# SpringMVC框架技术

版本：3.6.0

## 第5章 文件上传和json数据交互



# 本章教学目标

- 了解RESTful概述；
- 了解RESTful应用；
- 理解json数据交互概述；
- 掌握文件上传；
- 掌握json数据交互；

# 本章教学内容

Neuedu

| 节 | 知识点 | 掌握程度 | 难易程度 | 教学形式 | 对应在线微课 |
|---|---|---|---|---|---|
| 文件上传 | 导入jar包 | 掌握 | | 线下 | 导入jar包 |
| | 创建虚拟目录 | 掌握 | | 线下 | 创建虚拟目录 |
| | 配置解析器 | 掌握 | | 线下 | 配置解析器 |
| | 编写controller代码 | 掌握 | 难 | 线下 | 编写controller代码 |
| | 编写页面代码 | 掌握 | | 线下 | 编写页面代码 |
| json数据交互 | json数据交互概述 | 理解 | | 线下 | json数据交互概述 |
| | 环境搭建 | 掌握 | | 线下 | 环境搭建 |
| | 数据交互 | 掌握 | | 线下 | 数据交互 |
| **RESTful支持** | **RESTful概述** | **了解** | | **线上** | **RESTful概述** |
| | **RESTful应用** | **了解** | | **线上** | **RESTful应用** |
| 本章实战项目任务实现 | 实战项目任务实现 | 掌握 | | 线下 | 实战项目任务实现 |

# 本章实战项目任务

- 通过本章内容的学习，完成《跨境电商系统》商品录入功能
  - 包括商品基本信息录入和多图片上传
  - 成功，在控制台和jsp页面中显示商品信息

- 运行效果如图：



```
saveProduct start...
skuCd=GM00100
typeCd=PUB,price=7000
typeCd=PRO,price=6600
typeCd=VIP,price=6000
imageName=
imageName=
originalFilename=iphon
originalFilename=iphon
imageId=1000
imageName=bc044a4b-afb
imageUri=/upload/bc04
imageId=1001
imageName=9667309f-df
imageUri=/upload/9667
```

CONTENTS 目录



# 文件上传

Neuedu

- 可通过form表单上传文件对象，通过Spring配置解析，存储图片文件
- 文件上传实现步骤：
  - 导入jar包
  - 创建虚拟目录
  - 配置解析器
  - 编写controller代码
  - 编写页面代码

# 文件上传

- 示例：
  - 使用springmvc完成商品信息修改
  - 添加图片

修改商品信息：

| | |
|---|---|
| 商品名称 | 笔记本电脑 |
| 商品价格 | 8000.0 |
| 商品生产日期 | 2018-03-06 16:39:14 |
| 商品图片 |  选择文件 未选择任何文件 |
| 商品简介 | 笔记本电脑质量好 价格低 |
| 提交 | |

- 全部代码参见：ch05-springmvc01工程

# 导入jar包

- 在pom. xml中配置依赖包

```
 <!-- 文件上传 start -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.2.2</version>
</dependency>
<dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.4</version>
</dependency>
<!-- 文件上传 end -->
```

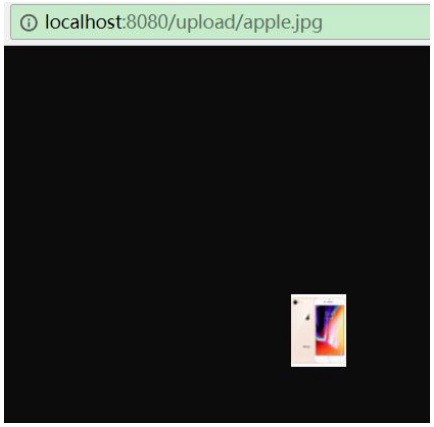
# 配置虚拟目录

Neuedu

- 在tomcat上配置图片虚拟目录，在tomcat下conf/server.xml中添加

```
<Context docBase="D:\images" path="/upload" reloadable="true" />
```

- docbase指定文件存储路径(D:\images本地要先建立)
- 配完启动tomcat，可通过localhost:8080/upload/文件名，访问存储的文件



# 配置解析器

- 在springmvc.xml中配置

```
<!-- 文件上传 -->
<bean id="multipartResolver"
    class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <!-- 设置上传文件的最大尺寸为5MB -->
    <property name="maxUploadSize">
        <value>5242880</value>
    </property>
</bean>
```

# 编写controller代码

Neuedu

- 编写如下代码

```
@RequestMapping("/editItemsSubmit")
public String editItemsSubmit(Integer id,ItemsCustom itemsCustom,MultipartFile items_pic) throws Exception {

//原始名称
String originalFilename = items_pic.getOriginalFilename();
if(items_pic !=null && originalFilename !=null && originalFilename.length()>0) {
    //存储图片的物理路径
    String pic_path = "D:\\images\\";
    //新的图片名称
    String newFileName = UUID.randomUUID() + originalFilename.substring(originalFilename.lastIndexOf("."));
    //新图片
    File newFile = new File(pic_path + newFileName);

    //将内存的数据写入磁盘
    items_pic.transferTo(newFile);

    //将新图片名称写到itemsCustom中
    itemsCustom.setPic(newFileName);
}
```

# 编写页面代码

Neuedu

- form标签中添加enctype=“multipart/form-data”

```
<form id="itemForm" enctype="multipart/form-data" action="${pageContext.request.contextPath }
```

- Img标签的src修改如下Img标签的src修改如下

```
<tr>
    <td>商品图片</td>
    <td>

     <c:if test="${itemsCustom.pic != null}">
            <img src="/upload/${itemsCustom.pic}" width=100 height=100/>
            <br/>
        </c:if>
        <input type="file"  name="items_pic"/>
    </td>
</tr>
```

# 课堂练习（5分钟）

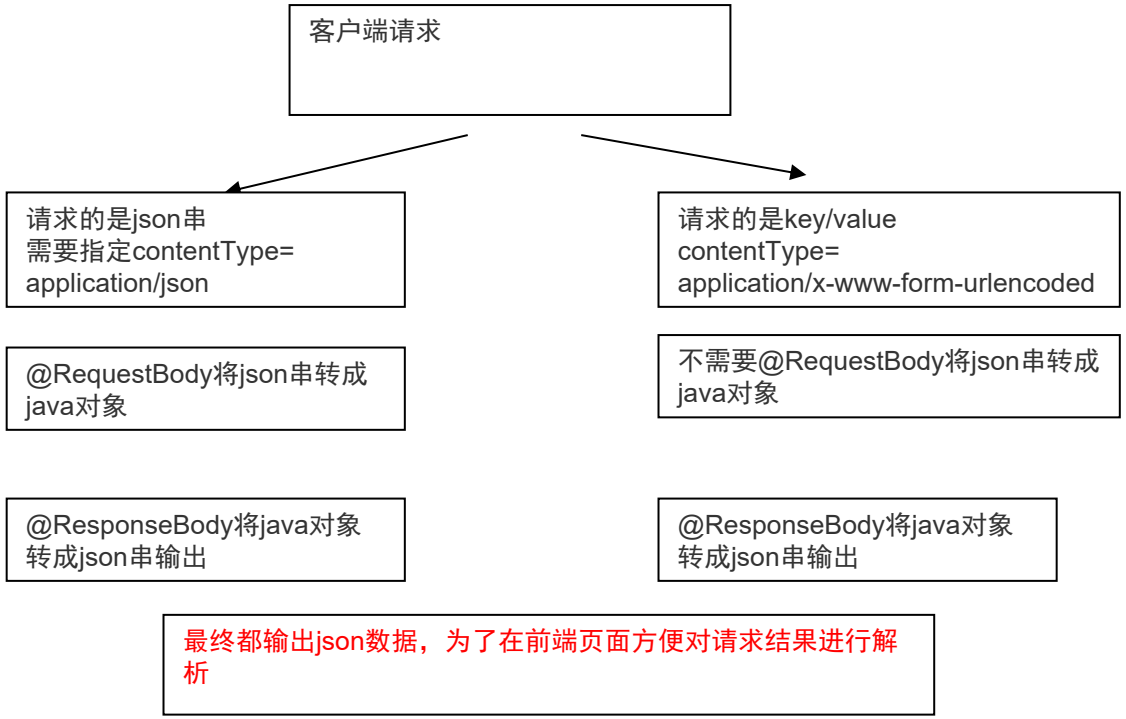
- 简述springmvc图片上传的方法步骤

CONTENTS 目录



# json数据交互

- json交互概述
  - json数据格式在接口调用中、html页面中较常用，json格式比较简单，解析还比较方便。
  - json数据交互思路



# json数据交互

- json 数据交互步骤
  - 导入jar包
  - 配置json转换器
  - 编写页面代码
  - 编写controller代码

- 全部代码参见：ch05-springmvc02工程

# 环境搭建

Neuedu

- 导入依赖的jar包

```
<dependency>
    <groupId>org.codehaus.jackson</groupId>
    <artifactId>jackson-core-asl</artifactId>
    <version>1.9.13</version>
</dependency>
<dependency>
    <groupId>org.codehaus.jackson</groupId>
    <artifactId>jackson-mapper-asl</artifactId>
    <version>1.9.13</version>
</dependency>
```

- 导入jquery文件

# 环境搭建

Neuedu

- 配置json转换器
  - 在springmvc.xml中配置

```
<bean
class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAd
apter">
    <property name="messageConverters">
    <list>
    <bean
class="org.springframework.http.converter.json.MappingJacksonHttpMessageConverter"></
bean>
    </list>
    </property>
</bean>
```

如果使用**<mvc:annotation-driven />** 则不用定义上边的内容。（推荐使用该方法）

# json数据交互

- 输入json串，输出是json串
  - jsp 页面

```
function requestJson(){
    $.ajax({
        type:'post',
        url:'${pageContext.request.contextPath}/requestJson.action',
        contentType:'application/json;charset=utf-8',
        data:'{"name":"手机","price":888}',
        success:function(data){
            alert(data);
            alert(data.price);
        }

    });

}
```

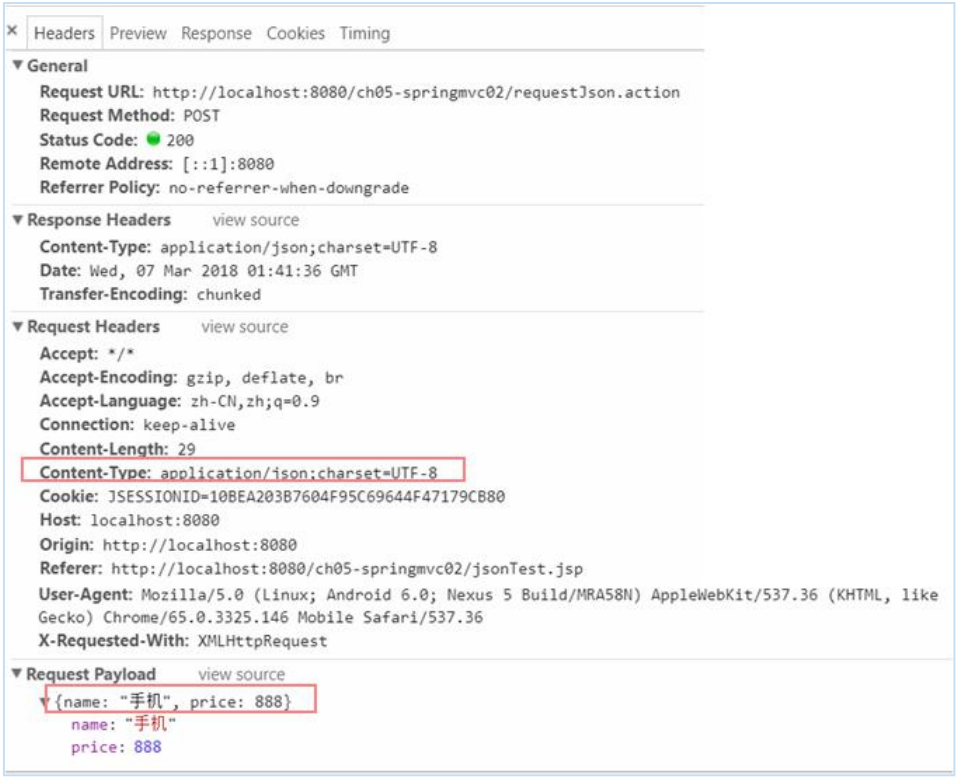
# json数据交互

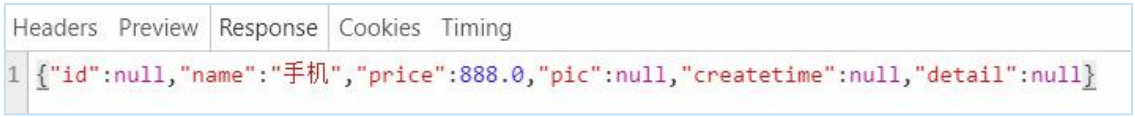
- 输入json串，输出是json串
  - controller

```
//请求json串(商品信息)，输出json(商品信息)
//@RequestBody将请求的商品信息的json串转成itemsCustom对象
//@ResponseBody将itemsCustom转成json输出
@RequestMapping("/requestJson")
public @ResponseBody ItemsCustom requestJson(@RequestBody ItemsCustom itemsCustom){
    //@ResponseBody将itemsCustom转成json输出
    return itemsCustom;
}
```

# json数据交互

- 输入json串，输出是json串
  - 浏览器访问请求后，查看http请求和响应



# json数据交互

- 输入key/value，输出是json串
  - jsp 页面

```
function responseJson() {
    $.ajax({
        type:'post',
        url:'${pageContext.request.contextPath}/responseJson.action',
        data:'name=手机&price=8888',
        success:function(data){
            alert(data.name);
        }
    });


}
```

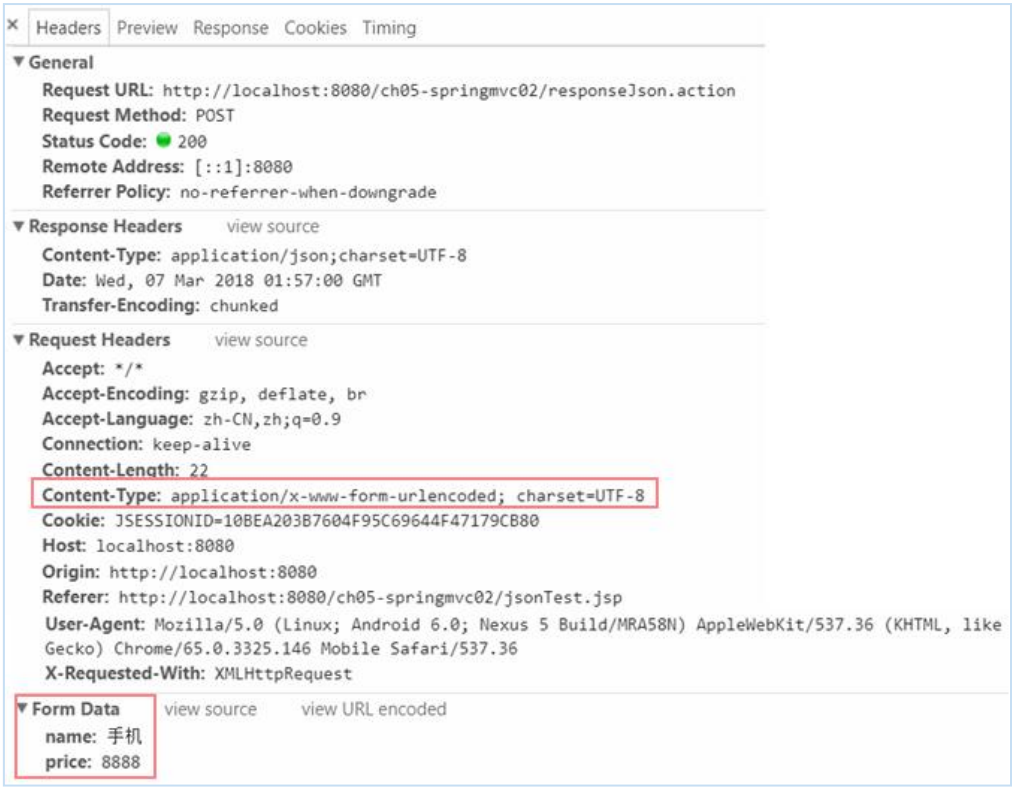
# json数据交互

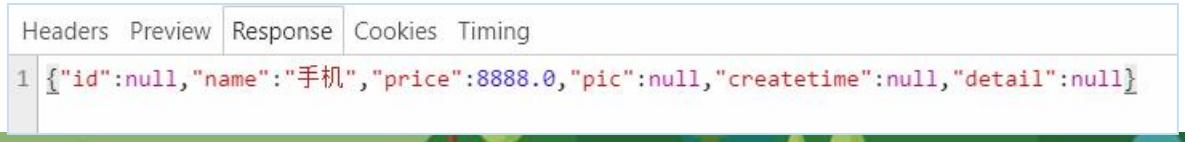
- 输入key/value，输出是json串
  - controller

```
//请求key/value，输出json
@RequestMapping("/responseJson")
public @ResponseBody ItemsCustom responseJson(ItemsCustom itemsCustom){
    //@ResponseBody将itemsCustom转成json输出
    return itemsCustom;
}
```

# json数据交互

- 输入json串，输出是json串
  - 浏览器访问请求后，查看http请求和响应



# 课堂练习（15分钟）

- 1、简述@ResponseRequest和@RequestResponse的作用
- 2、独立完成教学中的案例

CONTENTS

# 目录

# RESTful概述

- 什么是REST
  - REST 即表述性状态传递（英文：Representational State Transfer，简称REST）
  - 是Roy Fielding博士在2000年发表的博士论文中提出来的一种软件架构风格。
  - 是一种针对网络应用的设计和开发方式，可以降低开发的复杂性，提高系统的可伸缩性。
- RESTful架构
  - REST 指的是一组架构约束条件和原则。" 如果一个架构符合REST的约束条件和原则，我们就称它为RESTful架构。
  - RESTful架构，就是目前最流行的一种互联网软件架构。它结构清晰、符合标准、易于理解、扩展方便，所以得到越来越多网站的采用。
  - 每一个URI代表一种资源
  - 客户端和服务器之间，传递这种资源的某种表述
  - 客户端通过标准的HTTP方法，对服务器端资源进行操作

# RESTful概述

Neuedu

- 对url进行规范，写RESTful格式的url
  - 非REST的url：http://localhost:8080/SpringMVC/product/findItemsById.action?id=3
  - REST的url风格：http://localhost:8080/SpringMVC/product/findItemsById/3
  - 特点：url简洁，将参数通过url传到服务端
- http的方法规范
  - 使用标准的http方法（POST、GET、PUT、DELETE）
  - 不管是删除、添加、更新。使用url是一致的，如果进行删除，需要设置http的方法为delete，同理添加为post。
  - 后台controller方法：判断http方法，如果是delete执行删除，如果是post执行添加。
- 对http的contentType规范
  - 请求时指定contentType，如果是json数据，设置成json格式的type。

# RESTful应用

- 示例：
  - 根据id查询商品信息，返回json数据。


  - 全部代码参见：ch05-springmvc03工程

# RESTful应用

- 配置REST方法的前端控制器
  - 在web.xml中增加如下配置

```
<servlet>
  <servlet-name>springmvc_rest</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:springmvc.xml</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>springmvc_rest</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
```

  - 全部代码参见：ch05-springmvc03工程

# RESTful应用

- URL 模板模式映射
  - @RequestMapping(value="/findItemsById/{id}")：{×××}占位符，请求的URL可以是“/findItemsById/1”或“/findItemsById/2”，通过在方法中使用@PathVariable获取{×××}中的×××变量。
  - @PathVariable用于将请求URL中的模板变量映射到功能处理方法的参数上。
  - 如果RequestMapping中表示为"/findItemsById/{id}", id和形参名称一致，@PathVariable不用指定名称。

# RESTful应用

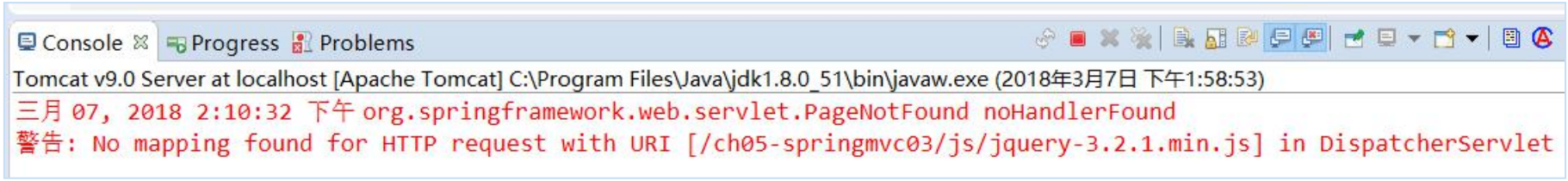
- URL 模板模式映射
  - controller定义方法，进行url映射使用REST风格的url

```
//根据id查询商品信息，返回json数据
@RequestMapping("/itemsView/{id}")
//public @ResponseBody ItemsCustom findItemsById(@PathVariable Integer id) throws Exception {
public @ResponseBody ItemsCustom findItemsById(@PathVariable("id") Integer id) throws Exception {
    ItemsCustom itemsCustom = itemsService.findItemsById(id);
    return itemsCustom;
}
```

  - 全部代码参见：ch05-springmvc03工程

# RESTful应用

- 对静态资源的解析
  - 配置前端控制器的url-parttern中指定/，对静态资源的解析出现问题：





  - 在springmvc.xml中添加静态资源解析方法。

```
<!-- 静态资源解析 包括js、css、img... -->
<mvc:resources location="/js/" mapping="/js/**"/>
<mvc:resources location="/img/" mapping="/img/**"/>
```

  - 全部代码参见：ch05-springmvc03工程

# 课堂练习（20分钟）

- 1、简述REST及RESTful架构
- 2、RESTful应用和非RESTful应用，在开发中有哪些不同？
- 3、理解并独立完成教学中的案例

CONTENTS 目录

# 本章实战项目任务实现

- 通过本章内容的学习，完成《跨境电商系统》商品录入功能
  - 包括商品基本信息录入和多图片上传
  - 成功，在控制台和jsp页面中显示商品信息
- 运行效果如图：



```
saveProduct start...
skuCd=GM00100
typeCd=PUB,price=7000
typeCd=PRO,price=6600
typeCd=VIP,price=6000
imageName=
imageName=
originalFilename=iphon
originalFilename=iphon
imageId=1000
imageName=bc044a4b-afb
imageUri=/upload/bc04
imageId=1001
imageName=9667309f-df
imageUri=/upload/9667
```

# 本章重点总结

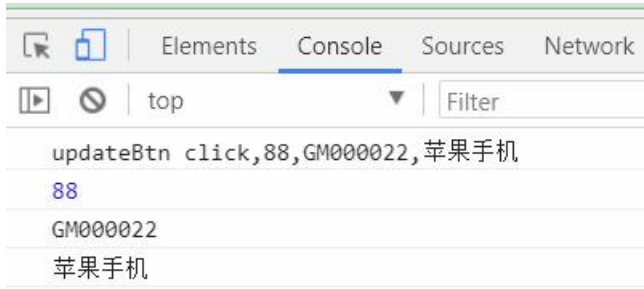
- 了解RESTful概述；
- 了解RESTful应用；
- 理解json数据交互概述；
- 掌握文件上传；
- 掌握掌握json数据交互；

# 课后作业【必做任务】

- 1、完成《跨境电商系统》商品基本信息的修改
  - 假设修改的数据已经查找并显示在页面中
  - 使用json数据交互完成商品基本信息修改
  - 点击更新按钮后，商品最新信息显示在浏览器的控制台

- 修改前和修改后的运行效果如图：





# 课后作业【选做任务】

- 完成《跨境电商系统》商品查询
  - 根据商品id查询，json格式返回商品信息
  - RESTful格式规范url，返回商品信息在页面展示

- 运行效果如图：


# 课后作业【线上任务】

- 线上任务
  - 安排学员线上学习任务（安排学员到睿道实训平台进行复习和预习的任务，主要是进行微课的学习）