

Contributors

section-matter

Author

Jon Schlinkert

- [github/jonschlinkert](https://github.com/jonschlinkert)
- [twitter/jonschlinkert](https://twitter.com/jonschlinkert)

Like front-matter, but supports multiple sections in a document.

Please consider following this project's author, [Jon Schlinkert](#), and consider starring the project to show your :heart: and support.

License

Copyright © 2017, [Jon Schlinkert](#). Released under the [MIT License](#).

This file was generated by [verb-generate-readme](#), v0.6.0, on October 23, 2017.

Install

Install with [npm](#):

Usage

Params

`input {String|Buffer|Object}`: If input is an object, its content property must be a string or buffer.

`{Object}`: options

The following code:

```

This is the first section.

---  

title: One  

---  

More content.  

---  

You might also be interested in these projects:  


- assemble: Get the rocks out of your socks! Assemble makes you fast at creating web projects... more | homepage
- gray-matter: Parse front-matter from a string or file. Fast, reliable and easy to use. Parses YAML... more | homepage
- verb: Documentation generator for GitHub projects. Verb is extremely powerful, easy to use, and is used... more | homepage

  

Content before the sections.  

Given the following string:  

With the exception of front-matter, which must be the very first thing in the string, the opening delimiter of all other sections must be followed by a string to be used as the key for the section.  

$ npm install -g verb#dev verb  

To generate the readme, run the following command:  

in the verb.md readme template)  

edit the readme directly. Any changes to the readme must be made  

(This project's readme.md is generated by verb, please don't  

Building docs  

$ npm install npm test  

Running and reviewing unit tests is a great way to get  

familiarized with a library and its API. You can install  

dependencies and run tests with the following command:  

Running Tests  

Please read the contributing guide for advice on opening  

issues, pull requests, and coding standards.  

String and an array of Sections objects.  

• returns {Object}: Returns an object with a content  

var sections = require('');  

var result = sections(input, options);  

// { content: 'Content before sections', sections: [] }  

Example

```

- [assemble](#): Get the rocks out of your socks! Assemble makes you fast at creating web projects... [more](#) | [homepage](#)
- [gray-matter](#): Parse front-matter from a string or file. Fast, reliable and easy to use. Parses YAML... [more](#) | [homepage](#)
- [verb](#): Documentation generator for GitHub projects. Verb is extremely powerful, easy to use, and is used... [more](#) | [homepage](#)

Related projects

```

var input = '~~~\ntitle:
    bar\n~~~\n\nfoo\nn~~~one\nntitle:
    One\n~~~\nThis is one';
console.log(sections(input,
    {section_delimiter: '~~~'}));

```

Results in:

```

{
  content: '',
  sections: [
    {
      key: '',
      data: 'title: bar',
      content: '\nfoo'
    },
    {
      key: 'one',
      data: 'title: One',
      content: 'This is one'
    }
  ]
}

```

About

Contributing

Pull requests and stars are always welcome. For bugs and feature requests, [please create an issue](#).

```

console.log(sections(input));

Results in:

{
  content: 'Content before the sections.
  \n\n---\n\nMore content.\n',
  sections: [
    {
      key: 'one',
      data: 'title: One',
      content: '\nThis is the first
      section.'
    }
  ]
}

```

Options

`options.section_parse`

Type: function

Default: undefined

Function to be called on each section after it's parsed from the string.

Example

Given the following string (`foo.md`):

```

This is content before the sections.

---one
title: First section
sections: [
    content: "This is content before the
    sections.\n",
    key: "section-one",
    data: { title: "First section" },
    content: "\nThis is section one."
]
---two
title: Second section
sections: [
    content: "This is content before the
    sections.\n",
    key: "section-two",
    data: { title: "Second section" },
    content: "\nThis is section two."
]
Using the following custom section_parse function:

var fs = require('fs');
var path = require('path');
var yaml = require('js-yaml');
var sections = require('section');

var section_parse = function(section) {
    var options = {
        section_key = 'section-' + section.key,
        section_data = section.data,
        section_log = console.log(section)
    };
    yaml.safeLoad(section.data);
    section.log(options);
    return options;
};

var str = fs.readFileSync('foo.md');
var options = {
    section_parse: section_parse
};
sections(str, options);
console.log(result);

```

Results in:

```

This is content before the sections.

---one
title: First section
sections: [
    content: "This is content before the
    sections.\n",
    key: "section-one",
    data: { title: "First section" },
    content: "\nThis is section one."
]
---two
title: Second section
sections: [
    content: "This is content before the
    sections.\n",
    key: "section-two",
    data: { title: "Second section" },
    content: "\nThis is section two."
]

```

options.section_delimiter

Type: string
Default: ---

Delimiter to use as the separator for sections. With the exception of front-matter, which must be the very first thing in the string, the opening delimiter of all other sections must be followed by a string to be used as the key for the section.

Example