
Appie Documentation

Release 1.0

Arnaud Loonstra

February 16, 2016

1	Installation	3
1.1	Appie module reference	3
2	Indices and tables	5
	Python Module Index	7

Appie is a simple static generator for dynamic websites. Dynamic websites? Yes! Although Appie can generate HTML files its focus is on providing all information of files residing in your website's directory. As a HTML file can not see what's available on your webserver Appie will provide a JSON file containing exactly this. From this the HTML file, using javascript, can figure things out. By matching to certain filenames the JSON file can be filled with specific data. For example a file with the .textile extension is parsed into HTML. Same holds for files ending with a .md extension.

Installation

Use pip to install Appie:

```
pip install https://github.com/sphaero/appie/archive/master.zip
```

Appie is developed for Python3!

Contents:

1.1 Appie module reference

1.1.1 Appie main class

```
class appie.Appie(*args, **kwargs)
```

```
    add_directory_parser(inst)
```

Adds a parser instance to match on directory names

Parameters **inst** (*instance*) – parser instance

```
    add_file_parser(inst)
```

Adds a parser instance to match on filenames

Parameters **inst** (*instance*) – parser instance

```
    parse()
```

Parse the full directory tree in self._buildroot

```
    parse_file(d, wd='')
```

Parse a dictionary leaf

Parameters

- **d** (*dict*) – the dictionary to parse
- **wd** (*string*) – string containing the target directory

```
    save_dict(d, filepath)
```

Save dictionary to json file

Parameters

- **d** (*dict*) – the dictionary to save
- **filepath** (*string*) – string containing the full target filepath

1.1.2 Appie Base Parsers

class appie.**AppieBaseParser**

The default parser, does nothing but load files prepended with ‘_’ (underscore)

parse (*match_key*, *d*, *wd*, **args*, ***kwargs*)

Parses the dictionary *d* if *match_key* matches something it wants. Raises `StopParsing` if it doesn’t allow for further parsing of its tree.

Parameters

- **match_key** (*str*) – filename to match on
- **d** (*dict*) – dictionary belonging to *match_key*
- **wd** (*list*) – current working directory as a list

class appie.**AppieTextileParser**

Simple textile file to html parser

parse (*match_key*, *d*, *wd*, **args*, ***kwargs*)

Parses textile files (*match_key*) with `.textile` extension to html. Raises `AppieExceptStopParsing` when a file is matched and parsed.

Parameters

- **match_key** (*str*) – filename to match on
- **d** (*dict*) – dictionary belonging to *match_key*
- **wd** (*list*) – current working directory as a list

1.1.3 Appie Extensions

class appie.extensions.**AppieJPGParser** (**args*, ***kwargs*)

PNG parser converting JPGs to progressive JPG and a JPG thumb if they are larger than the ‘`jpg_size`’ setting.

Note to not parse JPG images and just copy them to the build root use a capital extension (`.JPG`).
The parsers are case sensitive!

class appie.extensions.**AppieMarkdownParser**

Simple markdown file to html parser

class appie.extensions.**AppiePNGParser** (**args*, ***kwargs*)

PNG parser converting PNGs to JPG and a JPG thumb

Note to not parse PNG images and just copy them to the build root use a capital extension (`.PNG`).
The parsers are case sensitive!

Indices and tables

- `genindex`
- `modindex`
- `search`

a

`appie.extensions`, 4

A

`add_directory_parser()` (`appie.Appie` method), 3
`add_file_parser()` (`appie.Appie` method), 3
`Appie` (class in `appie`), 3
`appie.extensions` (module), 4
`AppieBaseParser` (class in `appie`), 4
`AppieJPGParser` (class in `appie.extensions`), 4
`AppieMarkdownParser` (class in `appie.extensions`), 4
`AppiePNGParser` (class in `appie.extensions`), 4
`AppieTextileParser` (class in `appie`), 4

P

`parse()` (`appie.Appie` method), 3
`parse()` (`appie.AppieBaseParser` method), 4
`parse()` (`appie.AppieTextileParser` method), 4
`parse_file()` (`appie.Appie` method), 3

S

`save_dict()` (`appie.Appie` method), 3