

Part 1: 下载富文本编辑器源码及JSP代码

官网下载传送门: [UEditor](#) 下载链接

下载版本如图所示:

 [首页](#) [下载](#) [UE演示](#) NEW [UM演示](#) [插件](#) NEW [文档](#) [API](#) [论坛](#) [百度脑图](#) [百度 H5](#)

common-fileupload.jar 包存在 ddos 漏洞, 请尽快升级到最新的版本, 直达链接: [Apache fileupload jar](#)
UEditor 所提供的所有后端代码都仅为 DEMO 作用, 切不可直接使用到生产环境中, 目前已知 php 的代码会存在 ssrf 的安全漏洞。修复方式: 使用最新的 Uploader <https://github.com/fex-team/ueditor/blob/dev-1.5.0/php/Uploader.class.php>

开发版

Mini版

最新版本

如果您是百度产品线用户, 我们可以提供定制服务, 欢迎加hi群: 1446488 [更多历史版本](#)

[1.4.3.3 完整源码]	完整源码 源码版如何编译?	2016-05-26
[1.4.3.3 PHP 版本]	UTF-8版 GBK版	2016-05-26
[1.4.3.3 Asp 版本]	UTF-8版 GBK版	2016-05-26
[1.4.3.3 .Net 版本]	UTF-8版 GBK版	2016-05-26
[1.4.3.3 Jsp 版本]	UTF-8版 GBK版	2016-02-25

历史版本

[1.4.3.2 完整源码]	完整源码 源码版如何编译?	2016-02-25
[1.4.3.2 PHP 版本]	UTF-8版 GBK版	2016-02-25
[1.4.3.2 Asp 版本]	UTF-8版 GBK版	2016-02-25
[1.4.3.2 .Net 版本]	UTF-8版 GBK版	2016-02-25
[1.4.3.2 Jsp 版本]	UTF-8版 GBK版	2016-02-25

其他

本地表情文件	下载地址
本地列表文件	下载地址

@ 百度ueditor版权所有

https://blog.csdn.net/Sunshine_Moon/

Part 2:搭建运行环境

1. 将源码文件夹中这个文件夹放入\src\main\java\com\下



2. 将jsp文件下这些东西放入\src\main\resources\static下

名称	修改日期	类型	大小
dialogs	2016/8/10 11:06	文件夹	
jsp	2016/8/10 11:06	文件夹	
lang	2016/8/10 11:06	文件夹	
themes	2016/8/10 11:06	文件夹	
third-party	2016/8/10 11:06	文件夹	
index.html	2016/8/10 11:06	HTML 文件	7 KB
ueditor.all.js	2016/8/10 11:06	JScript Script 文件	1,111 KB
ueditor.all.min.js	2016/8/10 11:06	JScript Script 文件	375 KB
ueditor.config.js	2016/8/10 11:06	JScript Script 文件	23 KB
ueditor.parse.js	2016/8/10 11:06	JScript Script 文件	36 KB
ueditor.parse.min.js	2016/8/10 11:06	JScript Script 文件	15 KB

https://blog.csdn.net/Sunshine_Moon

3. 配置资源文件路径[1.5版本springboot可以不配]

```
1  @Configuration
2  public class WebMvcAdapterConfig extends WebMvcConfigurationSupport{
3
4      @Override
5      protected void addResourceHandlers(ResourceHandlerRegistry registry) {
6          // TODO Auto-generated method stub
7          registry.addResourceHandler("/static/**").addResourceLocations("classpath:/static/");
8          registry.addResourceHandler("/path/**").addResourceLocations("file:/C:/upload/img/");
9          super.addResourceHandlers(registry);
10     }
11
12     @Override
```

```

13     public void addViewControllers(ViewControllerRegistry registry) {
14         // TODO Auto-generated method stub
15         registry.addViewController("/index.html").setViewName("index");
16         super.addViewControllers(registry);
17     }
18 }

```

4. 添加依赖包

```

<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.3.3</version>
</dependency>

<dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.4</version>
</dependency>
<!--UEditor依赖的jar包-->
<dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>20160810</version>
</dependency>

<dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.11</version>
</dependency>
<dependency>
    <groupId>cn.meddb</groupId>
    <artifactId>ueditor</artifactId>
    <version>20180110</version>
</dependency>

```

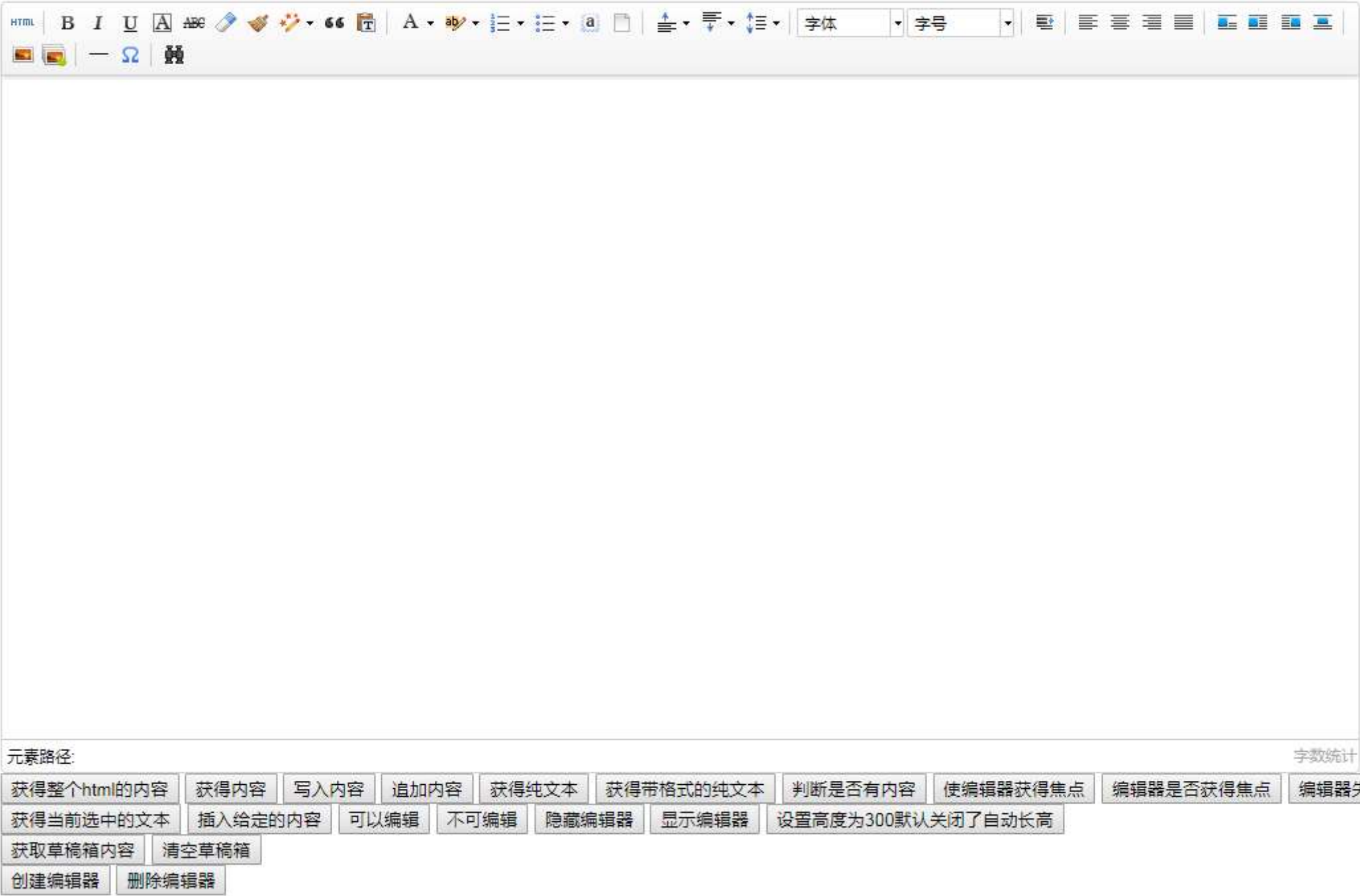
https://blog.csdn.net/Sunshine_Moon

5. 修改index.html js引用路径,根据4中配置决定

```
<script type="text/javascript" charset="utf-8" src="static/ueditor.config.js"></script>
<script type="text/javascript" charset="utf-8" src="static/ueditor.all.min.js"> </script>
<!-- 建议手动加在语言，避免在ie下有时因为加载语言失败导致编辑器加载失败-->
<!-- 这里加载的语言文件会覆盖你在配置项目里添加的语言类型，比如你在配置项目里配置的是英文，这里加载的中文，那最后就是中文-->
<script type="text/javascript" charset="utf-8" src="static/lang/zh-cn/zh-cn.js"></script>
```


6. 启动项目,如果访问页面看到下图,代表配置初步成功,已经可以使用基本功能[我的工具栏自定义过,所以会比正常的少一些控件]

完整demo



https://blog.csdn.net/Sunshine_Moon

Part 3:上传图片功能配置

1. 完成Part 2配置后,点击图片上传功能,发现不可用,提示后台配置没有正常加载

2. 将jsp文件夹下的config.json文件放入\src\main\resources\下



3. 依据源码里的controller.jsp，写一个映射路径为config的控制层方法

```
1  /**
2   * 百度富文本编辑器
3   * @param request
4   * @param response
5   */
6  @RequestMapping(value="/config")
7  public void config(HttpServletRequest request, HttpServletResponse response) {
8      response.setContentType("application/json");
9      String rootPath = request.getSession().getServletContext().getRealPath("/");
10     try {
11         String exec = new ActionEnter(request, rootPath).exec();
12         PrintWriter writer = response.getWriter();
13         writer.write(exec);
14         writer.flush();
15         writer.close();
16     } catch (JSONException e) {
17         e.printStackTrace();
18     } catch (IOException e) {
19         // TODO Auto-generated catch block
20         e.printStackTrace();
21     }
22 }
```

4. 修改ConfigManage类的getConfigPath()方法

```
1  private String getConfigPath () {
2      try{
3          //获取classpath下的config.json路径
4          return this.getClass().getClassLoader().getResource("config.json").toURI().getPath();
5      }catch (URISyntaxException e){
6          return null;
7      }
```

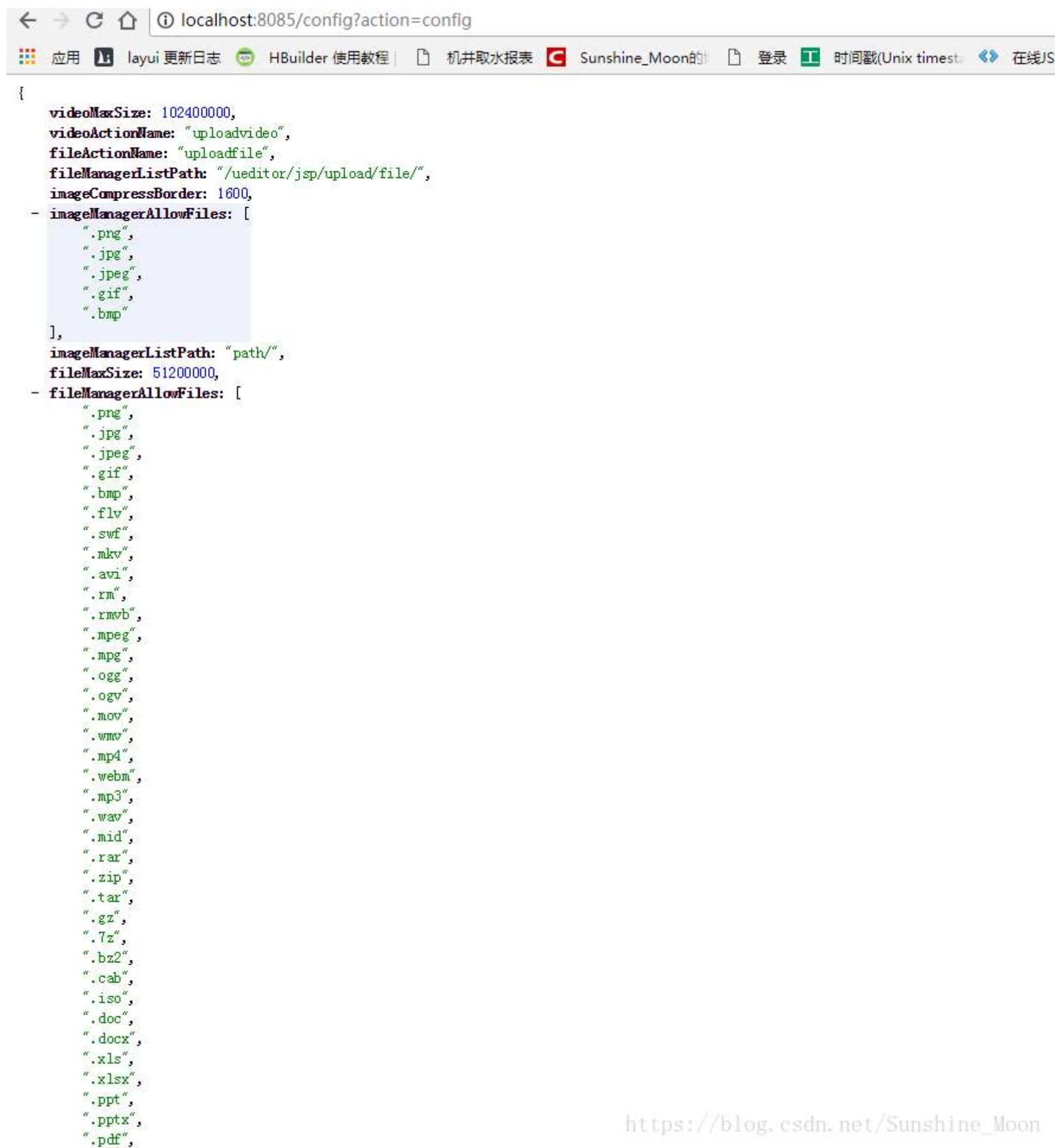
```
7      }  
8  
9  }
```

5. 配置ueditor.config.js

```
ueditor.config.js  
16      * 鉴于很多同学在使用编辑器的时候出现的种种路径问题，此处强烈建议大家使用"相对于网站根目录的相对路径"进行配置。  
17      * "相对于网站根目录的相对路径"也就是以斜杠开头的形如"/myProject/ueditor/"这样的路径。  
18      * 如果站点中有多个不在同一层级的页面需要实例化编辑器，且引用了同一UEditor的时候，此处的URL可能不适用于每个页面的编辑器。  
19      * 因此，UEditor提供了针对不同页面的编辑器可单独配置的根路径，具体来说，在需要实例化编辑器的页面最顶部写上如下代码即可。当然，需要令此处的URL等于对应的  
20      * window.UEEDITOR_HOME_URL = "/xxxx/xxxx/";  
21      */  
22      var URL = window.UEEDITOR_HOME_URL || getUEBasePath();  
23  
24      /**  
25      * 配置项主体。注意，此处所有涉及到路径的配置别遗漏URL变量。  
26      */  
27      window.UEEDITOR_CONFIG = {  
28  
29          //为编辑器实例添加一个路径，这个不能被注释  
30          UEDITOR_HOME_URL: URL  
31  
32          // 服务器统一请求接口路径  
33          , serverUrl: "/config"  
34  
35          //工具栏上的所有的功能按钮和下拉框，可以在new编辑器的实例时选择自己需要的重新定义  
36          , toolbars: [[  
37              'source', '|',  
38              'bold', 'italic', 'underline', 'fontborder', 'strikethrough', 'removeformat', 'formatmatch', 'autotypeset',  
39              'rowspacingtop', 'rowspacingbottom', 'lineheight', '|',  
40              'fontfamily', 'fontsize', '|',  
41              'indent', '|',  
42              'justifyleft', 'justifycenter', 'justiftright', 'justifyjustify', '|',  
43              'imagenone', 'imageleft', 'imageright', 'imagecenter', '|',  
44              'simpleupload', 'insertimage', '|',  
45              'horizontal', 'spechars', '|',  
46              'searchreplace'
```

https://blog.csdn.net/Sunshine_Moon

6. 运行项目路径<http://localhost:8080/config?action=config>, 如下图显示则表示可读取到config.json文件[我这是自动转换json格式插件, 正常情况配置文件内容输出出来就可以了]



The screenshot shows a web browser window with the address bar displaying `localhost:8085/config?action=config`. The browser's developer tools are open, showing the JSON response of the request. The JSON object contains configuration settings for video and file uploads, including maximum sizes, action names, file manager paths, image compress borders, and lists of allowed file extensions for both image and file managers.

```
{
  videoMaxSize: 102400000,
  videoActionName: "uploadvideo",
  fileActionName: "uploadfile",
  fileManagerListPath: "/ueditor/jsp/upload/file/",
  imageCompressBorder: 1600,
  - imageManagerAllowFiles: [
    ".png",
    ".jpg",
    ".jpeg",
    ".gif",
    ".bmp"
  ],
  imageManagerListPath: "path/",
  fileMaxSize: 51200000,
  - fileManagerAllowFiles: [
    ".png",
    ".jpg",
    ".jpeg",
    ".gif",
    ".bmp",
    ".flv",
    ".swf",
    ".mkv",
    ".avi",
    ".rm",
    ".rmvb",
    ".mpeg",
    ".mpg",
    ".ogg",
    ".ogv",
    ".mov",
    ".wmv",
    ".mp4",
    ".webm",
    ".mp3",
    ".wav",
    ".mid",
    ".rar",
    ".zip",
    ".tar",
    ".gz",
    ".7z",
    ".bz2",
    ".cab",
    ".iso",
    ".doc",
    ".docx",
    ".xls",
    ".xlsx",
    ".ppt",
    ".pptx",
    ".pdf",
  ]
}
```

7. 修改BinaryUploader 类, 解决其无法获得带字节流的request的问题

```
1  public class BinaryUploader {
2
3      public static final State save(HttpServletRequest request,
4          Map<String, Object> conf) {
5          // FileStream fileStream = null;
6          // boolean isAjaxUpload = request.getHeader( "X_Requested_With" ) != null;
7
8          if (!ServletFileUpload.isMultipartContent(request)) {
9              return new BaseState(false, AppInfo.NOT_MULTIPART_CONTENT);
10         }
11
12         // ServletFileUpload upload = new ServletFileUpload(
13             // new DiskFileItemFactory());
14         //
15         // if ( isAjaxUpload ) {
16         //     upload.setHeaderEncoding( "UTF-8" );
17         // }
18
19         try {
20             // FileItemIterator iterator = upload.getItemIterator(request);
21             //
22             // while (iterator.hasNext()) {
23             //     fileStream = iterator.next();
24             //
25             //     if (!fileStream.isFormField())
26             //         break;
27             //     fileStream = null;
28             // }
29             //
30             // if (fileStream == null) {
31             //     return new BaseState(false, AppInfo.NOTFOUND_UPLOAD_DATA);
32             // }
33             MultipartHttpServletRequest multipartRequest = (MultipartHttpServletRequest) request;
34             MultipartFile multipartFile = multipartRequest.getFile(conf.get("fieldName").toString());
35             if(multipartFile==null){
36                 return new BaseState(false, AppInfo.NOTFOUND_UPLOAD_DATA);
37             }
38
39             String savePath = (String) conf.get("savePath");
40             //String originFileName = fileStream.getName();
41             String originFileName = multipartFile.getOriginalFilename();
42             String suffix = FileType.getSuffixByFilename(originFileName);
43
44             originFileName = originFileName.substring(0,
45                 originFileName.length() - suffix.length());
```

```

46         savePath = savePath + suffix;
47
48         long maxSize = ((Long) conf.get("maxSize")).longValue();
49
50         if (!validType(suffix, (String[]) conf.get("allowFiles"))) {
51             return new BaseState(false, AppInfo.NOT_ALLOW_FILE_TYPE);
52         }
53
54         savePath = PathFormat.parse(savePath, originFileName);
55
56         String physicalPath = (String) conf.get("rootPath") + savePath;
57
58         //InputStream is = fileStream.openStream();
59         InputStream is = multipartFile.getInputStream();
60         State storageState = StorageManager.saveFileByInputStream(is,
61             physicalPath, maxSize);
62         is.close();
63
64         if (storageState.isSuccess()) {
65             storageState.putInfo("url", PathFormat.format(savePath));
66             storageState.putInfo("type", suffix);
67             storageState.putInfo("original", originFileName + suffix);
68         }
69
70         return storageState;
71         // } catch (FileUploadException e) {
72         //     return new BaseState(false, AppInfo.PARSE_REQUEST_ERROR);
73         // } catch (IOException e) {
74         // }
75         return new BaseState(false, AppInfo.IO_ERROR);
76     }
77
78
79
80     private static boolean validType(String type, String[] allowTypes) {
81         List<String> list = Arrays.asList(allowTypes);
82
83         return list.contains(type);
84     }
85 }

```

Part 4:配置图片上传路径

1. 打开config.json文件配置上传路径[图片回显路径是映射图片上传路径的,一般在Part 2 第3步实现]

```
/* 前后端通信相关的配置,注释只允许使用多行方式 */
{
    "basePath": "C:/upload/img/",
    /* 上传图片配置项 */
    "imageActionName": "uploadimage", /* 执行上传图片的action名称 */
    "imageFieldName": "upfile", /* 提交的图片表单名称 */
    "imageMaxSize": 2048000, /* 上传大小限制,单位B */
    "imageAllowFiles": [".png", ".jpg", ".jpeg", ".gif", ".bmp"], /* 上传图片格式显示 */
    "imageCompressEnable": true, /* 是否压缩图片,默认是true */
    "imageCompressBorder": 1600, /* 图片压缩最长边限制 */
    "imageInsertAlign": "none", /* 插入的图片浮动方式 */
    "imageUrlPrefix": "/path/", /* 图片访问路径前缀 */
    "imagePathFormat": "{time}{rand:6}", /* 上传保存路径,可以自定义保存路径和文件名格式 */
    /* {filename} 会替换成原文件名,配置这项需要注意中文乱码问题 */
    /* {rand:6} 会替换成随机数,后面的数字是随机数的位数 */
    /* {time} 会替换成时间戳 */
    /* {yyyy} 会替换成四位年份 */
    /* {yy} 会替换成两位年份 */
    /* {mm} 会替换成两位月份 */
    /* {dd} 会替换成两位日期 */
    /* {hh} 会替换成两位小时 */
    /* {ii} 会替换成两位分钟 */
    /* {ss} 会替换成两位秒 */
    /* 非法字符 \ : * ? " < > | */
    /* 具体看线上文档: fex.baidu.com/ueditor/#use-format_upload_filename */

    /* 涂鸦图片上传配置项 */
    "scrawlActionName": "uploadscrawl", /* 执行上传涂鸦的action名称 */
    "scrawlFieldName": "upfile", /* 提交的图片表单名称 */
    "scrawlPathFormat": "/ueditor/jsp/upload/image/{yyyy}{mm}{dd}/{time}{rand:6}", /* 上传保存路径,可以自定义保存路径和文件名格式 */
    "scrawlMaxSize": 2048000, /* 上传大小限制,单位B */
    "scrawlUrlPrefix": "", /* 图片访问路径前缀 */
    "scrawlInsertAlign": "none",

    /* 截图工具上传 */

```

图片上传基础路径

图片回显路径

图片命名规则

https://blog.csdn.net/Sunshine_Moon

2. 打开ConfigManager.java,在getConfig方法中修改

```
1 conf.put( "basePath", this.jsonConfig.getString("basePath") );
2 conf.put( "savePath", savePath );
3 conf.put( "rootPath", this.rootPath );
```

3. 打开BinaryUploader.java,在save方法中修改


```
1 String basePath=(String) conf.get("basePath");
2 String physicalPath = basePath + savePath;
```

4. 打开application.properties新增

```
1 web.upload-path=C:/
2 spring.mvc.static-path-pattern=/**
3 spring.resources.static-locations=classpath:/META-INF/resources/,classpath:/resources/,classpath:/static/,classpath:/public/,file:${web.upload-path}
```