

使用 Node.js 实现无数据库的静态博客工具

在这篇教程中，我们将学习如何使用 Node.js 创建一个无数据库的静态博客工具。静态博客工具是一种通过静态文件生成和展示博客内容的工具，不需要使用数据库，适合个人博客或小型网站。

1. 初始化项目

首先，我们需要创建一个新的 Node.js 项目，并初始化 `package.json` 文件。

```
1 mkdir static-blog && cd static-blog && npm init -y
```

2. 安装必要的依赖

接下来，我们需要安装一些 npm 包来处理文件和生成 HTML 页面。这里我们将使用 `fs-extra` 来处理文件操作，使用 `marked` 来解析 Markdown 文件。

```
1 npm install fs-extra marked
```

3. 创建文件结构

我们需要创建一个目录结构，用于存放博客文章、模板和生成的静态页面。

```
mkdir -p src/posts && mkdir -p src/templates && mkdir -p dist
```

- `src/posts`：用于存放 Markdown 格式的博客文章。
- `src/templates`：用于存放 HTML 模板。
- `dist`：用于存放生成的静态页面。

4. 编写博客文章

在 `src/posts` 目录下编写一些 Markdown 格式的博客文章。例如，创建一个 `first-post.md` 文件：

```
1 # 第一篇博客文章 这是我的第一篇博客文章。
2 ## 小节 这里是小节内容。
```

5. 创建 HTML 模板

在 `src/templates` 目录下创建一个 `template.html` 文件，用于定义生成的 HTML 页面的结构：

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>{{ title }}</title>
7   </head>
8   <body>
9     <h1>{{title}}</h1>
10    <div>{{content}}</div>
11  </body>
12 </html>
```

这里我们使用 `{{title}}` 和 `{{content}}` 占位符来插入文章的标题和内容。

6. 编写生成脚本

创建一个 `build.js` 文件，用于读取 Markdown 文件、解析内容并生成静态 HTML 页面。

```
1 const fs = require('fs-extra');
2 const path = require('path');
3 const marked = require('marked');
4 const postsDir = path.join(__dirname, 'src', 'posts');
5 const templatePath = path.join(__dirname, 'src', 'templates', 'template.html');
6 const outputDir = path.join(__dirname, 'dist');
7 // 读取模板文件
8 const template = fs.readFileSync(templatePath, 'utf-8');
9 // 读取所有 Markdown 文件
10 const files = fs.readdirSync(postsDir);
11 files.forEach(file => {
12   const filePath = path.join(postsDir, file);
13   const markdown = fs.readFileSync(filePath, 'utf-8');
14   const htmlContent = marked(markdown);
15   // 获取文件名（不包含扩展名）
16   const fileName = path.basename(file, path.extname(file));
17   // 使用模板生成 HTML
```

```
18     const html = template
19       .replace('{{title}}', fileName)
20       .replace('{{content}}', htmlContent);
21     // 输出生成的 HTML 文件
22     fs.outputFileSync(path.join(outputDir,
23   ${fileName}.html
24   ), html);
25   });
26   console.log('博客生成完毕!');
```

7. 运行生成脚本

在 `package.json` 文件中添加一个脚本来运行我们的生成脚本：

```
1 {
2   ...
3   "scripts": {
4     "build": "node build.js"
5   },
6   ...
7 }
```

然后在终端运行：

```
1 npm run build
```

这将读取 `src/posts` 目录下的 Markdown 文件，使用 `src/templates` 目录下的 HTML 模板生成静态页面，并输出到 `dist` 目录。

8. 查看生成的博客

生成完毕后，你可以在 `dist` 目录下查看生成的 HTML 文件。用浏览器打开这些文件，查看生成的静态博客页面。