

json数据的传输及上传文件

文件的上传

思路：

文件一般会建立一个专门的服务器去存储文件数据，把文件的名字再文件存储到文件服务器后存储到数据库，方便日后获取。

配置虚拟目录

1. 可以通过eclipse配置：

□

2. 通过tomcat配置：在E:\Tomcat\conf\Catalina\localhost路径下新建一个XML文件，注意：XML文件的名字也就是项目部署成功后的web根路径.假如定义这test.xml,那么后面的访问路径就是http://ip:port/test,文件内容如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Context docBase="D:\project\XXX" reloadable="true" debug="0" path="/test"/>
```

path:网络访问名(path="/test"在这里不写也可以,此时的path实际上是由xml文件的名字决定的)

docBase:项目实际路径

reloadable:指定当class文件变化的时候，是否重新加载

debug:是否是调试模式

加入jar包

CommonsMultipartResolver解析器依赖commons-fileupload和commons-io，加入如下jar包：

□

springMVC.xml中配置文件解析器

```
<!-- 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>
```

```
/商品修改提交
@RequestMapping("/editItemSubmit")
public String editItemSubmit(Items items, MultipartFile pictureFile)throws Exception{

    //原始文件名称
    String pictureFile_name = pictureFile.getOriginalFilename();
    //新文件名称
    String newFileName = UUID.randomUUID().toString()
    +pictureFile_name.substring(pictureFile_name.lastIndexOf("."));

    //上传图片
    File uploadPic = new java.io.File("F:/develop/upload/temp/"+newFileName);

    if(!uploadPic.exists()){
        uploadPic.mkdirs();
    }
    //向磁盘写文件
    pictureFile.transferTo(uploadPic);
```

页面的定义:

```
<!--form添加enctype="multipart/form-data": -->
<form id="itemForm"
  action="${pageContext.request.contextPath }/item/editItemSubmit.action"
  method="post" enctype="multipart/form-data">
<input type="hidden" name="pic" value="${item.pic }" />
  //file的名称与controller形参一致:
  <tr>
    <td>商品图片</td>
    <td><c:if test="${item.pic !=null}">
      
      <br />
    </c:if> <input type="file" name="pictureFile" /></td>
  </tr>
</form>
```

json数据交互

- @RequestBody 作用: @RequestBody注解用于读取http请求的内容(字符串), 通过springmvc提供的HttpMessageConverter接口将读到的内容转换为json、xml等格式的数据并绑定到controller方法的参数上。
- @ResponseBody 作用: 该注解用于将Controller的方法返回的对象, 通过HttpMessageConverter接口转换为指定格式的数据如: json,xml等, 通过Response响应给客户端

环境准备

Springmvc默认用MappingJacksonHttpMessageConverter对json数据进行转换, 需要加入jackson的包, 如下:

□

6.配置json转换器

在注解适配器中加入messageConverters

```
<!--注解适配器 -->
<bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter">
  <property name="messageConverters">
    <list>
      <bean class="org.springframework.http.converter.json.MappingJacksonHttpMessageConverter"></bean>
    </list>
  </property>
</bean>
```

注意: 如果使用<mvc:annotation-driven /> 则不用定义上边的内容。所以推荐使用这个注解。

- controller中定义

```
// 商品修改提交json信息, 响应json信息
@RequestMapping("/editItemSubmit_RequestJson")
public @ResponseBody Items editItemSubmit_RequestJson(@RequestBody Items items) throws Exception {
  System.out.println(items);
  //itemService.saveItem(items);
  return items;
}
```

- 页面定义

```
//请求json响应json
function request_json(){
```

```
$.ajax({
  type: "post",
  url: "${pageContext.request.contextPath }/item/editItemSubmit_RequestJson.action",
  contentType: "application/json;charset=utf-8",
  data: '{"name":"测试商品","price":99.9}',
  success: function(data){
    alert(data);
  }
});
}
```

实际开发中常用第二种方法，请求key/value数据，响应json结果，方便客户端对结果进行解析。

完