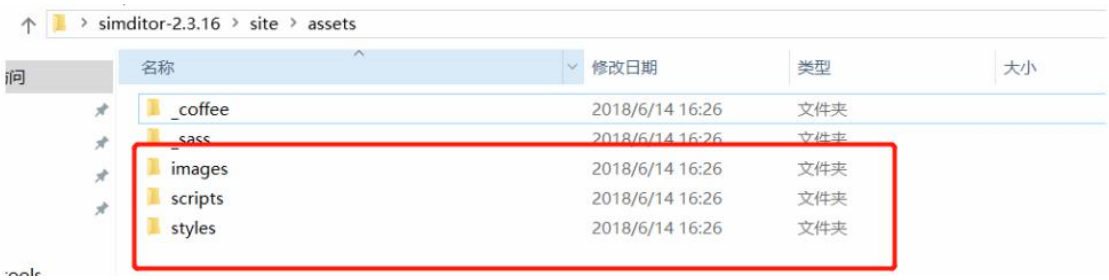
Simditor 是 Tower 开源的一个基于浏览器所见即所得的文本编辑器，它的理念是保持简单，避免过度的功能，使用方法如下：

Step1 .下载路径 下载最新版本的 simditor 的 zip，如图所示：

下载解压后会发现是整个 simditor 项目，很多东西项目根本不需要，只需要将\site\assets 下的 images、scripts、styles 文件夹拷

贝到我们项目中，如图所示：



Step2.在项目的 webapp 中新建文件夹 smiditor，将上面的文件放到此文件夹中，如图所示

在 Step3 . 在页面中引用需要的文件

```
<!--css 引用-->
<div class="page-sidebar-wrapper" th:include="/common/page_sidebar::pageSidebar">
<!--js 引用-->
<script type="text/javascript" th:src="@{/scripts/simditor/scripts/module.js}"></script>
<script type="text/javascript" th:src="@{/scripts/simditor/scripts/hotkeys.js}"></script>
<script type="text/javascript" th:src="@{/scripts/simditor/scripts/uploader.js}"></script>
<script type="text/javascript" th:src="@{/scripts/simditor/scripts/simditor.js}"></script>
```

注意：如果页面中没有引用 jquery 的 js 还需要引用 jquery 的 js

Step4. 在页面中创建一个 textarea 元素，代码如下：

```
<div class="form-group">
        <label class="col-md-3 control-label">详情
          <span class="required"> * </span>
        </label>
      <div class="col-md-7" id="contextText">
    <div class="form-group" align="center" id="nochecke">
          <textarea id="editor" type="text/plain"    hidden="true" autofocus>
          </textarea>
    </div>
     </div>
    <div style="display: none">
    <input type="input" id="content" name="content" value=''>
    </div>
</div>
```

注意：textarea 的 id 值要与接下来的初始化 simditor 的 JS 代码对应起来，隐藏的 input 是 form 表单提交时获取不到 textarea 的值，所以需要一个能获取到的 input，input 的 name 值要和后台接收的字段一致。

为了能实时更新隐藏 input 的值，js 中需要有个同步的方法

```
function contextTextOnblur() {
    $(document).on("blur", "#contextText", function() {
        var content = $(".simditor-body").html();
        $("#content").val(content);
    });
}
```

Step5. 初始化 simditor，代码如下：

```
$(function(){
        Simditor.locale = 'zh-CN';//设置中文
        var editor = new Simditor({
            textarea: $("#editor"),   //textarea 的 id
            placeholder: '',
            toolbar:   ['title', 'bold', 'italic', 'underline', 'strikethrough', 'fontScale', 'color', '|',
```

```
'ol', 'ul', 'blockquote', 'code', 'table', '|', 'link', 'image', 'hr', '|', 'indent', 'outdent', 'alignment'],
//工具条都包含哪些内容
                pasteImage: true,//允许粘贴图片
                defaultImage: '/simditor/images/image.png',//编辑器插入的默认图片，此处可
以删除
                upload : {
                    url : '/smiditor/uploadSimditorImg', //文件上传的接口地址
                    params: null, //键值对,指定文件上传接口的额外参数,上传的时候随文件
一起提交
                    fileKey:'file', //服务器端获取文件数据的参数名
                    connectionCount: 3,
                    leaveConfirm: '正在上传文件'
                }
            });
        });
```

注意：文件上需要返回特定的字段，成功时返回

```
{
  "success":true,
  "file_path":""
}
```

失败时返回：

```
{
  "success":false
}
```

Step6. 实现的 controller 中

```
@Controller
@RequestMapping("/smiditor")
public class SmiditorUploadImageController extends BaseController {

    @Autowired
    UploadSmiditorImageServie uploadSmiditorImageServie;

    @RequestMapping("/uploadSimditorImg")
    @ResponseBody
    public JSONObject uploadSimditorImg(MultipartFile file) {
        JSONObject json = new JSONObject();
        try {
            json = uploadSmiditorImageServie.uploadSimditorImg(file);
        } catch (Exception e) {
            LogWriter.writeErrorLog("上传编辑器图片失败：" + e);
        }

        return json;
    }
```

```
}
```

注意： 参数名要和前台的 js 中的 filekey 一致，做图片上传的时候无论图片上传成功与否都会显示出来，二者唯一的区别就是图片的路径， 如果图片上传成功，图片的路径应该是正常的路径，如果图片上传失败图片的路径是被 base64 加密过的路径。

Step7. 实现 service 方法

```
    public JSONObject uploadSimditorImg(MultipartFile file) {
        JSONObject json = new JSONObject();
        try {
        //调用图片上传的方法
        ResultObject<UploadResult>resultObject =fileUploadService.uploadSingleImage(file);
            if(ValidateUtil.validateStringIsNullOrEmpty(resultObject.getData().getFileUrl())
== false){
                json.put("success", true);
                json.put("file_path", resultObject.getData().getFileUrl());
            }else{
                json.put("success", false);
            }
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return json;
    }
```

至此 simditor 就可以正常使用啦。

注意：simditor 默认的宽高是随内容的长短变化的，如果想固定宽高，找到 simditor/styles/simditor.css 文件中 141 行，可以加上固定宽高。

```
.simditor {
  position: relative;
  border: 1px solid #c9d8db;
   width: 80%;
}
```