



用 JSDOC 编写 JavaScript 文档

📅 2017.12.23 👤 Scarletsky 📁 technology 👁 热度 56°C

简介

JSDOC 是一个 API 文档生成器，你只需要在代码中添加特定格式的注释，它就可以从注释中为你生成 HTML 文档。

安装

全局安装：

```
npm install -g jsdoc
```

如果你更倾向项目内使用，你也可以选择：

```
npm install jsdoc --save-dev
```

基本使用

使用 JSDOC 非常简单，先为 JavaScript 文件写好注释，然后用 JSDOC 去解析即可：

```
/path/to/jsdoc file1.js file2.js ...
```

但这种方式只适合少量文件时使用，当文件数量多了，再加上其他参数，维护起来就会非常麻烦。

所以更好的做法是编写一份配置文件 `jsdoc.json`，然后通过 `-c` 参数来指定：

```
/path/to/jsdoc -c jsdoc.json
```

无论 JSDOC 是通过全局安装，还是局部安装，都建议使用 npm scripts 来调用它，即在 `package.json` 的 `scripts` 中添加命令(这里命名成 `build:doc`，可以根据自己爱好定义其他名字)：

```
{
  "scripts": {
    "build:doc": "jsdoc -c jsdoc.json"
  }
}
```

文章

简介
安装
基本
配置
注释
标
命
(N
参考

了!

配置文件

JSDOC 的配置项非常多，最常用的是下面这些：

```
{
  "source": {
    "include": [ "src/" ],
    "exclude": [ "src/libs" ]
  },
  "opts": {
    "template": "node_modules/docdash",
    "encoding": "utf8",
    "destination": "./docs/",
    "recurse": true,
    "verbose": true
  }
}
```

source 表示传递给 JSDOC 的文件

source.include 表示 JSDOC 需要扫描哪些文件

source.exclude 表示 JSDOC 需要排除哪些文件

opts 表示传递给 JSDOC 的选项

opts.template 生成文档的模板，默认是 `templates/default`

opts.encoding 读取文件的编码，默认是 `utf8`

opts.destination 生成文档的路径，默认是 `./out/`

opts.recurse 运行时是否递归子目录

opts.verbose 运行时是否输出详细信息，默认是 `false`

注释

我们知道，JSDOC 的工作原理是通过分析 JavaScript 文件中的注释来生成 HTML 文档的。

但是，如果想 JSDOC 生成正确的结果，我们需要编写正确格式的注释才行。它接受如下格式的注释：

```
/**
 * @author Scarlex
 * @class
 * @name Application
 * @description Base Class of Application.
```

```
export default class Application {  
  /**  
   * @private  
   * @function  
   * @name Application#initialize  
   * @description Initialize the application.  
   */  
  initialize() {  
  }  
}
```

这是我们在 JavaScript 中常见的级块注释。

需要注意的是，JSDOC 的解析器要求注释必须以 `/**` 开头，如果是以 `/*`、`/***` 或多于三个星号的注释都会被忽略。

标签 (Tags)

有了这种级块注释，我们就可以在里面根据需要编写文档了。JSDOC 为我们提供了非常丰富的标签，它的解析器会对这些标签进行额外处理。这些标签大概可以分成两类：级块标签和行内标签。

级块标签：位于注释的最顶层。JSDOC 中绝大部分标签都是级块标签。

行内标签：位于级块标签内的标签，如 `@link`、`@tutorial`。

下面介绍一些常见的级块标签：

@author 该类/方法的作者。

@class 表示这是一个类。

@function/@method 表示这是一个函数/方法(这是同义词)。

@private 表示该类/方法是私有的，JSDOC 不会为其生成文档。

@name 该类/方法的名字。

@description 该类/方法的描述。

@param 该类/方法的参数，可重复定义。

@return 该类/方法的返回类型。

@link 创建超链接，生成文档时可以为链接到其他部分。

@example 创建例子。

命名路径 (Namepaths)

不知道你有没有发现，上面例子中是用 `Application#initialize` 来表示一个实例方法的。如果是静态方法，那应该怎么表示呢？JSDOC 有自

关闭

[博客](#) [分类](#) [标签](#) [关于](#) [RSS](#) [搜索](#)

了解了这些之后，我们就可以用 JSDOC 为 JavaScript 代码编写文档了，快去试试吧！

参考资料

<https://github.com/jsdoc3/jsdoc>

<http://usejsdoc.org/>



支持一下