

Markdown文档预览

需求分解：用户上传Markdown文件，实现页面预览。

Markdown

Markdown文件设计目的是方便写作的时候，标记格式、同时使标记格式后的纯文本，比起 HTML 源码，更有可读性。

Markdown文件的渲染需要首先将其解析成HTML文件，没然后搭配CSS样式进行渲染成对应页面。

第三方库

Remark

[Remark.js](#)

- 世界上最流行的 markdown 解析器
- 会生成AST（抽象语法树），使得检查和更改文件内容更加容易
- 拥有丰富的插件生态，且可自定义

简单来说，Remark通过生成AST树的方式，将markdown 作为结构化数据，使得可以轻松实现转换HTML，格式检查和格式化等操作。

安全性：

由于 markdown 可以转换为 HTML，不当使用 HTML 会使您面临[跨站点脚本\(XSS\)](#)攻击，因此使用 remark 可能不安全。

原因：用户输入可以包括脚本标签和其他类型的跨域活动内容并危害安全。[remark-vue 使用Vue createElement](#)在 Vue 中构建 DOM：这意味着您可以在应用程序中显示已解析和格式化的 Markdown 内容，而无需使用 `domPropsInnerHTML`。

解决方案：采用 [remark-vue](#)

Marked

[Marked.js](#)

- ⚡ 为速度而生
- ⬇ 用于解析降价的低级编译器，无需长时间缓存或阻塞
- 🏊 轻量级，同时实现支持的风格和规格的所有降价功能
- 🌐 在浏览器、服务器或命令行界面 (CLI) 中工作

于是综合上述选型，考虑采用 marked.js，因为它拥有最多的Stars (29.4k)，并且安全性较高。

Marked实现

安装依赖

```
npm install marked
```

```
import marked from 'marked';
```

实时渲染Markdown

无论何时用户在文本框中输入文本，它都会从标记库传递到标记函数，标记库将标记编译成 HTML，并且计算属性会在它发生变化时立即将其重新呈现给我们的应用程序。

```
<template>
<!-- {{ markdown }} -->
```

```
<textarea v-model="markdown"></textarea>
<div v-html="markdownToHtml"></div>

</template>
<script>
import marked from 'marked';
export default {
  name: 'App',
  data(){
    return {
      markdown: "# Hello world",
    };
  },
  computed: {
    markdownToHtml(){
      return marked(this.markdown);
    }
  }
}
</script>
<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```