

狂神说SpringBoot16：富文本编辑器

秦疆 狂神说 2020-03-27

狂神说SpringBoot系列连载课程，通俗易懂，基于SpringBoot2.2.5版本，欢迎各位狂粉转发关注学习。未经作者授权，禁止转载

聊聊富文本编辑器

简介

思考：我们平时在博客园，或者CSDN等平台进行写作的时候，有同学思考过他们的编辑器是怎么实现的吗？

在博客园后台的选项设置中，可以看到一个文本编辑器的选项：



其实这个就是富文本编辑器，市面上有许多非常成熟的富文本编辑器，比如：

- **Editor.md**——功能非常丰富的编辑器，左端编辑，右端预览，非常方便，完全免费
 - 官网：<https://pandao.github.io/editor.md/>
- **wangEditor**——基于javascript和css开发的 Web富文本编辑器，轻量、简洁、界面美观、易用、开源免费。
 - 官网：<http://www.wangeditor.com/>
- **TinyMCE**——TinyMCE是一个轻量级的基于浏览器的所见即所得编辑器，由JavaScript写成。它对IE6+和Firefox1.5+都有着非常良好的支持。功能齐全，界面美观，就是文档是英文的，对开发人员英文水平有一定要求。
 - 官网：<https://www.tiny.cloud/docs/demo/full-featured/>
 - 博客园

- **百度 ueditor**——UEditor是由百度web前端研发部开发所见即所得富文本web编辑器，具有轻量，功能齐全，可定制，注重用户体验等特点，开源基于MIT协议，允许自由使用和修改代码，缺点是已经没有了更新。
 - 官网：<https://ueditor.baidu.com/website/onedemo.html>
- **kindeditor**——界面经典。
 - 官网：<http://kindeditor.net/demo.php>
- **Textbox**——Textbox是一款极简但功能强大的在线文本编辑器，支持桌面设备和移动设备。主要功能包含内置的图像处理和存储、文件拖放、拼写检查和自动更正。此外，该工具还实现了屏幕阅读器等辅助技术，并符合WAI-ARIA可访问性标准。
 - 官网：<https://textbox.io/>
- **CKEditor**——国外的，界面美观。
 - 官网：<https://ckeditor.com/ckeditor-5/demo/>
- **quill**——功能强大，还可以编辑公式等
 - 官网：<https://quilljs.com/>
- **simditor**——界面美观，功能较全。
 - 官网：<https://simditor.tower.im/>
- **summernote**——UI好看，精美
 - 官网：<https://summernote.org/>
- **jodit**——功能齐全
 - 官网：<https://xdsoft.net/jodit/>
- **froala Editor**——界面非常好看，功能非常强大，非常好用（非免费）
 - 官网：<https://www.froala.com/wysiwyg-editor>

总之，目前可用的富文本编辑器有很多.....这只是其中的一部分

Editor.md

我这里使用的就是Editor.md，作为一个资深码农，Markdown必然是我们程序猿最喜欢的格式，看下面，就爱上了！

1、建一个SpringBoot项目配置

```
spring:
datasource:
    username: root
    password: 123456
    #?serverTimezone=UTC解决时区的报错
    url: jdbc:mysql://localhost:3306/springboot?
serverTimezone=UTC&useUnicode=true&characterEncoding=utf-8
    driver-class-name: com.mysql.cj.jdbc.Driver
```

```
<resources>
    <resource>
        <directory>src/main/java</directory>
        <includes>
            <include>**/*.xml</include>
        </includes>
        <filtering>true</filtering>
    </resource>
</resources>
```

2、实体类:

```
//文章类
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Article implements Serializable {

    private int id; //文章的唯一ID
    private String author; //作者名
    private String title; //标题
    private String content; //文章的内容

}
```

3、mapper接口:

```
@Mapper
@Repository
public interface ArticleMapper {

    //查询所有的文章
    List<Article> queryArticles();

    //新增一个文章
    int addArticle(Article article);

    //根据文章id查询文章
    Article getArticleById(int id);

}
```

```
//根据文章id删除文章
int deleteArticleById(int id);
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.kuang.mapper.ArticleMapper">

    <select id="queryArticles" resultType="Article">
        select * from article
    </select>

    <select id="getArticleById" resultType="Article">
        select * from article where id = #{id}
    </select>

    <insert id="addArticle" parameterType="Article">
        insert into article (author,title,content) values (#{author},#{title},#{content});
    </insert>

    <delete id="deleteArticleById" parameterType="int">
        delete from article where id = #{id}
    </delete>

</mapper>
```

既然已经提供了 myBatis 的映射配置文件，自然要告诉 spring boot 这些文件的位置

```
mybatis:
mapper-locations: classpath:com/kuang/mapper/*.xml
type-aliases-package: com.kuang.pojo
```

编写一个Controller测试下，是否ok;

文章编辑整合（重点）

- 1、导入 editor.md 资源，删除多余文件
- 2、编辑文章页面 editor.html、需要引入 jQuery;

```
<!DOCTYPE html>
<html class="x-admin-sm" lang="zh" xmlns:th="http://www.thymeleaf.org">

<head>
    <meta charset="UTF-8">
    <title>秦疆'Blog</title>
```

```

<meta name="renderer" content="webkit">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="viewport" content="width=device-width,user-scalable=yes, minimum-scale=0.4,
initial-scale=0.8,target-densitydpi=low-dpi" />
<!--Editor.md-->
<link rel="stylesheet" th:href="@{/editormd/css/editormd.css}"/>
<link rel="shortcut icon" href="https://pandao.github.io/editor.md/favicon.ico"
type="image/x-icon" />
</head>

<body>

<div class="layui-fluid">
  <div class="layui-row layui-col-space15">
    <div class="layui-col-md12">
      <!--博客表单-->
      <form name="mdEditorForm">
        <div>
          标题: <input type="text" name="title">
        </div>
        <div>
          作者: <input type="text" name="author">
        </div>
        <div id="article-content">
          <textarea name="content" id="content" style="display:none;"> </textarea>
        </div>
      </form>

    </div>
  </div>
</div>
</body>

<!--editormd-->
<script th:src="@{/editormd/lib/jquery.min.js}"></script>
<script th:src="@{/editormd/editormd.js}"></script>

<script type="text/javascript">
  var testEditor;

  //window.onload = function(){ }
  $(function() {
    testEditor = editormd("article-content", {
      width : "95%",
      height : 400,
      syncScrolling : "single",
      path : "../editormd/lib/",
      saveHTMLToTextarea : true,    // 保存 HTML 到 Textarea
      emoji: true,
      theme: "dark",//工具栏主题
      previewTheme: "dark",//预览主题
    });
  });

```

```

editorTheme: "pastel-on-dark", //编辑主题
tex : true, // 开启科学公式TeX语言支持，默认关闭
flowChart : true, // 开启流程图支持，默认关闭
sequenceDiagram : true, // 开启时序/序列图支持，默认关闭，
//图片上传
imageUpload : true,
imageFormats : ["jpg", "jpeg", "gif", "png", "bmp", "webp"],
imageUploadURL : "/article/file/upload",
onload : function() {
    console.log('onload', this);
},
/*指定需要显示的功能按钮*/
toolbarIcons : function() {
    return ["undo","redo","|",
        "bold","del","italic","quote","ucwords","uppercase","lowercase","|",
        "h1","h2","h3","h4","h5","h6","|",
        "list-ul","list-ol","hr","|",
        "link","reference-link","image","code","preformatted-text",
        "code-block","table","datetime","emoji","html-entities","pagebreak","|",
        "goto-line","watch","preview","fullscreen","clear","search","|",
        "help","info","releaseIcon","index"]
},

/*自定义功能按钮，下面我自定义了2个，一个是发布，一个是返回首页*/
toolbarIconTexts : {
    releaseIcon : "<span bgcolor=\`gray\`>发布</span>",
    index : "<span bgcolor=\`red\`>返回首页</span>",
},

/*给自定义按钮指定回调函数*/
toolbarHandlers:{
    releaseIcon : function(cm, icon, cursor, selection) {
        //表单提交
        mdEditorForm.method = "post";
        mdEditorForm.action = "/article/addArticle";//提交至服务器的路径
        mdEditorForm.submit();
    },
    index : function(){
        window.location.href = '/';
    },
}
});
});
</script>

</html>

```

3、编写Controller，进行跳转，以及保存文章

```

@Controller
@RequestMapping("/article")
public class ArticleController {

```

```

@GetMapping("/toEditor")
public String toEditor(){
    return "editor";
}

@PostMapping("/addArticle")
public String addArticle(Article article){
    articleMapper.addArticle(article);
    return "editor";
}
}

```

图片上传问题

1、前端js中添加配置

```

//图片上传
imageUpload : true,
imageFormats : ["jpg", "jpeg", "gif", "png", "bmp", "webp"],
imageUploadURL : "/article/file/upload", // 这个是上传图片时的访问地址

```

2、后端请求，接收保存这个图片，需要导入 FastJson 的依赖！

```

//博客图片上传问题
@RequestMapping("/file/upload")
@ResponseBody
public JSONObject fileUpload(@RequestParam(value = "editormd-image-file", required = true)
MultipartFile file, HttpServletRequest request) throws IOException {
    //上传路径保存设置

    //获得SpringBoot当前项目的路径: System.getProperty("user.dir")
    String path = System.getProperty("user.dir")+"/upload/";

    //按照月份进行分类:
    Calendar instance = Calendar.getInstance();
    String month = (instance.get(Calendar.MONTH) + 1)+"月";
    path = path+month;

    File realPath = new File(path);
    if (!realPath.exists()){
        realPath.mkdir();
    }

    //上传文件地址
    System.out.println("上传文件保存地址: "+realPath);

    //解决文件名问题: 我们使用uuid;
    String filename = "ks-"+UUID.randomUUID().toString().replaceAll("-", "");
    //通过CommonsMultipartFile的方法直接写文件（注意这个时候）

```



```

file.transferTo(new File(realPath + "/" + filename));

//给editormd进行回调
JSONObject res = new JSONObject();
res.put("url", "/upload/" + month + "/" + filename);
res.put("success", 1);
res.put("message", "upload success!");

return res;
}

```

3、解决文件回显显示的问题，设置虚拟目录映射！在我们自己拓展的MvcConfig中进行配置即可！

```

@Configuration
public class MyMvcConfig implements WebMvcConfigurer {

    // 文件保存在真实目录/upload/下，
    // 访问的时候使用虚路径/upload，比如文件名为1.png，就直接/upload/1.png就ok了。
    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/upload/**")
            .addResourceLocations("file:" + System.getProperty("user.dir") + "/upload/");
    }
}

```

表情包问题

自己手动下载，emoji 表情包，放到图片路径下：

修改editormd.js文件

```

// Emoji graphics files url path
editormd.emoji = {
    path : "../editormd/plugins/emoji-dialog/emoji/",
    ext : ".png"
};

```

文章展示

1、Controller 中增加方法

```

@GetMapping("/{id}")
public String show(@PathVariable("id") int id, Model model) {
    Article article = articleMapper.getArticleById(id);
    model.addAttribute("article", article);
    return "article";
}

```

2、编写页面 article.html

```
<!DOCTYPE html>
```

```

<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
  <title th:text="${article.title}"></title>
</head>
<body>

<div>
  <!--文章头部信息：标题，作者，最后更新日期，导航-->
  <h2 style="margin: auto 0" th:text="${article.title}"></h2>
  作者: <span style="float: left" th:text="${article.author}"></span>
  <!--文章主体内容-->
  <div id="doc-content">
    <textarea style="display:none;" placeholder="markdown" th:text="${article.content}">
  </textarea>
  </div>

</div>

<link rel="stylesheet" th:href="@{/editormd/css/editormd.preview.css}" />
<script th:src="@{/editormd/lib/jquery.min.js}"></script>
<script th:src="@{/editormd/lib/marked.min.js}"></script>
<script th:src="@{/editormd/lib/prettify.min.js}"></script>
<script th:src="@{/editormd/lib/raphael.min.js}"></script>
<script th:src="@{/editormd/lib/underscore.min.js}"></script>
<script th:src="@{/editormd/lib/sequence-diagram.min.js}"></script>
<script th:src="@{/editormd/lib/flowchart.min.js}"></script>
<script th:src="@{/editormd/lib/jquery.flowchart.min.js}"></script>
<script th:src="@{/editormd/editormd.js}"></script>

<script type="text/javascript">
  var testEditor;
  $(function () {
    testEditor = editormd.markdownToHTML("doc-content", { //注意：这里是上面DIV的id
      htmlDecode: "style,script,iframe",
      emoji: true,
      taskList: true,
      tocm: true,
      tex: true, // 默认不解析
      flowChart: true, // 默认不解析
      sequenceDiagram: true, // 默认不解析
      codeFold: true
    });
  });
</script>
</body>
</html>

```

重启项目，访问进行测试！大功告成！

小结：

有了富文本编辑器，我们网站的功能就会又多一项，大家到了这里完全可以有时间写一个属于自己的博客网站了，根据所学的知识是没有任何问题的！

end

视频同步更新，这次一定！



“赠人玫瑰，手有余香”

狂神说的赞赏码

狂神说



