
docstar

Release 0.1.0

Sep 27, 2017

Contents

1	Getting Started	3
2	Work Smarter, Not Harder	5
2.1	Synchronize package.json with docs config	5
3	Indices and tables	7

Docstar is a collection of tools to assist with build great looking docs for JavaScript libraries. These tools are designed to help bridge the gap between [Sphinx](#) and current JavaScript documentation tools.

The goal is to make documenting a library a well understood and repeatable process, which in turn makes it feel easier. The result being that having documentation that is easier to write and maintain, improves the quality of documentation have a net positive effect for the whole community.

CHAPTER 1

Getting Started

Getting started with DocStar is simple. Firstly install it using NPM:

```
npm install -g docstar
```

Once installed you should then be able to run it from the command-line. For example, running `docstar --help` should yield something like:

```
Usage: docstar [options]

Options:
  -h, --help            output usage information
  -V, --version          output the version number
  -p, --path [path]     Target output path (default to current directory)
```

OK, so given you have docstar installed and available, the first thing to do is to scaffold a sphinx docs directory for your project. The process I would recommend is:

1. Change to your node project folder (i.e. the folder that has a `package.json` for the project). DocStar takes care of a few things like keeping configs in sync through working with your `package.json` file.
2. Create a new boilerplate config with the following command:

```
docstar config create
```

If this is a new project, then a `docs` folder will be created, and a template `conf.py` file dropped into that folder. Additionally, the config will be synchronized with data (version, author, name) from your `package.json` file.

Work Smarter, Not Harder

Fact 1: I hate doing anything twice, that could have been done once and automated. **Fact 2:** See fact one.

It's not uncommon to have to duplicate information within your documentation that is already contained elsewhere in either your code or application meta-data. Docstar aims to reduce this duplication of information as much as possible, in a variety of ways.

Synchronize `package.json` with docs config

Certain information in your Sphinx generated documentation comes from placeholders that are defined in your `conf.py` file. In addition to being able to scaffold an initial `conf.py` file, docstar can also keep this file in sync with your `package.json` file.

Simply run the following command within your Node.js project directory:

```
docstar config sync
```

This will update particular pieces of information in configuration file. At this stage the following information is taken from your `package.json` file and placed in your `conf.py` file:

- package name => project
- package author(s) => copyright
- package version => version & release

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`