Pylibsass Documentation

Release 0.1.4

Ryan Senkbeil

June 27, 2013

CONTENTS

Release v0.1.4. (*Installation*)

Pylibsass is a simple to use wrapper around the libsass project. It is released under the MIT license.

```
>>> import pylibsass
>>> pylibsass.watch("app/static/scss", "app/static/css")
```

CHAPTER

ONE

USER GUIDE

1.1 Installation

Installation of pylibsass is easy. It comes with the libsass source code, so you will need to have the proper compilers installed before using.

1.1.1 Pip/Distribute

To install pylibsass, either use pip or add *pylibsass* to your requirements.txt file:

```
$ pip install pylibsass
```

If you must, you can also use *easy_install*:

```
$ easy_install pylibsass
```

1.1.2 Checking Out The Code

The home base for pylibsass is on GitHub.

To check out the repository:

```
$ git clone https://github.com/rsenk330/pylibsass.git
```

Once you have a copy of the code, you can install it:

```
$ python setup.py install
```

1.2 Usage

This part of the documentation will give a high level overview of the different functions that pylibsass provides.

1.2.1 Libsass API

One of the goals of pylibsass if to provide a wrapper around the few functions that the libsass API provides. Currently, it only implements one of the functions, *compile()*:

```
>>> import pylibsass.sass as sass
>>> sass.compile(".test { color: red; }")
```

Not all of the exposed libsass API functions are exposed yet, but that is coming soon.

1.2.2 Helpers

Since you probably don’t want to be detecting changes to files, reading them, calling the *compile()* function, and then writing them out to a CSS file, pylibsass leverages the Watchdog library to handle that for you:

```
>>> import pylibsass
>>> pylibsass.watch("app/static/scss", "app/static/css")
```

Any changes in the source directory (*app/static/scss*) will be automatically compiled to the destination directory (*app/static/css*).

CHAPTER

TWO

API

2.1 Pylibsass API

This section of the documentation provides an introduction to all of the pylibsass interface. All of the important stuff should be here. If there is something missing, the source code is available on *GitHub <https://github.com/rsenk330/pylibsass*.

2.1.1 Main Interface

The major functions will be exposed directly through the main *pylibsass* package.

`pylibsass.`**`watch`**`(`*source_path*, *dest_path*`)`

Watches for filesystem changes and compiles SCSS to CSS

This is an async function.

Parameters

- **source_path** (*str*) – The source directory to watch. All .scss files are monitored.
- **dest_path** (*str*) – The destination directory where the resulting CSS files should go. The filename will be identical to the original, except the extension will be *.css*.

2.1.2 Lower-Level Interface

The lower-level interface is used by the main interface. If you need to do something weird, these functions will be useful. These typically interact directly with libsass.

`pylibsass.sass.`**`compile_str`**`(`*contents*`)`

Compiles a SASS string

Parameters **contents** (*str*) – The SASS contents to compile

Returns The compiled CSS

class `pylibsass.sass.`**`LibSass`**

Wrapper class around libsass.

The class provides methods that mimic the functions in sass_interface.h

`compile``(`*ctx*`)`

Compiles a string to CSS

This calls the *sass_compile* function.

:param `pylibsass.sass.SassContext` ctx: The libsass context

`compile_file`(*ctx*)

Compiles a file to CSS

This calls the *sass_compile_file* function.

:param `pylibsass.sass.SassFileContext` ctx: The libsass context

`compile_folder`(*ctx*)

Compiles a folder to CSS

This calls the *sass_compile_folder* function.

:param `pylibsass.sass.SassFolderContext` ctx: The libsass context

`sass_new_context`()

Gets a new sass context object.

This calls the *sass_new_context()* function.

`sass_new_file_context`()

Gets a new sass file context object.

This calls the *sass_new_file_context()* function.

`sass_new_folder_context`()

Gets a new sass folder context object.

This calls the *sass_new_folder_context()* function.

2.1.3 C Structs

The C structs are ctypes definitions for the libsass structs. See the libsass project documentation for more information.

class `pylibsass.sass.`**`SassContext`**

class `pylibsass.sass.`**`SassFileContext`**

class `pylibsass.sass.`**`SassFolderContext`**

CHAPTER

THREE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX