
jigsaw Documentation

Release 2.1.3

nint8835

Nov 10, 2019

Contents:

1	jigsaw package	1
1.1	Submodules	1
1.2	jigsaw.Plugin module	1
1.3	jigsaw.PluginLoader module	1
2	plugin.json (Plugin Manifest)	5
3	Indices and tables	7
	Python Module Index	9
	Index	11

1.1 Submodules

1.2 jigsaw.Plugin module

class `jigsaw.Plugin.JigsawPlugin` (*manifest*, **args*)

Bases: `object`

Base class for all jigsaw plugins

__init__ (*manifest*, **args*)

Initializes the plugin

Parameters

- **manifest** – The plugin manifest
- **args** – Any other arguments passed to the plugin

disable ()

Handles cleaning up before disabling/unloading a plugin

enable ()

Handles the setup of a plugin on enable

1.3 jigsaw.PluginLoader module

class `jigsaw.PluginLoader.PluginLoader` (*plugin_paths*=(), *log_level*=20, *plugin_class*=<class 'jigsaw.Plugin.JigsawPlugin'>)

Bases: `object`

The main plugin loader class

`__init__ (plugin_paths=(), log_level=20, plugin_class=<class 'jigsaw.Plugin.JigsawPlugin'>)`

Initializes the plugin loader

Parameters

- **plugin_paths** – Paths to load plugins from
- **log_level** – Log level
- **plugin_class** – Parent class of all plugins

`disable_all_plugins ()`

Calls the disable method on all initialized plugins

`enable_all_plugins ()`

Calls the enable method on all initialized plugins

`get_all_plugins ()`

Gets all loaded plugins

Returns List of all plugins

`get_manifest (plugin_name)`

Gets the manifest for a specified plugin

Parameters **plugin_name** – The name of the plugin

Returns The manifest for the specified plugin

`get_module (name)`

Gets the module for a plugin

Parameters **name** – Name of the plugin

Returns The module

`get_plugin (name)`

Gets a loaded plugin

Parameters **name** – Name of the plugin

Returns The plugin

`get_plugin_loaded (plugin_name)`

Returns if a given plugin is loaded

Parameters **plugin_name** – The plugin to check to loaded status for

Returns Whether the specified plugin is loaded

`load_manifest (path)`

Loads a plugin manifest from a given path

Parameters **path** – The folder to load the plugin manifest from

`load_manifests ()`

Loads all plugin manifests on the plugin path

`load_plugin (manifest, *args)`

Loads a plugin from the given manifest

Parameters

- **manifest** – The manifest to use to load the plugin
- **args** – Arguments to pass to the plugin

load_plugins (*args)

Loads all plugins

Parameters **args** – Arguments to pass to the plugins

quickload (*args)

Loads all manifests, loads all plugins, and then enables all plugins :param args: The args to pass to the plugin

reload_all_manifests ()

Reloads all loaded manifests, and loads any new manifests

reload_all_plugins (*args)

Reloads all initialized plugins

reload_manifest (manifest)

Reloads a manifest from the disk :param manifest: The manifest to reload

reload_plugin (name, *args)

Reloads a given plugin

Parameters

- **name** – The name of the plugin
- **args** – The args to pass to the plugin

unload_plugin (name)

Unloads a specified plugin :param name: The name of the plugin

plugin.json (Plugin Manifest)

The `plugin.json` file (also known as the plugin manifest) is the file that defines all information a plugin requires to be loaded. While jigsaw specifies its own options that go in the manifest, the software that implements jigsaw is welcome to implement their own.

The options specified by jigsaw are:

name Required. The name of the plugin. This value **MUST** be unique.

dependencies A list of strings containing names of plugins that will be loaded before this plugin is loaded. If any plugins listed cannot be found or loaded, this plugin will not be loaded.

module_name The name the plugin will be imported as. Used for internal import workings. Defaults to plugin name with spaces replaced with underscores.

path The name of the file that contains the main *plugin class*. Defaults to `__init__.py`

main_class The main plugin class. Defaults to `Plugin`

Example `plugin.json`:

```
{
  "name": "Example Plugin",
  "dependencies": ["Example Dependency 1", "Example Dependency 2"],
  "main_class": "ExamplePlugin"
}
```


CHAPTER 3

Indices and tables

- genindex
- modindex
- search

j

jigsaw.Plugin, 1

jigsaw.PluginLoader, 1

Symbols

`__init__()` (*jigsaw.Plugin.JigsawPlugin* method), 1

`__init__()` (*jigsaw.PluginLoader.PluginLoader* method), 1

D

`disable()` (*jigsaw.Plugin.JigsawPlugin* method), 1

`disable_all_plugins()` (*jigsaw.PluginLoader.PluginLoader* method), 2

E

`enable()` (*jigsaw.Plugin.JigsawPlugin* method), 1

`enable_all_plugins()` (*jigsaw.PluginLoader.PluginLoader* method), 2

G

`get_all_plugins()` (*jigsaw.PluginLoader.PluginLoader* method), 2

`get_manifest()` (*jigsaw.PluginLoader.PluginLoader* method), 2

`get_module()` (*jigsaw.PluginLoader.PluginLoader* method), 2

`get_plugin()` (*jigsaw.PluginLoader.PluginLoader* method), 2

`get_plugin_loaded()` (*jigsaw.PluginLoader.PluginLoader* method), 2

J

`jigsaw.Plugin` (module), 1

`jigsaw.PluginLoader` (module), 1

`JigsawPlugin` (class in *jigsaw.Plugin*), 1

L

`load_manifest()` (*jigsaw.PluginLoader.PluginLoader* method), 2

`load_manifests()` (*jigsaw.PluginLoader.PluginLoader* method), 2

`load_plugin()` (*jigsaw.PluginLoader.PluginLoader* method), 2

`load_plugins()` (*jigsaw.PluginLoader.PluginLoader* method), 2

P

`PluginLoader` (class in *jigsaw.PluginLoader*), 1

Q

`quickload()` (*jigsaw.PluginLoader.PluginLoader* method), 3

R

`reload_all_manifests()` (*jigsaw.PluginLoader.PluginLoader* method), 3

`reload_all_plugins()` (*jigsaw.PluginLoader.PluginLoader* method), 3

`reload_manifest()` (*jigsaw.PluginLoader.PluginLoader* method), 3

`reload_plugin()` (*jigsaw.PluginLoader.PluginLoader* method), 3

U

`unload_plugin()` (*jigsaw.PluginLoader.PluginLoader* method), 3