#### 1.2参数传值

1.2.1 后台传值前台 map

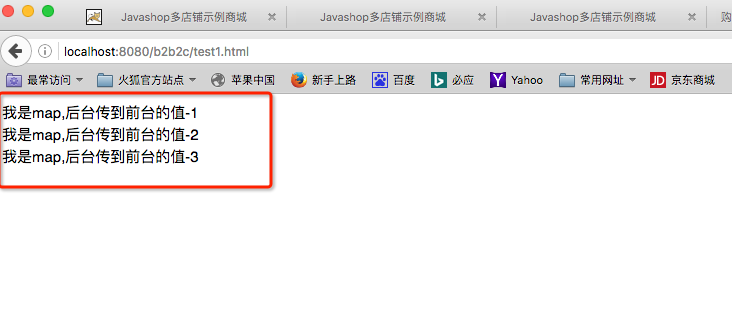
后端Tag



前端页面

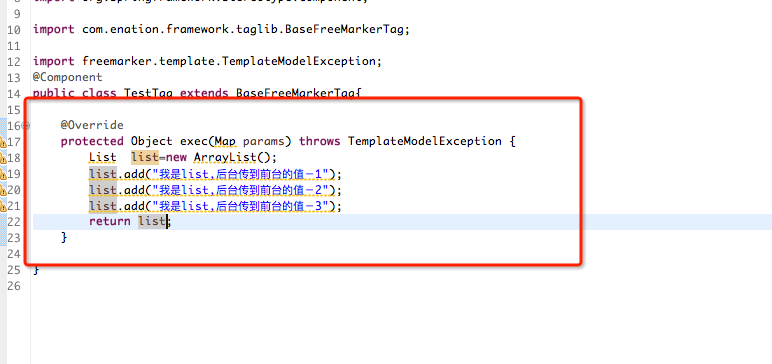


前端显示

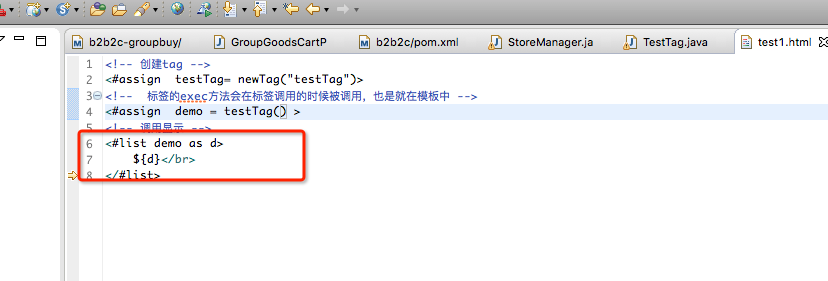


1.2.2 后台传值前台 list

后端Tag



前端页面



前端显示

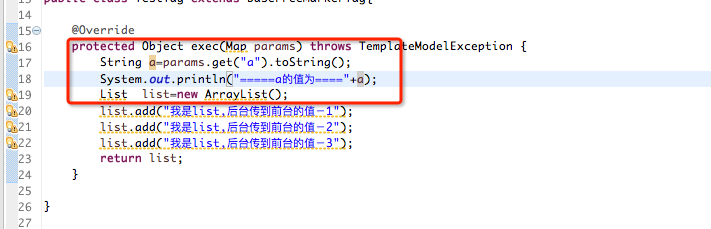


1.2.3 前台传值后台 单值

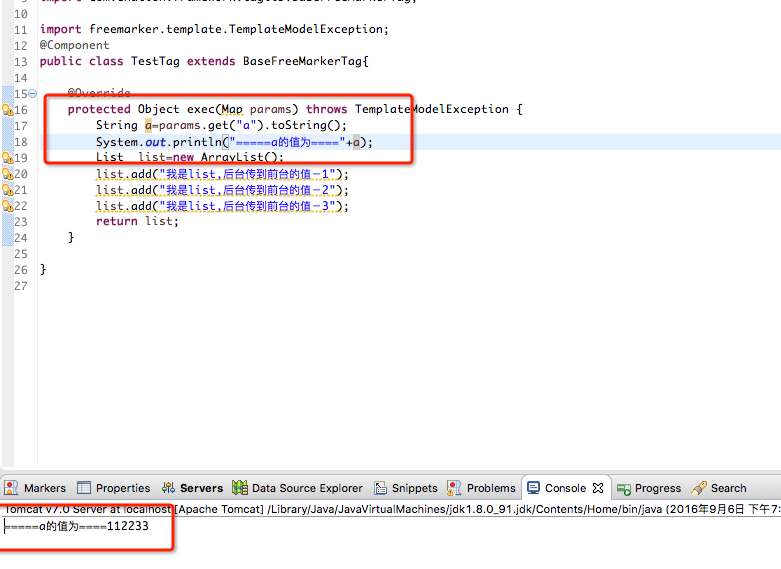
前端传值



后台接收

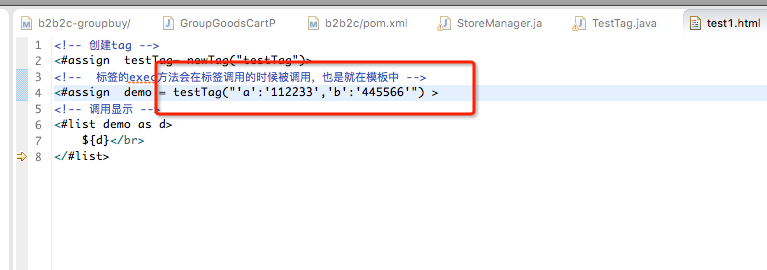


结果显示

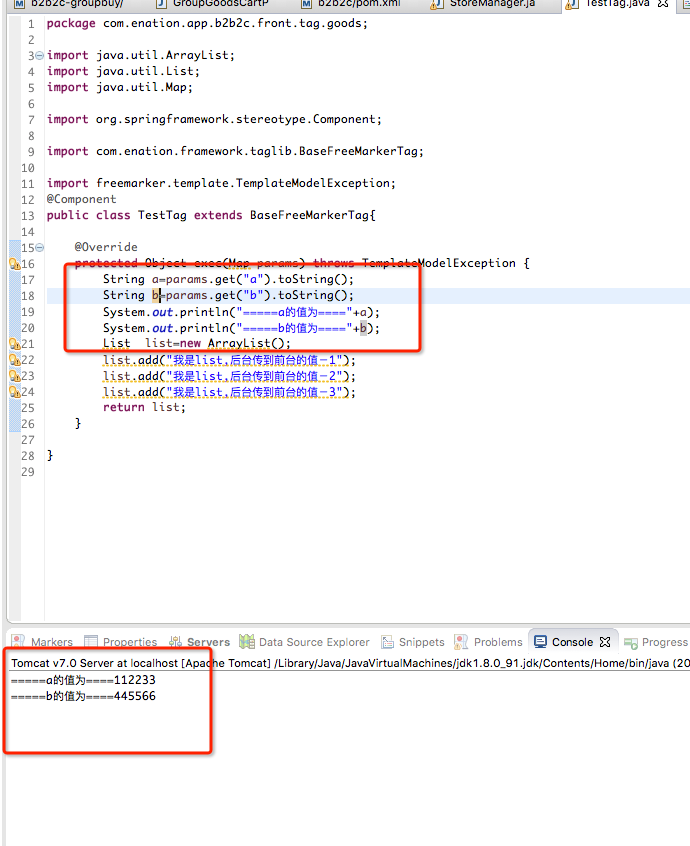


1.2.4 前台传值后台 多值

前端传值



后台接收打印



### 2、使用

 <#assign  testTag= newTag("testTag")>

assign这个关键字是Freemarker定义的，用来在模板中声明一个变量，

testTag是声明的变量名，

newTag("testTag")创建了一个测试标签，并将引用赋值给了testTag变量，

而下面这行代码是调用了刚刚创建testTag标签：

<#assign  demo = testTag()>

调用这个标签会得到值，上述代码把他赋值给demo变量。这 样我们就可以通过#list 指令来循环这个列表了。

小提示

　　在页面中可以直接获取在地址栏和form中传递的参数哦，如：/test.html?name=wf

　　那么在页面中直接通过${name}就可以输出地址栏中定义的name参数。同样的地，form表单中的参数通过上述方法可以一直获取、输出。

## API的调用

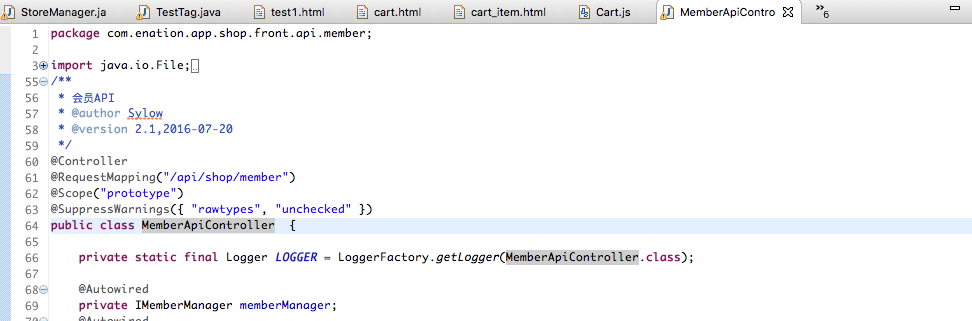
### 1使用

我们还以上述购物车的删除操作为例，调用购物车删除api代码如下：



上述的api是通过访问一个url:/api/shop/cart/delete.do来调用的，调用结果是json格式，如：{"result":1,"message":"删除购物项成功","data":null}

### 2、api开发指南



@Controller标注此类为控制器。

@RequestMapping("/api/shop/member")用来处理请求地址映射的注解决定了此api的访问路径

@Scope("prototype")生命bean的生命周期为prototype,这是为了多线程时的数据安全

像上述示例的api，需要通过如下地址访问：/api/shop/cart/delete.do这样就会调用delete方法，这个api的返回结果会是：{"result":1,"message":"删除购物项成功","data":null}

返回的json串的规范对于我们意义重大，我们规定：

　 1、json串中必须含有结果值，我们规定用result来表示，1表示成功，0表示失败

2、json串的名称部分必须用双引号，字串值也必须用双引号

javashop提供了常用方法帮助大家生成json串，这些方法有：

**基于JsonResultUtil中的方法**

1、getSuccessJson()方法返回成功结果的字符串

JsonResultUtil.getSuccessJson(“成功”);

返回

{"result":1,"message":"成功","data":null}

2、getErrorJson()方法返回失败结果的字符串

JsonResultUtil.getSuccessJson(“失败”);

返回

{"result":0,"message":"失败","data":null}

3、getObjectJson() 方法返回json格式的对象

JsonResultUtil.getObjectJson(data);

返回

{"result":1,"message":"","data":””"}

。。。。。。

## 四：url模版映射

可能会存在一些特殊的模版，比如商品详情页，他的访问地址一般为goods-1.html，其中的1表示这个商品的id,通过这个id我们的商品详细标签才能由数据库中查询此商品的数据，我们有很多的商品不可能写很多的模版来展示商品详情,那么此时就用到了模版映射

通过正则我们建立如下映射

/goods-(\d+).html -> /goods.html那么访问所有的/goods-{数字}.html 这样的地址都会解析goods.html这个模板了。