

1 将百度的静态文件(js,css,png等) 目录结构为

```
1 src/main/resources/static/ext/baidu/dialogs
2 src/main/resources/static/ext/baidu/lang
3 src/main/resources/static/ext/baidu/themes
4 src/main/resources/static/ext/baidu/...
5 src/main/resources/static/ext/baidu/...
6 src/main/resources/static/ext/baidu/...
```

其中最重要的文件是 ueditor.config.js 文件， 修改其中的内容，将下面所示的serverUrl 改成自己的Controller对应的请求地址

```
1 // 服务器统一请求接口路径
2 , serverUrl: "/config"
```

2 新建Controller

配置Controller

```
1 import java.io.IOException;
2 import java.io.PrintWriter;
3
4 import javax.servlet.http.HttpServletRequest;
5 import javax.servlet.http.HttpServletResponse;
6
7 import org.json.JSONException;
8 import org.springframework.stereotype.Controller;
9 import org.springframework.web.bind.annotation.RequestMapping;
10 import com.baidu.ueditor.ActionEnter;
11
12 @Controller
13 public class UeditorConfigController {
14
15     /**
16      * 百度富文本编辑器
17      * @param request
18      * @param response
19      */
20     @RequestMapping(value="/config")
21     public void config(HttpServletRequest request, HttpServletResponse response) {
22         response.setContentType("application/json");
23         String rootPath = request.getSession().getServletContext().getRealPath("/");
24         try {
25             String exec = new ActionEnter(request, rootPath).exec();
26             PrintWriter writer = response.getWriter();
27             writer.write(exec);
28             writer.flush();
29             writer.close();
30         } catch (JSONException e) {
31             e.printStackTrace();
32         } catch (IOException e) {
33             // TODO Auto-generated catch block
34             e.printStackTrace();
35         }
36     }
37 }
```



```

23             /* {ss} 会替换成两位秒 */
24             /* 非法字符 \ : * ? " < > | */
25             /* 具体看线上文档: fex.baidu.com/ueditor/#use-format_upload_filename
26
27         /* 涂鸦图片上传配置项 */
28         "scrawlActionName": "uploadscrawl", /* 执行上传涂鸦的action名称 */
29         "scrawlFieldName": "upfile", /* 提交的图片表单名称 */
30         "scrawlPathFormat": "/ueditor/jsp/upload/image/{yyyy}{mm}{dd}/{time}{rand:6}", /* 上传保存路径,可
31         "scrawlMaxSize": 2048000, /* 上传大小限制,单位B */
32         "scrawlUrlPrefix": "", /* 图片访问路径前缀 */
33         "scrawlInsertAlign": "none",
34
35         /* 截图工具上传 */
36         "snapscreenActionName": "uploadimage", /* 执行上传截图的action名称 */
37         "snapscreenPathFormat": "/ueditor/jsp/upload/image/{yyyy}{mm}{dd}/{time}{rand:6}", /* 上传保存路
38         "snapscreenUrlPrefix": "", /* 图片访问路径前缀 */
39         "snapscreenInsertAlign": "none", /* 插入的图片浮动方式 */
40
41         /* 抓取远程图片配置 */
42         "catcherLocalDomain": ["127.0.0.1", "localhost", "img.baidu.com"],
43         "catcherActionName": "catchimage", /* 执行抓取远程图片的action名称 */
44         "catcherFieldName": "source", /* 提交的图片列表表单名称 */
45         "catcherPathFormat": "/ueditor/jsp/upload/image/{yyyy}{mm}{dd}/{time}{rand:6}", /* 上传保存路径,
46         "catcherUrlPrefix": "", /* 图片访问路径前缀 */
47         "catcherMaxSize": 2048000, /* 上传大小限制,单位B */
48         "catcherAllowFiles": [".png", ".jpg", ".jpeg", ".gif", ".bmp"], /* 抓取图片格式显示 */
49
50         /* 上传视频配置 */
51         "videoActionName": "uploadvideo", /* 执行上传视频的action名称 */
52         "videoFieldName": "upfile", /* 提交的视频表单名称 */
53         "videoPathFormat": "/ueditor/jsp/upload/video/{yyyy}{mm}{dd}/{time}{rand:6}", /* 上传保存路径,可
54         "videoUrlPrefix": "", /* 视频访问路径前缀 */
55         "videoMaxSize": 102400000, /* 上传大小限制,单位B,默认100MB */
56         "videoAllowFiles": [
57             ".flv", ".swf", ".mkv", ".avi", ".rm", ".rmvb", ".mpeg", ".mpg",
58             ".ogg", ".ogv", ".mov", ".wmv", ".mp4", ".webm", ".mp3", ".wav", ".mid"], /* 上传视频格式显示
59
60         /* 上传文件配置 */
61         "fileActionName": "uploadfile", /* controller里,执行上传视频的action名称 */
62         "fileFieldName": "upfile", /* 提交的文件表单名称 */
63         "filePathFormat": "/ueditor/jsp/upload/file/{yyyy}{mm}{dd}/{time}{rand:6}", /* 上传保存路径,可以自
64         "fileUrlPrefix": "", /* 文件访问路径前缀 */
65         "fileMaxSize": 51200000, /* 上传大小限制,单位B,默认50MB */
66         "fileAllowFiles": [
67             ".png", ".jpg", ".jpeg", ".gif", ".bmp",
68             ".flv", ".swf", ".mkv", ".avi", ".rm", ".rmvb", ".mpeg", ".mpg",
69             ".ogg", ".ogv", ".mov", ".wmv", ".mp4", ".webm", ".mp3", ".wav", ".mid",
70             ".rar", ".zip", ".tar", ".gz", ".7z", ".bz2", ".cab", ".iso",
71             ".doc", ".docx", ".xls", ".xlsx", ".ppt", ".pptx", ".pdf", ".txt", ".md", ".xml"
72         ], /* 上传文件格式显示 */
73
74         /* 列出指定目录下的图片 */
75         "imageManagerActionName": "listimage", /* 执行图片管理的action名称 */
76         "imageManagerListPath": "/ueditor/jsp/upload/image/", /* 指定要列出图片的目录 */

```

```

77     "imageManagerListSize": 20, /* 每次列出文件数量 */
78     "imageManagerUrlPrefix": "", /* 图片访问路径前缀 */
79     "imageManagerInsertAlign": "none", /* 插入的图片浮动方式 */
80     "imageManagerAllowFiles": [".png", ".jpg", ".jpeg", ".gif", ".bmp"], /* 列出的文件类型 */
81
82     /* 列出指定目录下的文件 */
83     "fileManagerActionName": "listfile", /* 执行文件管理的action名称 */
84     "fileManagerListPath": "/ueditor/jsp/upload/file/", /* 指定要列出文件的目录 */
85     "fileManagerUrlPrefix": "", /* 文件访问路径前缀 */
86     "fileManagerListSize": 20, /* 每次列出文件数量 */
87     "fileManagerAllowFiles": [
88         ".png", ".jpg", ".jpeg", ".gif", ".bmp",
89         ".flv", ".swf", ".mkv", ".avi", ".rm", ".rmvb", ".mpeg", ".mpg",
90         ".ogg", ".ogv", ".mov", ".wmv", ".mp4", ".webm", ".mp3", ".wav", ".mid",
91         ".rar", ".zip", ".tar", ".gz", ".7z", ".bz2", ".cab", ".iso",
92         ".doc", ".docx", ".xls", ".xlsx", ".ppt", ".pptx", ".pdf", ".txt", ".md", ".xml"
93     ] /* 列出的文件类型 */
94
95 }

```

5 测试

controller跳转到页面去

```

1  import java.io.IOException;
2  import java.io.PrintWriter;
3
4  import javax.servlet.http.HttpServletRequest;
5  import javax.servlet.http.HttpServletResponse;
6
7  import org.json.JSONException;
8  import org.springframework.stereotype.Controller;
9  import org.springframework.web.bind.annotation.RequestMapping;
10
11 import com.baidu.ueditor.ActionEnter;
12
13 @Controller
14 @RequestMapping("/ueditor")
15 public class UeditorController extends BaseController{
16
17     @RequestMapping("index.do")
18     public String index() {
19         return "system/ueditor/index";
20     }
21 }

```

使用的页面system/ueditor/index.html代码

```

1 <html>
2 <head>
3     <title>完整demo</title>
4     <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5     <script type="text/javascript" charset="utf-8" th:src="@{/ext/baidu/ueditor.config.js}"></script>
6     <script type="text/javascript" charset="utf-8" th:src="@{/ext/baidu/ueditor.all.min.js}"></script>

```

```

7      <!-- 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-->
8      <!-- 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
9      <script type="text/javascript" charset="utf-8" th:src="@{/ext/baidu/lang/zh-cn/zh-cn.js}"></script>
10
11      <style type="text/css">
12          div{
13              width:100%;
14          }
15      </style>
16  </head>
17  <body>
18  <div>
19      <h1>完整demo</h1>
20      <script id="editor" type="text/plain" style="width:1024px;height:500px;"></script>
21  </div>
22  <div id="btns">
23      <div>
24          <button onclick="getAllHtml()">获得整个html的内容</button>
25          <button onclick="getContent()">获得内容</button>
26          <button onclick="setContent()">写入内容</button>
27          <button onclick="setContent(true)">追加内容</button>
28          <button onclick="getContentTxt()">获得纯文本</button>
29          <button onclick="getPlainTxt()">获得带格式的纯文本</button>
30          <button onclick="hasContent()">判断是否有内容</button>
31          <button onclick="setFocus()">使编辑器获得焦点</button>
32          <button onmousedown="isFocus(event)">编辑器是否获得焦点</button>
33          <button onmousedown="setblur(event)">编辑器失去焦点</button>
34
35      </div>
36      <div>
37          <button onclick="getText()">获得当前选中的文本</button>
38          <button onclick="insertHtml()">插入给定的内容</button>
39          <button id="enable" onclick="setEnabled()">可以编辑</button>
40          <button onclick="setDisabled()">不可编辑</button>
41          <button onclick=" UE.getEditor('editor').setHide()">隐藏编辑器</button>
42          <button onclick=" UE.getEditor('editor').setShow()">显示编辑器</button>
43          <button onclick=" UE.getEditor('editor').setHeight(300)">设置高度为300默认关闭了自动长高</button>
44      </div>
45
46      <div>
47          <button onclick="getLocalData()">获取草稿箱内容</button>
48          <button onclick="clearLocalData()">清空草稿箱</button>
49      </div>
50
51  </div>
52  <div>
53      <button onclick="createEditor()">
54          创建编辑器</button>
55      <button onclick="deleteEditor()">
56          删除编辑器</button>
57  </div>
58
59  <script type="text/javascript">
60

```

```

61 //实例化编辑器
62 //建议使用工厂方法getEditor创建和引用编辑器实例，如果在某个闭包下引用该编辑器，直接调用UE.getEditor('editor')
63 var ue = UE.getEditor('editor');
64
65
66 function isFocus(e){
67     alert(UE.getEditor('editor').isFocus());
68     UE.dom.domUtils.preventDefault(e)
69 }
70 function setblur(e){
71     UE.getEditor('editor').blur();
72     UE.dom.domUtils.preventDefault(e)
73 }
74 function insertHtml() {
75     var value = prompt('插入html代码', '');
76     UE.getEditor('editor').execCommand('insertHtml', value)
77 }
78 function createEditor() {
79     enableBtn();
80     UE.getEditor('editor');
81 }
82 function getAllHtml() {
83     alert(UE.getEditor('editor').getAllHtml())
84 }
85 function getContent() {
86     var arr = [];
87     arr.push("使用editor.getContent()方法可以获得编辑器的内容");
88     arr.push("内容为: ");
89     arr.push(UE.getEditor('editor').getContent());
90     alert(arr.join("\n"));
91 }
92 function getPlainTxt() {
93     var arr = [];
94     arr.push("使用editor.getPlainTxt()方法可以获得编辑器的带格式的纯文本内容");
95     arr.push("内容为: ");
96     arr.push(UE.getEditor('editor').getPlainTxt());
97     alert(arr.join('\n'))
98 }
99 function setContent(isAppendTo) {
100     var arr = [];
101     arr.push("使用editor.setContent('欢迎使用ueditor')方法可以设置编辑器的内容");
102     UE.getEditor('editor').setContent('欢迎使用ueditor', isAppendTo);
103     alert(arr.join("\n"));
104 }
105 function setDisabled() {
106     UE.getEditor('editor').setDisabled('fullscreen');
107     disableBtn("enable");
108 }
109
110 function setEnabled() {
111     UE.getEditor('editor').setEnabled();
112     enableBtn();
113 }
114

```

```

115 function getText() {
116     //当你点击按钮时编辑区域已经失去了焦点，如果直接用getText将不会得到内容，所以要在选回来，然后取得内容
117     var range = UE.getEditor('editor').selection.getRange();
118     range.select();
119     var txt = UE.getEditor('editor').selection.getText();
120     alert(txt)
121 }
122
123 function getContentTxt() {
124     var arr = [];
125     arr.push("使用editor.getContentTxt()方法可以获得编辑器的纯文本内容");
126     arr.push("编辑器的纯文本内容为：");
127     arr.push(UE.getEditor('editor').getContentTxt());
128     alert(arr.join("\n"));
129 }
130 function hasContent() {
131     var arr = [];
132     arr.push("使用editor.hasContents()方法判断编辑器里是否有内容");
133     arr.push("判断结果为：");
134     arr.push(UE.getEditor('editor').hasContents());
135     alert(arr.join("\n"));
136 }
137 function setFocus() {
138     UE.getEditor('editor').focus();
139 }
140 function deleteEditor() {
141     disableBtn();
142     UE.getEditor('editor').destroy();
143 }
144 function disableBtn(str) {
145     var div = document.getElementById('btns');
146     var btns = UE.dom.domUtils.getElementsByTagName(div, "button");
147     for (var i = 0, btn; btn = btns[i++];) {
148         if (btn.id == str) {
149             UE.dom.domUtils.removeAttributes(btn, ["disabled"]);
150         } else {
151             btn.setAttribute("disabled", "true");
152         }
153     }
154 }
155 function enableBtn() {
156     var div = document.getElementById('btns');
157     var btns = UE.dom.domUtils.getElementsByTagName(div, "button");
158     for (var i = 0, btn; btn = btns[i++];) {
159         UE.dom.domUtils.removeAttributes(btn, ["disabled"]);
160     }
161 }
162
163 function getLocalData () {
164     alert(UE.getEditor('editor').execCommand( "getlocaldata" ));
165 }
166
167 function clearLocalData () {
168     UE.getEditor('editor').execCommand( "clearlocaldata" );

```

```
169         alert("已清空草稿箱")
170     }
171 </script>
172 </body>
173 </html>
```

ps : 异常情况

1 请求后台配置项http错误, 上传功能将不能正常使用

打开页面出现上面说的错误, 查看浏览器控制台 发现有一个请求报错, 在UeditorConfigController类中进行代码调试

```
1 http://127.0.0.1:8080/config?action=config&&noCache=1548408199632
```

最后发现问题是打包的时候 没有包含 config.json文件