apyshell

Release 1.0

Mark Anacker

Nov 17, 2023

CONTENTS

CHAPTER

ONE

APYSHELL PACKAGE

1.1 Subpackages

1.2 Submodules

1.2.1 apyshell.apyshell module

apyshell - Python Embedded apy script runnner

This script creates a framework for running lightweight scripts under the apyengine interpreter. It demonstrates embedding and controlling the engine. It can be run either stand-alone, or itself embedded into an application.

Credits

- version: 1.0
- last update: 2023-Nov-17
- License: MIT
- Author: Mark Anacker <closecrowd@pm.me>
- Copyright (c) 2023 by Mark Anacker

apyshell(*script*, *basedir='/opt/apyshell/scripts'*, *extensiondir='/opt/apyshell/extensions'*, *extension_opts=None*, *args=None*, *initscript=None*, *globals=False*)
Execute a script file under anyengine.

This is the main entry point.

Args: script : The .apy script to execute.

Returns: The return value. True for success, False otherwise.

savepid(*pfile*)
Save our current PID if we can.

Note: this doesn't do any checking of the file path. Take care if running under root.

usage()
Help message.

1.2.2 apyshell.extensionapi module

extensionapi - The interface between extensions and the engine.

An instance of this class is passed to the extensions at load time to give them an API back into the engine via the ExtensionMgr

Credits

- version: 1.0
- last update: 2023-Nov-13
- License: MIT
- Author: Mark Anacker <closecrowd@pm.me>
- Copyright (c) 2023 by Mark Anacker

class ExtensionAPI(*eng*, *parent*, *internal=True*)

Bases: `object`

getSysvar_(*name*, *default=None*)
Return the value of a system variable or a default

getvar_(*vname*, *default=None*)
Return the value of a script variable or a default

handleEvents(*name*, *data*)

isDef_(*name*)
Return True is the name is defined

listDefs_(*exception_list=None*)
Return a list of script def procs

list_Modules_()
Return a list of install_()-ed modules

loadScript_(*filename*, *persist=False*)
Load and execute a script file

logError(*modname=''*, **args*)

regcmd(*name*, *func=None*)
Add a new command callable by scripts

registerCmds(*mdict*)

setSysvar_(*name*, *val*)
Set a system variable

setvar_(*vname*, *val*)
Set a script variable

unregisterCmds(*mdict*)

debug(**args*)

1.2.3 apyshell.extensionmgr module

extensionmgr - Handles extension load/unload for apyshell.

This script supports the extension handling commands for apyshell.

Credits

- version: 1.0
- last update: 2023-Nov-13
- License: MIT
- Author: Mark Anacker <closecrowd@pm.me>
- Copyright (c) 2023 by Mark Anacker

class ExtensionMgr(*eng*, *epath*, *options=None*)
Bases: `object`

handleEvents(*name*, *data*)

isExtLoaded_(*ename*)
Return True if an extension is loaded

listExtensions_()
Return a list[] of currently-loaded extensions

loadExtension_(*ename*)
Load an extension by name (no paths allowed), and it has to be available in the extensions directory.

register()

scanExtensions_()
Return a list[] of available extensions

scanForExtensions(*dir*, *iflag*)

shutdown()

unloadExtension_(*ename*)
Remove a currently-loaded extension

quoteSpecial(*orig*)

1.2.4 apyshell.support module

support - Various stand-alone support functions for apyshell.

This file contains various utility functions used by apyshell.

Credits

- version: 1.0
- last update: 2023-Nov-13
- License: MIT
- Author: Mark Anacker <closecrowd@pm.me>
- Copyright (c) 2023 by Mark Anacker

checkFileName (*fname*)
Check a file name

Checks the given filename for characters in the set of a-z, A-Z, _ or -

Args: fname : The name to check

Returns: True if the name is entirely in that set False if there were invalid char(s)

debugMsg (*source=''*, **args*)

enableDebug (*arg*)

errorMsg (*source=''*, **args*)

get_queue (*q*, *defvalue=None*, *timeout=0*)

getparam (*table*, *key*, *default*, *remove=False*)

retError (*api*, *module*, *msg*, *ret=False*)

sanitizePath (*path*)
Clean a path.

Remove dangerous characters from a path string.

Args: path : The strin with the path to clean

Returns: The cleaned path or None if there was a problem

setparam (*table*, *key*, *value*, *replace=False*)

swapparam (*table*, *oldkey*, *newkey*, *value=None*)

unlock__ (*lock*)

1.3 Module contents

ApyShell - Python Embedded apy script runnner

This package creates a framework for running lightweight scripts under the apyengine interpreter. It demonstrates embedding and controlling the engine. It can be run either stand-alone, or itself embedded into an application.

The companion project "apyengine" has documentation and examples of how to use the apyengine outside of this package. The entire apyengine package is included in this package for convenience. <https://github.com/closecrowd/apyengine>

Credits

- version: 1.0
- last update: 2023-Nov-17
- License: MIT
- Author: Mark Anacker <closecrowd@pm.me>
- Copyright (c) 2023 by Mark Anacker

PYTHON MODULE INDEX

a

INDEX