Yamp

by @angrykoala

> Yet Another Markdown Parser

The aim of this package is to provide an easy-to-use toolbox for markdown-related task including Html & Pdf conversion.

- **GitHub:** https://github.com/angrykoala/yamp
- **Npm:** https://www.npmjs.com/package/yamp

Features

- HTML conversion
- PDF conversion
- Code highlight support
- Github-style and academic output
- API to use *yamp* programmatically
- Custom styles
- CSS-embedded HTML (just open it offline in any browser)
- HTML tags support (for PDF output too)
- Include other files in your markdown
- HTML presentations
- Front Matter metadata
- Custom xejs-based tags (`{{}}`)
- Koalafied

Upcoming features

- Custom templates
- Client-side web support (browserify)

> Check the project roadmap and our cute kanban board

Installation

To use *yamp* cli, install it globally using **npm**:

```
npm install -g yamp
```

If you want to use the API instead, install it locally:

```
npm install --save yamp
```

then, include yamp in your javascript:

```
var yamp = require('yamp');
```

Usage

To create a `.pdf` file from your *markdown* file, simply type:

```
yamp <file.md>
```

For example:

```
yamp README.md
```

Will generate `readme.pdf`.

Options

- `-h`, `--help` to display a basic usage information
- `-V`, `--version` to display *yamp* version installed
- `-o`, `--output <file>` output filename (without extension) e.g. `yamp my_file.md -o final_name`
- `--pdf` to generate a pdf (default)
- `--html` to generate html
- `--remark` to generate a html presentation using remark
- `-t`, `--title [value]` to add a custom title to Html pages
- `--style <file>` to set change the css style (supports the provided styles and custom styles)
  - Option not suported along with `--no-style`
- `--no-style` to disable CSS styling
  - Option not supported along with `--style <file>`
- `--list-styles` will list all the styles provided by yamp
  - These styles will be supported by `--style` option
- `--minify` to minify Html output
- `--no-tags` to disable custom Yamp tags
- `--no-highlight` to disable code highlight
- `--no-front-matter` to disable front-matter metadata
- `-k`, `--koala` to koalify your outputs

To generate pdf and html with default styling and options:

```
yamp myFile.md --pdf --html
```

> The `--no-highlight` and `--no-style` options will greatly reduce your Html and Pdf outputs

Yamp tags

Yamp supports extra tags in your markdown files. Currently using xejs templates. All tags are written between double braces `{{ ... }}` and are not case-sensitive

- `include [file.md]` : Includes the given text file (markdown or not), the tags on the included file will also be parsed, allowing nested file structure.
- `date` : Will write the current date (at the moment of rendering).
- `page break` : Will force a page break in pdf output.
- `yamp version` : Will display the yamp version used to render the document.

> Starting a tag with `{{#` will create a comment tag that will not be rendered into the final file

Yamp styles

Yamp provides several styles for your document (supported for html and pdf outputs).

- `github.css` Default style, will look similar to **Github** style
- `acm-sig.css` Academic style based on ACM SIG templates

You can select any of these styles with the option `--style [style.css]`, the same option will enable you to use your own files instead `--style [myfolder/mystyle.css]`

> You can always check the styles with the option `--list-styles`

API

Include *yamp* in your javascript with:

```
var yamp = require('yamp');
```

You'll have access to different *renderers* to process your files:

- `yamp.renderers.html` to process a markdown file into an full Html page
- `yamp.renderers.pdf` to process a markdown into a pdf

To use a renderer:

```
var myRenderer = new renderers.pdf(options);
renderer.renderFile(myFile, function(err){
    if (err) return console.log("Error while rendering: "+err);
    else console.log("Rendering was successful");
});
```

Options

The options accepted by the default renderers are:

- **outputFilename**: name of the output filename (without extension), will default to the input filename
- **highlight**: (*true*) indicates if code blocks should be highlighted
- **style**: (*true*) indicates if default style should be used or no style at all. If a filename is passed, it will use it as custom css style
- **minify**: (*false*) whether the Html output should be minified or not
- **title**: Custom title for the Html page
- **tags**: (*true*) whether to parse yamp tags or not (`{{ ... }}`)
- **koala**: (*false*) true to koalify your outputs

Creating new renderers

If you need a custom renderer, instead of using one of the defaults you can extend directly from **Renderer** class or any of the default renderers:

```
class MyCustomRenderer extends yamp.Renderer {
    constructor(options) {
        super(options, "default.ejs", yamp.parsers.md2Html);
        this.output="html"; //desired output extension
    }
```

```
    beforeLoad(filename){
        //Modify filename or this.fileLoader before loading it
    }


    beforeRender(templateOptions) {
        // Modify the data passed to the template before rendering, inc
luding title, content and options
    }

    afterRender(content) {
        // Modify template result (Html)
    }

    fileOutput(content,done) {
        // Write file (preferably to this.options.outputFilename) in th
e desired format using a parser
    }
}
```

Custom parser: It is possible to use a custom parser from markdown to Html instead of the built-in *yamp.parsers.md2html*, the parser must be a function of the type `function(originalString,options,callback)` that will translate from `originalString` (markdown) to html, calling the `callback(err,res)` afterwards.

If, instead of extending from `yamp.Renderer` you are extending from one of the default renderers, you should only re-implement the methods you need, and usually you should call `super().methodName` to maintain its basic functionality.

Development Instructions

To contribute to **yamp** you should clone the official repository https://github.com/angrykoala/yamp or your own *fork* with `git`.

You can also download it from GitHub clicking here

- To install execute `npm install` in the downloaded/cloned folder
- To test, execute `npm test`

    - The tests will also run *jshint*
- To execute the CLI, execute `npm start -- <file> [options]`
- To install your local version globally, execute `npm install -g .` on the project folder
- To generate documentation (with installed version of *yamp*) execute `npm run docs`

> It is strongly recommended to install the npm repository version instead of your local copy

Contributors

If you want to contribute to yamp please:

1. Read CONTRIBUTING.md
2. Fork from dev branch
3. Make sure tests passes before pull request
4. Check the opened and closed issues before creating one

Thanks for your help!

Acknowledgments

- Marked as markdown parser
- Github-markdown.css as default style
- Highlight.js for code highlighting
- html-pdf for pdf generation
- remark for html slides output
- pubcss for academic output style

> YAMP is developed under GNU GPL-3 license by @angrykoala