CONFIGURATION

CONFIGURATION FILES

The DocPad configuration file sits within the root of your DocPad project and is named as one of the following. Each name provides a special meaning. Here are the valid names:

- `docpad.js` a Node JavaScript file, will generally look like:

  `module.exports = {/*the configuration*/}`

- `docpad.json` a JSON file, does not allow functions, will generally look like: `{/*the configuration*/}`

- `docpad.coffee` a Node CoffeeScript file, will generally look like:

  `module.exports = /*the configuration*/`

- `docpad.cson` a cson file, will generally look like:

  `/*the configuration*/`

The advantage of `docpad.js` and `docpad.coffee` over `docpad.json` and `docpad.cson` is that they allow us to declare functions, as well as call functions. However, for instances where we cannot trust the contents of the configuration files you would want to use the `docpad.json` or `docpad.cson` as they can't do anything naughty.

The advantage of `docpad.coffee` and `docpad.cson` over `docpad.js` and `docpad.json` is that they allow us to use the CoffeeScript syntax which is a lot more lenient.

Generally, you'll usually find either a `docpad.coffee` file or a `docpad.cson` file.

AVAILABLE CONFIGURATION

The following code is of a `docpad.coffee` file that contains every single configurable DocPad options - in reality you should only include the options that you have actually customised - using the following example for reference only.

```coffee
# DocPad ConfigurationdocpadConfig = {    # =========================
=========    # Paths Configuration    # Root Path    # The root path
 of our our project    rootPath: process.cwd()  # default    # Packa
ge Path    # The project's package.json path    # If it is a relativ
e path, it will have the resolved `rootPath` prepended to it    pack
agePath: 'package.json'  # default    # Config Paths    # An array o
f paths that we try to extract our docpad configuration from    conf
igPaths: [  # default        'docpad.js'        'docpad.coffee'
   'docpad.json'        'docpad.cson'    ]    # Plugin Paths    # An
 array of special paths which to load single plugins from    # (e.g.
, ['/a/path/to/a/docpad-plugin-special'])    pluginPaths: []  # defa
ult    # Plugins Paths    # An array of paths which to load multiple
 plugins from    pluginsPaths: [  # default        'node_modules'
    'plugins'    ]    # Reload Paths    # An array of special paths
 that when changes occur in, we reload our configuration    reloadPa
ths: []  # default    # Regenerate Paths    # An array of special pa
ths that when changes occur in, we regenerate our website    regener
atePaths: []  # default    # Regenerate Delay    # The time (in mill
iseconds) to wait after a source file has    # changed before using
it to regenerate. Updating over the    # network (e.g., via FTP) can
 cause a page to be partially    # rendered as the page is regenerat
ed *before* the source file    # has completed updating: in this cas
e increase this value.    regenerateDelay: 100    # default    # Out
 Path    # Where should we put our generated website files?    # If
it is a relative path, it will have the resolved `rootPath` prepende
d to it    outPath: 'out'  # default    # Src Path    # Where can we
 find our source website files?    # If it is a relative path, it wi
ll have the resolved `rootPath` prepended to it    srcPath: 'src'  #
 default    # Documents Paths    # An array of paths which contents
will be treated as documents    # If it is a relative path, it will
have the resolved `srcPath` prepended to it    documentsPaths: [  #
default        'render'        'documents'    ]    # Files Paths
# An array of paths which contents will be treated as files    # If
it is a relative path, it will have the resolved `srcPath` prepended
 to it    filesPaths: [  # default        'static'        'files'
    'public'    ]    # Layouts Paths    # An array of paths which c
ontents will be treated as layouts    # If it is a relative path, it
 will have the resolved `srcPath` prepended to it    layoutsPaths: [
   # default        'layouts'    ]    # Ignore Paths    # Can be set
to an array of absolute paths that we should ignore from the scannin
g process    ignorePaths: false  # default    # Ignore Hidden Files
    # Whether or not we should ignore files that start with a dot fro
m the scanning process    ignoreHiddenFiles: false  # default    # I
gnore Common Patterns    # Whether or not we should ignore commonly
undesired files from the scanning process    # (e.g., .DStore, thumb
s.db, .git, files that start with a tilda, etc.)    ignoreCommonPatt
erns: true  # default    # Ignore Custom Patterns    # Can be set to
```

```
gnoreCustomPatterns: false  # default    # ========================
========    # Server Configuration    # Port    # Use to change the
port that DocPad listens to    # By default we will detect the appro
priate port number for our environment    # if no environment port n
umber is detected we will use 9778 as the port number    # Checked e
nvironment variables are:    # - PORT - Heroku, Nodejitsu, Custom
 # - VCAP_APP_PORT - AppFog    # - VMC_APP_PORT - CloudFoundry    po
rt: null  # default    # Max Age    # The default caching time limit
 that is sent with the response to the client    # Can be set to `fa
lse` to disable caching    maxAge: 86400000  # default    # Server
Express    # The Express.js instance that we want DocPad to use    #
 If not set, we will create our own    serverExpress: null  # defaul
t    # Server HTTP    # The Node.js HTTP server instance that we wan
t DocPad to use    # If not set, we will create our own    serverHtt
p: null  # default    # ===================================    # Middl
eware Configuration    # Which middlewares would you like us to acti
vate    # Setting `extendServer` to `false` will set all of these to
 `false` automatically    # Extend Server    # Whether or not we sho
uld extend the server with our custom middleware    extendServer: tr
ue  # default    # The standard middlewares (bodeParser, methodOverr
ide, Express router)    middlewareStandard: true  # default    # The
 standard bodyParser middleware    middlewareBodyParser: true  # def
ault    # The standard methodOverride middleware    middlewareMethod
Override: true  # default    # The standard Express router middlewar
e    middlewareExpressRouter: true  # default    # Our own 404 middl
eware    middleware404: true  # default    # Our own 500 middleware
    middleware500: true  # default    # ==============================
====    # Logging Configuration    # Log Level    # Up to which leve
l of logging should we output    logLevel: (if ('-d' in process.argv
) then 7 else 6)  # default    # Logger    # The caterpillar instan
ce that we want to use    # If not set, we will create our own    lo
gger: null  # default    # Growl    # Whether or not we should displ
ay system notifications as things progress within DocPad    growl: t
rue  # default    # Catch Exceptions    # Whether or not DocPad shou
ld catch uncaught exceptions    catchExceptions: true  # default
# Report Errors    # Whether or not we should report errors back to
the DocPad Team    reportErrors: process.argv.join('').indexOf('test
') is -1  # default (don't enable if we are running inside a test)
  # Report Statistics    # Whether or not we should report statistic
s back to the DocPad Team    reportStatistics: process.argv.join('')
.indexOf('test') is -1  # default (don't enable if we are running in
side a test)    # Airbrake Token    # The airbrake token we should u
se for reporting errors    # By default, uses the DocPad Team's toke
n    airbrakeToken: null  # default    # MixPanel Token    # The mix
panel token we should use for reporting statistics    # By default,
uses the DocPad Team's token    mixpanelToken: null  # default    #
==============================    # Other Configuration    # Dete
ct Encoding    # Should we attempt to auto detect the encoding of ou
r files?    # Useful when you are using foreign encoding (e.g., GBK)
 for your files    # Only works on unix systems currently (limit of
iconv module)    detectEncoding: false    # Render Single Extensions
    # Whether or not we should render single extensions by default
  renderSingleExtensions: false  # default    # Render Passes    # H
ow many times should we render documents that reference other docume
nts?    renderPasses: 1  # default    # Check Version    # Whether o
```

```coffee
# default    # Welcome    # Whether or not we should display any custom welcome callbacks    welcome: false # default    # Prompts    # Whether or not we should display any prompts    prompts: false # default    # Helper Url    # Used for subscribing to newsletter, account information, and statistics, etc.    helperUrl: 'https://docpad.org/helper/' # default    # Safe Mode    # If enabled, we will try our best to sandbox our template rendering so that they cannot modify things outside of them    # Not yet implemented    safeMode: false    # default    # Collections    # A hash of functions that create collections    collections: {} # default    # Regenerate Every    # Performs a regenerate every x milliseconds, useful for always having the latest data    regenerateEvery: false # default    # ====================================    # Template Configuration    # Template Data    # Use to define your own template data and helpers that will be accessible to your templates    # Complete listing of default values can be found here: http://docpad.org/docs/template-data    templateData: # example    # Specify some site properties    site:    # The production URL of our website    url: "http://website.com"    # The default title of our website    title: "Your Website"    # The website description (for SEO)    description: """    When your website appears in search results in say Google, the text here will be shown underneath your website's title.    """    # The website keywords (for SEO) separated by commas    keywords: """    place, your, website, keywoards, here, keep, them, related, to, the, content, of, your, website    """    # -----------------------------    # Helper Functions    # Get the prepared site/document title    # Often we would like to specify particular formatting to our page's title    # we can apply that formatting here    getPreparedTitle: ->    # if we have a document title, then we should use that and suffix the site's title onto it    if @document.title    "#{@document.title} | #{@site.title}"    # if our document does not have its own title, then we should just use the site's title    else    @site.title    # Get the prepared site/document description    getPreparedDescription: ->    # if we have a document description, then we should use that, otherwise use the site's description    @document.description or @site.description    # Get the prepared site/document keywords    getPreparedKeywords: ->    # Merge the document keywords with the site keywords    @site.keywords.concat(@document.keywords or []).join(', ')    # ================================    # Plugin Configuration    # Skip Unsupported Plugins    # Set to `false` to load all plugins whether or not they are compatible with our DocPad version or not    skipUnsupportedPlugins: true    # default    # Enable Unlisted Plugins    # Set to false to only enable plugins that have been explicity set to `true` inside `enabledPlugins`    enabledUnlistedPlugins: true # default    # Enabled Plugins    enabledPlugins: # example    # Disable the Pokemon Plugin    pokemon: false    # Enable the Digimon Plugin    # Unless, enableUnlistedPlugins is set to false, all plugins are enabled by default    digimon: true    # Configure Plugins    # Should contain the plugin short names on the left, and the configuration to pass the plugin on the right    plugins: # example    # Disable NIB within the Stylus Plugin    stylus:    useNib: fals
```

```
      # Events    # Allows us to bind listeners to the events that DocPad
    emits    # Complete event listing can be found here: http://docpad.
    org/docs/events    events: # example        # Server Extend
    # Used to add our own custom routes to the server before the docpad
    routes are added        serverExtend: (opts) ->            # Extract
     the server from the options            {server} = opts            d
    ocpad = @docpad    # ==============================    # Environm
    ent Configuration    # Locale Code    # The code we shall use for ou
    r locale (e.g., `en`, `fr`, etc.)    # If not set, we will attempt t
    o detect the system's locale, if the locale can't be detected or if
    our locale file is not found for it, we will revert to `en`    local
    eCode: null # default    # Environment    # Which environment we sh
    ould load up    # If not set, we will default the `NODE_ENV` environ
    ment variable, if that isn't set, we will default to `development`
      env: null # default    # Environments    # Allows us to set custo
    m configuration for specific environments    environments:  # defaul
    t        development: # default            # Always refresh from se
    rver            maxAge: false # default            # Only do these
    if we are running standalone via the `docpad` executable
    checkVersion: process.argv.length >= 2 and /docpad$/.test(process.ar
    gv[1])  # default            welcome: process.argv.length >= 2 and /
    docpad$/.test(process.argv[1])  # default            prompts: proces
    s.argv.length >= 2 and /docpad$/.test(process.argv[1])  # default
          # Listen to port 9005 on the development environment
        port: 9005  # example}# Export the DocPad Configurationmodule.e
    xports = docpadConfig
```

ENVIRONMENT CONFIGURATION FILE

We also support `.env` environment configuration file, the format works like so:

```
KEY=VALUEKEY2=VALUE2
```

All key-value pairs are added to the `process.env` environment variable.

This is useful for setting sensitive information, like API keys, database information, and so on.

If you're using `.env` using for this purpose, be sure to add `.env` to your `.gitignore` file!

GLOBAL CONFIGURATION FILE

DocPad will also create a global configuration file located at
`~/.docpad.cson` (or `~/Dropbox/.docpad.cson` if you
have Dropbox installed). The `~` indicates your home directory (e.g.,
`/Users/balupton`), not your project directory.

This file is typically used to store information relating to your DocPad
preferences, such as whether or not you have agreed to the TOS and
subscribed to our newsletter or not, however it can also be used to load in
configuration that you would like applied to all of your projects. Normally, we
wouldn't touch this file, however it is useful to on the odd occasion.

For instance, if we wanted to set the default DocPad port to always be `9779`
instead of `9778` , we can update the global configuration file to include:

```
{    port: 9779}
```

COMMAND LINE   **DOCUMENTATION**   **DEBUGGING**
INTERFACE

Edit and improve this page!

DOCPAD IS A **BEVRY** CREATION.

DOCPAD   GITHUB   SUPPORT