PLUGINS

Here's a list of all available DocPad plugins. If you've created a plugin, be sure to include it in this listing! :)

INSTALLING

To install a plugin, run `docpad install #{thePluginName}` inside your project directory. As an example, to install the eco plugin, you would run `docpad install eco`

To uninstall a plugin, run `docpad uninstall #{thePluginName}` inside your project directory.

In older versions of DocPad, you would run `npm install --save docpad-plugin-#{thePluginName}` to install, and `npm uninstall --save docpad-plugin-#{thePluginName}` to uninstall.

RENDERERS

These are plugins that add support for extra markups and languages to DocPad:

WRITING MARKUPS

- marked - Supports Markdown to HTML `.html.(md|markdown)` viamarked
- multimarkdown - Supports MultiMarkdown to HTML `.html.(md|markdown)` via node-multimarkdown
- orgmode - Supports converting org-mode to HTML `.html.org`

- textile - Supports Textile to HTML `.html.textile`

DATA MARKUPS

- cson - Supports CSON to JSON `.json.cson`
- yaml - Supports YAML to JSON `.json.(yaml|yml)` and JSON to YAML `.(yml|yaml).json`

CSS PRE-PROCESSORS

- less - Supports LessCSS to CSS `.css.less`
- roole - Supports Roole to CSS `.css.roo`
- sass - Supports SCSS and SASS to CSS (includes compass support) `.css.(sass|scss)`
- nodesass - Supports SCSS to CSS (using node-sass) `.css.scss`
- styl - Supports Styl to CSS `.css.styl`
- stylus - Supports Stylus to CSS `.css.(styl|stylus)`

JAVASCRIPT PRE-PROCESSORS

- coffeescript - Supports CoffeeScript to JavaScript `.js.coffee`
- js2coffee - Supports JavaScript to CoffeeScript `.coffee.js`
- livescript - Supports LiveScript to JavaScript `.js.ls`
- move - Supports Move to JavaScript `.js.move`
- typescript - Supports TypeScript to JavaScript `.js.ts`

HTML PRE-PROCESSORS

COFFEESCRIPT TEMPLATING ENGINES

- eco - Supports Eco to anything `.anything.eco`

- coffeekup - Supports CoffeeKup to anything `.anything.coffee`
- html2coffee - Supports HTML to CoffeeKup `.coffee.html`
- coffeemugg - Supports CoffeeMugg to anything `.anything.coffee`
- hamlcoffee - Supports Haml Coffee to HTML `.html.hamlc`
- teacup - Supports Teacup to HTML `.html.coffee`

HAML-LIKE TEMPLATING ENGINES

- haml - Supports Haml to anything `.anything.haml`
- jade - Supports Jade to anything `.anything.jade`
- html2jade - Supports HTML to Jade `.jade.html`

MOUSTACHE TEMPLATING ENGINES

- handlebars - Supports Handlebars/Moustache to anything `.anything.(hb|hbs|handlebars)`
- hogan - Supports Hogan/Mustache to anything `.anything.hogan`

OTHER TEMPLATING ENGINES

- consolidate - Supports many template engines via Consolidate.js
- slim - Supports Slim to anything `.anything.slim`
- swig - Supports Swig to HTML `.html.swig`
- htmlmin - Supports minifying HTML with HTML-Minifier `.html.anything`
- vash - Supports Vash, an implementation of the Razor template syntax for JavaScript, to anything `.anything.vash`

LANGUAGES

- cmds - Adds some shell scripting fun. Supports `.anything.bash|.anything.sh|.anything.ps1|.anything.cmd`
- php - Supports PHP to anything `.anything.(php|phtml)`
- ruby - Supports Ruby and ERuby to anything `.anything.(ruby|erb)`

HELPERS

These are plugins that add extra functionality to DocPad:

- addthis - Adds the AddThis toolbar into your project
- alias - Adds configurable hard or soft aliases for Docpad documents.
- assets - Change URL of asset files to contain hash of contents, allowing for effective caching whilst enabling cache busting when contents change
- associatedfiles - Lets you easily associate files to a particular document, and then grab the collection for them
- basicauth - Adds basic authentication to your project
- browserifybundles - Add configuration to your DocPad configuration file to create browserify bundles of your scripts
- browserifydocs - Browserify your documents by adding `browserify: true` to their meta data
- buildr - Supports bundling scripts and styles (including pre-processors like CoffeeScript, LESS, etc.) using Buildr
- cachr - Allows you to cache remote URLs locally from within your templates
- cleancss - Concatinate and minify CSS files with the `cleancss: true` meta data
- cleanurls - Adds support for URLs like `/blog/hello` as well as the original URL `/blog/hello.html`
- coffeelint - Prints coffeelint errors to the console
- csv - Adds support for CSV data mapping. The comma seperated data files work just like a database, map from column 1 to column 2
- datefromfilename - Automaticaly set the `date` meta-data property by determining it from the document's filename
- dateurls - Adds support for date-based URLs like `/2013/04/27/hello.html`
- facebookcomments - Adds the Facebook Comment Widget to your project
- feedr - Allows you to render remote feeds within your templates
- frontend - CSS and JavaScript asset manager and compiler for DocPad
- gist - Pulls in gists into your document
- grunt - Run Grunt.js tasks when building with DocPad
- heapdumper - Generates a heapdump snapshot for chosen DocPad event(s), viewable in the Chrome profiler
- highlightjs - Adds Highlight.js syntax highlighting to code snippets
- ignoreincludes - Avoid writing include files to the `/out` directory
- isexc - Adds the ability to apply AOT compilation to JavaScript files

- jsonfragment - Writes each documents content without layout and its meta data into a separate `.json` file for quick loading via AJAX.
- livereload - Automatically reloads the page whenever a regeneration is performed
- lunr - Client-side full-text and faceted search using Lunr.js
- menu - Automatically generates menu from `/render` folder
- moment - Date formatting and access to Moment.js library
- nativecomments - Adds support for native comments to DocPad
- navlinks - Adds the ability to generate a navigation bar for documents with links to the next and previous document of a specified collection.
- paged - Adds multiple page support to documents allowing you to render one document out to many pages
- partials - Adds the ability to create re-usable partials for your templates within DocPad
- pygments - Adds Pygments syntax highlighting to code snippets
- raw - Copies all files in the `/raw` directory to `/out` without going through DocPad's generation process. Useful for files that cause out of memory/speed issues when placed in `/static` directory.
- copy - Alternative to raw pluging with performace optimizations. Copies all files in the `/raw` directory to `/out` without going through DocPad's generation process. Useful for files that cause out of memory/speed issues when placed in `/static` directory.
- react - Renders markup for React Components
- redirector - Creats redirects (301 or meta-refresh) via configuration.
- related - Scans your documents `tags: 'tag1', 'tag2'` metadata to produce a listing of related documents
- rss - Generates an RSS feed for a configurable collection
- scheduling - Schedules content so that it is not rendered out before the `date` specified in the content's meta-data.
- services - Adds support for many 3rd party services to DocPad
- shortcodes - Adds various Wordpress style shortcodes (e.g., `[video id="123"]`) to simplify template writing.
- sitemap - Generates a `sitemap.xml` file for your site from the `html` documents collection
- tableofcontents - Automatically generate table of contents
- text - Render `templateData` properties without needing template engine, useful for abstraction in configuration files
- thumbnails - Manages thumbnail generation of your image files
- imagin - Alternative to thumbails plugin with support for `raw` and `copy` plugins for performance optimization. Manages thumbnail

generation of your image files.
- tinylivereload - A LiveReload plugin that doesn't alter your HTML. Works with the Chrome/Firefox LiveReload extensions.
- uglify - Compress and minify JavaScript files with the `uglify: true` meta data
- umd - Wrap specified JavaScript documents in the Universal Module Definition (UMD) allowing them to run in AMD, Require.js, CommonJS/Node.js and Vanilla environments automatically
- sanitizer - A helper for HTML string sanitization based on Google Caja

DEPLOYERS

These are plugins that make deploying to particular services even easier:

- ghpages - Deploy to GitHub Pages as easy as `docpad deploy-ghpages`
- sunny - Uploads site to cloud (AWS, Google Storage) after generation

ADMIN INTERFACES

DocPad's plan from the very beginning has been to be interface agnostic. This means that we will be able to utilise existing interfaces, customer interfaces, and decoupled interfaces. Allowing us to always utilise the best experiences for everyone involved.

EXISTING INTERFACES

DocPad's plugin/extension infrastructure supports existing coupled interfaces by importing their data directly into the DocPad Database. So if you love using Tumblr, WordPress, Medium, MongoDB, or GitHub repos for your content, you don't have to give them up. Just install the importer plugin for them, and DocPad will import the data from that service into the DocPad database for rendering.

- downloader - Download (and optionally extract) files into your project, used in the Bootstrap Skeleton to pull in Bootstrap
- repocloner - Clone repos into your project, awesome for creating

- tumblr - Imports Tumblr data directly into your DocPad Database, used in the Syte Skeleton to pull in Tumblr data
- mongodb - Imports collections from MongoDB.

CUSTOM INTERFACES

DocPad's plugin/extension infrastructure supports custom Admin Interfaces tightly coupled to the DocPad experience. So far we have the following extensions that add Admin Interfaces to DocPad:

- DocPad Collections Editor - A simple WYSIWYG editor for DocPad Collections
- MiniCMS - Adds an admin interface to DocPad
- CMS Docpad - Lightweight CMS UI with WYSIWYG editor for pages and parts

DECOUPLED INTERFACES

DocPad's plugin/extension infrastructure supports existing decoupled interfaces by providing plugin/extension adapters to the interface allowing the interface to interact directly with the DocPad Database, or theoretically any backend providing an interface was made for it. So far we have the following extensions that add Decoupled Interfaces to DocPad:

- Use Prose with DocPad to create a Wiki - Tutorial on how to useProse.io as an Admin Interface for DocPad
- WebWrite's InlineGUI (not yet ready) - Edit your content from any backend with this inline editing interface
- Edit & Deploy with GitHub.com & GHpages Plugin - Wiki on how to use GitHub.com to edit your DocPad website and then DocPad's `ghpages` plugin to deploy to GitHub Pages.

GUIDES

These are miscellaneous things that you can do with DocPad:

- Localising and formatting dates
- Automatically set custom meta data for collections
- Absolute URL Helper

- Sitemap Generation
- Concatenate your scripts with Browserify
- Minify your assets with Grunt
- Minify your assets automatically post-deployment with Cloudflare
- Custom Routing
- Paging Solutions
- Responsive Layouts in Stylus
- Thoughts on a DocPad GUI
- Getting Ruby, SASS and DocPad working on Heroku
- Respond with JSON when asked to
- Require authentication to view certain documents
- Use DocPad and GitHub as a Wiki

COMPLETE PLUGIN LISTING

You can find a complete listing of all DocPad Plugins on the NPM Registry using the `docpad-plugin` keyword. Though note, this listing is not curated by the DocPad Team, so be careful.

CREATE YOUR OWN!

It's easy to write plugin for DocPad. Get started now on our Write a Plugin Page!

REQUESTED

Here is a list of plugins waiting to be coded up :-)

DOCPAD IS A **BEVRY** CREATION.

DOCPAD **GITHUB** **SUPPORT**