# Ssm整合Ueditor

1.新建web project，并配置好spring mvc环境。

2.下载百度编辑器UEditor,本文以（ueditor1\_4\_3\_3-utf8-jsp）版本为例。

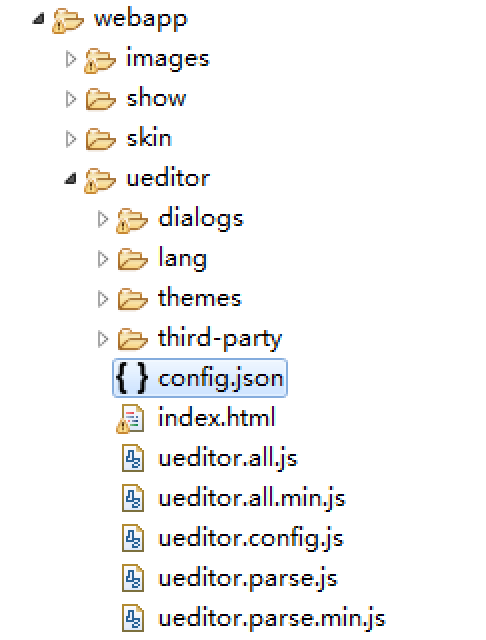


1. 在工程WebRoot下创建目录ueditor,并把下载得到的文件全部（jsp目录除外）复制过来。并在工程中添加依懒

|  |
| --- |
| <dependency>  <groupId>com.gitee.qdbp.thirdparty</groupId>  <artifactId>ueditor</artifactId>  <version>1.4.3.3</version>  </dependency>  <dependency>  <groupId>org.json</groupId>  <artifactId>json</artifactId>  <version>20160810</version>  </dependency> |

4.把下载得到jsp目录下config.json复制到ueditor目录下。完成3-4步如下图：

如图：



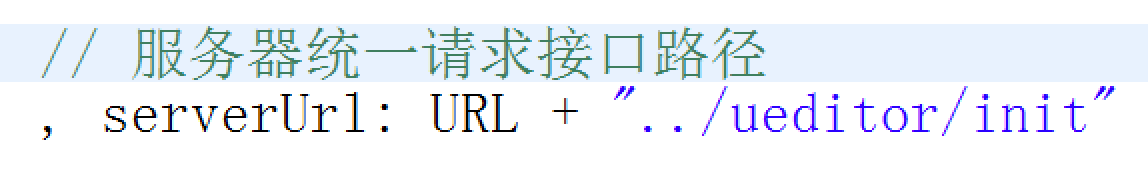
5.配置后台

创建UEditorController，代码为下载得到jsp目录下controller.jsp里的代码，则改写成如下代码：

|  |
| --- |
| **import** java.io.IOException;  **import** java.io.PrintWriter;  **import** java.nio.file.Files;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** org.apache.log4j.Logger;  **import** org.springframework.stereotype.Controller;  **import** org.springframework.web.bind.annotation.RequestMapping;  **import** com.baidu.ueditor.ActionEnter;  @Controller  **public** **class** UEditorController {  **private** Logger logger = Logger.*getLogger*(UEditorController.**class**);  /\*\*  \* 初始化百度编辑器，可传入其他参数生成不同的编辑器,总入口  \*  \* **@param** response  \* **@param** request  \*/  @RequestMapping("/ueditor/init")  **public** **void** initUeditor(HttpServletResponse response, HttpServletRequest request) {  response.setContentType("application/json");  // 配置路径，首先获取static根目录绝对路径  String rootPath = request.getSession().getServletContext().getRealPath("/");  // 将config.json放到与ueditor.config.js同一级的目录下。将ueditor所有文件放入到wapapp-ueditor下  // 设置获取服务端配置文件地址修正路径，此路径同时作用于文件上传  PrintWriter writer = **null**;  **try** {  String exec = **new** ActionEnter(request, rootPath).exec();  writer = response.getWriter();  writer.write(exec);  writer.flush();  } **catch** (IOException e) {  logger.error("百度编辑器初始化错误！", e);  } **finally** {  **if** (writer != **null**) {  writer.close();  }  }  }  } |

6.修改配置文件

从上面的controller可见，ueditor初化的请求足路径为ueditor/init，所以接下来修改前台ueditor.config.js文件，修改代码：



注意这里的相对路径

7.测试1

在地址栏输入：项目名/ueditor/init?action=config,出现类似下图所示内容，则配置成功，否则，即为失败。



1. 解决图片无法上传(ueditor与multipartResolver配置冲突)

访问，发现界面已经可以正常访问，但无法上传文件，在配置成功ueditor后，如果使用表单上传文件或者使用ajax异步传送文件返回url等操作，会上传失败，下面提出解决方案：

1.在文件中建立该类UEditorCommonsMultipartResolver

2.将url.contains("")中的值改为上文中我们创建的ueditor config读取类的RequestMapping的值

代表当遇到url为RequestMapping的文件则不拦截到multipartResolver中

|  |
| --- |
| **import** javax.servlet.http.HttpServletRequest;  **import** org.springframework.web.multipart.commons.CommonsMultipartResolver;  **public** **class** UEditorCommonsMultipartResolver **extends** CommonsMultipartResolver {  @Override  **public** **boolean** isMultipart(HttpServletRequest request) {  String url = request.getRequestURI();  **if** (url != **null** && url.contains("/ueditor/init")) {  **return** **false**;  } **else** {  **return** **super**.isMultipart(request);  }  }  } |

9.配置bean

在springMVC.xml中配置bean，这个class路径就是上文中写的UEditorCommonsMultipartResolver类的路径，代码如下：

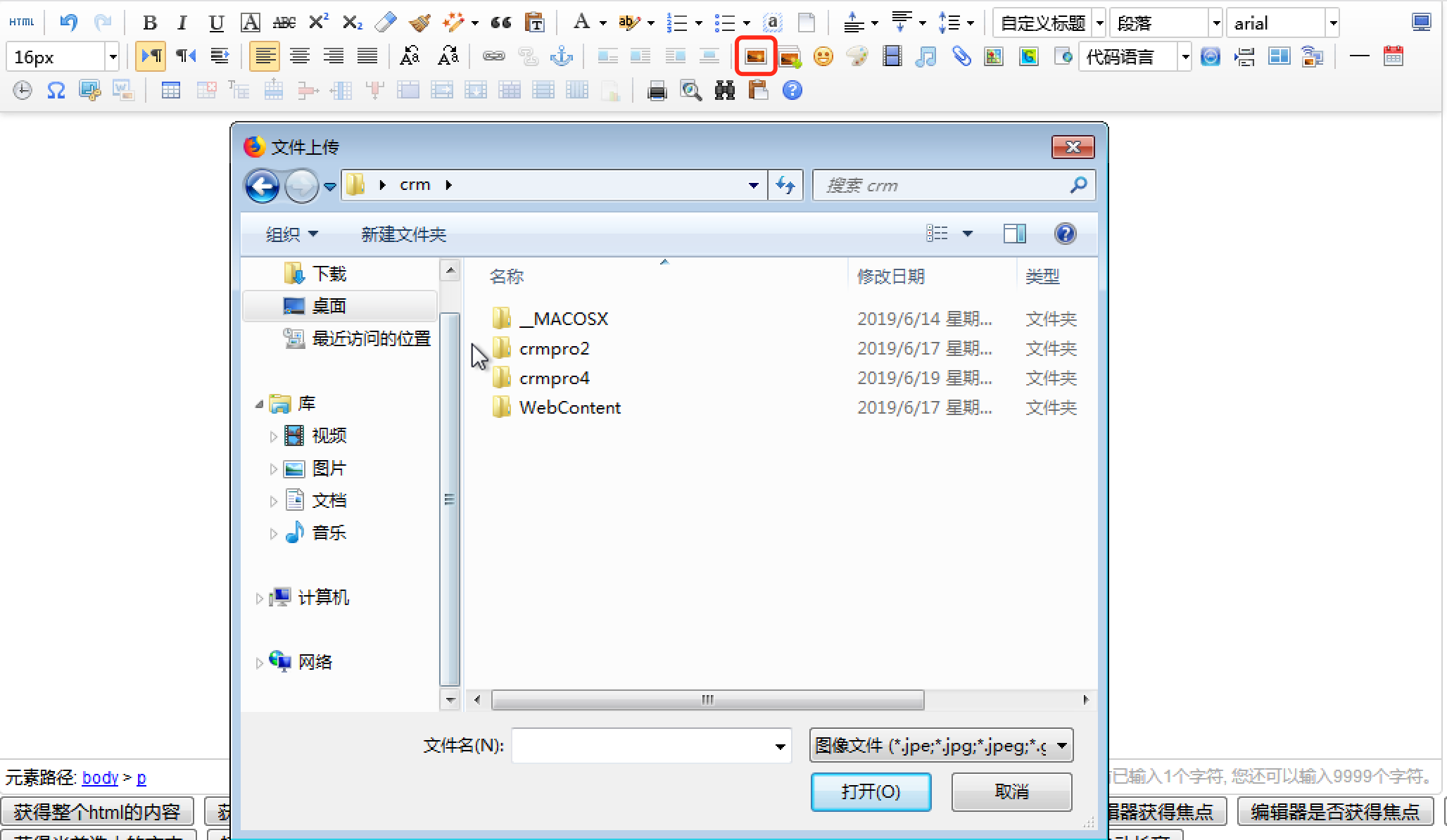
|  |
| --- |
| <bean id=*"multipartResolver"* class=*"com.offcn.crmpro.utils.UEditorCommonsMultipartResolver"*>  <!-- 上传文件大小上限，单位为字节（10MB） -->  <property name=*"maxUploadSize"*>  <value>10485760</value>  </property>  <!-- 请求的编码格式，必须和jSP的pageEncoding属性一致，以便正确读取表单的内容，默认为ISO-8859-1 -->  <property name=*"defaultEncoding"*>  <value>UTF-8</value>  </property>  </bean> |

这个时候就可以成功的将ueditor与multipartResolver分离了。

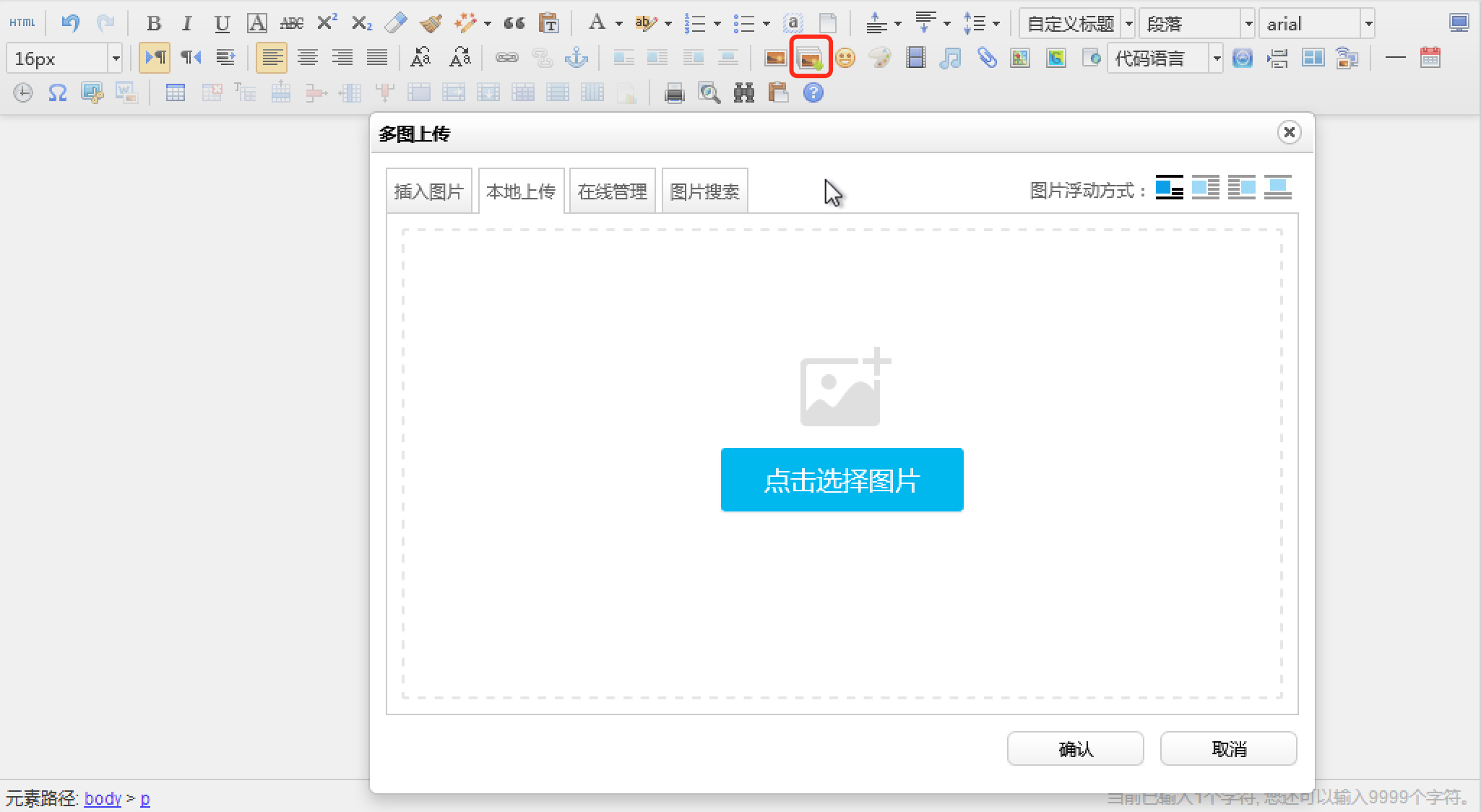
10.测试图片上传（**清除缓存**）

访问:http://localhost:8080/crmpro/ueditor/index.html

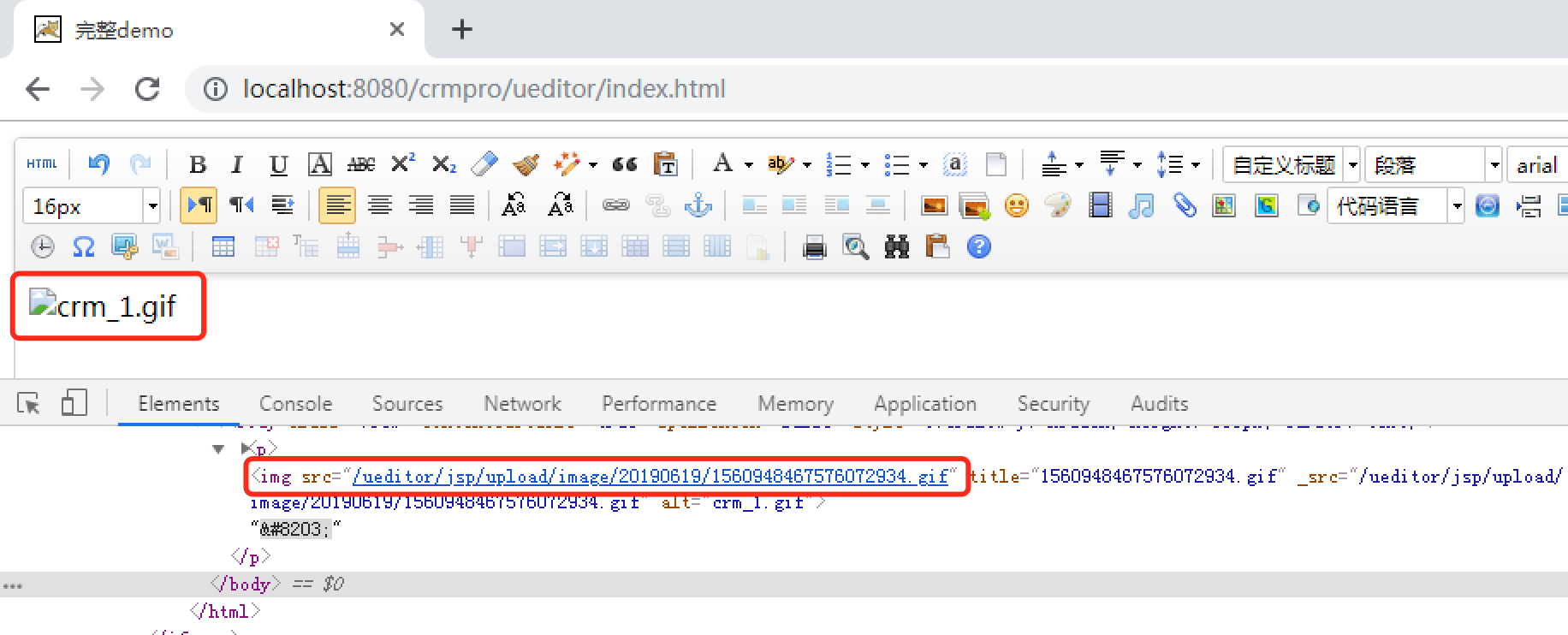
点击工具栏中单图片上传,如出现下图,则说明可以上传单图了



点击工具栏中多图片上传,如出现下图,则说明可以上传多图了



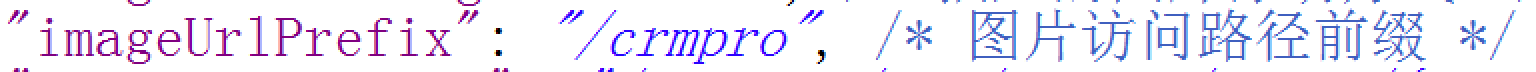
选择图片后发现上传成功了,但是无法显示出来,原因是回示的路径不正确



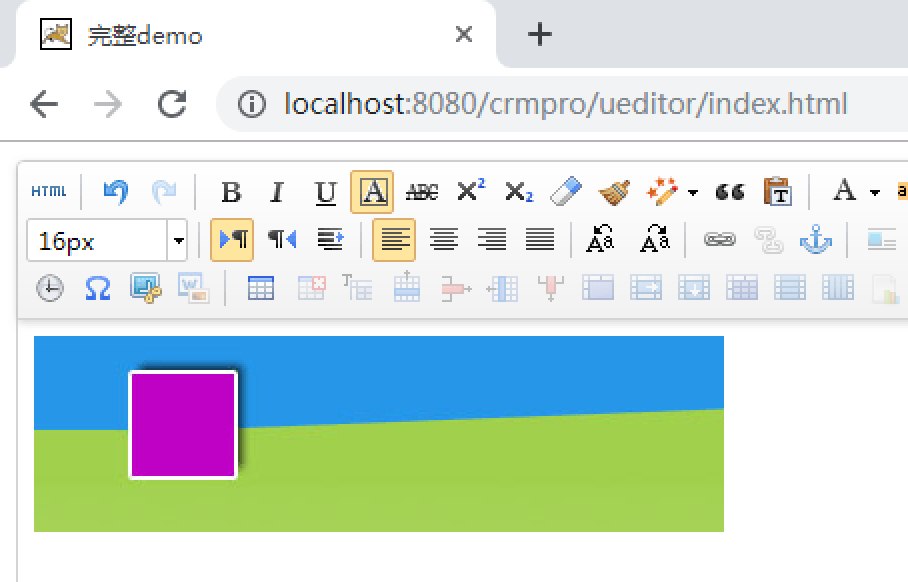
修改config.json



图片,视频,文件访问路径前缀统一加上项目名



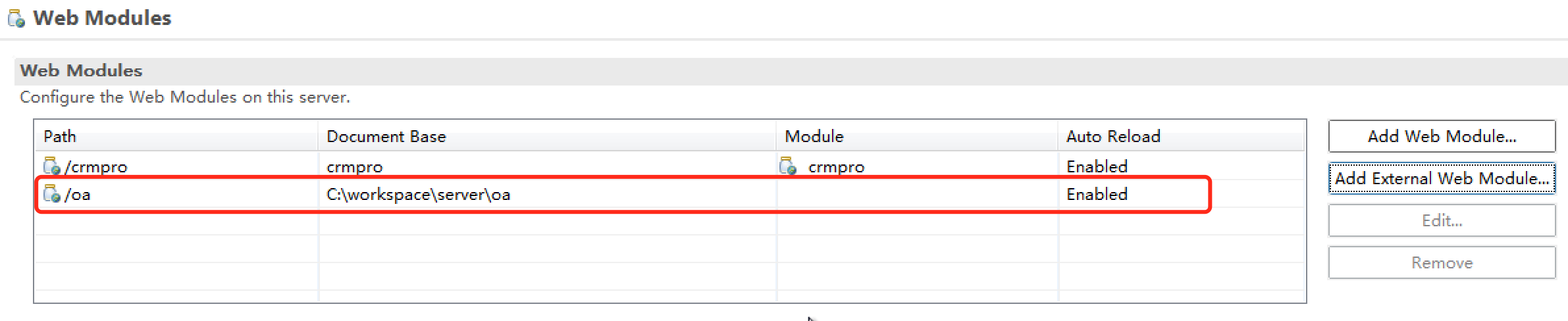
再一次上传,回显正常



# 上传图片到挂截路径

（解决tomcat重启后图片消失问题）

新建挂截路径



修改config.json

添加localSavePathPrefix "localSavePathPrefix":"C:/workspace/server/oa",

所有访问路径统一添加前缀，如

"imageUrlPrefix": *"/oa"*, /\* 图片访问路径前缀 \*/

ConfigManager类中的getConfig(inttype)方法，添 加了一句：

rootPath = jsonConfig.getString("localSavePathPrefix");