gs.recipe.base Documentation

Release 1.0.0

GroupServer.org

Sep 27, 2017

Contents

This product supplies the *gs.recipe.base.Recipe* **abstract** base class. It provides some useful methods that make writing `zc.buildout` recipes easier.

Contents:

CHAPTER 1

gs.recipe.base API Reference

Currently this module only supplies the *gs.recipe.base.Recipe* **abstract** base class.

class gs.recipe.base.**Recipe**(*buildout*, *name*, *options*)

Recipe **abstract** base class.

Parameters

- **buildout** (*dict*) – The buildout options.
- **name** (*str*) – The name of the recipe.
- **options** (*dict*) – The recipe options.

Raises

- **ValueError** – If buildout['buildout'] is missing
- **ValueError** – If buildout['buildout']['directory'] is missing
- **ValueError** – If buildout['buildout']['bin-directory'] is missing

Normally zc.buildout handles passing the correct values to buildout, name, and options. As a result conformance is not as hard as it looks from the signature of the __init__.

Concrete implementations of this base class must implement the *install()* method, and the *update()* method.

install()

A **concrete** implementation of this method should call the following.

- *should_run()* to determine if the recipe should run at all.
- *mark_locked()* to lock the recipe after it has run.

mark_locked()

Create a lock file for the recipe.

Returns None

A lock-file is used to record that a recipe has already been run, and it should be skipped. The presence or absence of the fille is important, rather than the contents of the file.

should_run()
Determine if the recipe should be run.

Returns `True` if the recipe should be run, `False` otherwise.

Return type `bool`

A recipe should be run in two possible scenarios.

1. The `run-once` buildout option is set to false, off, or no.
2. The `run-once` is absent — or set to any value other than false, off or no — and the lock-file that is created by the *`mark_locked()`* method is absent.

As a **side effect,** a message is displayed to `sys.stdout` if `should_run()` returns `False`. This message tells the administrator how to force the recipe to be run.

update()
Update the component, to be filled out by **concrete** implementations.

Example

In the following example the **concrete** class `SetupGSRecipe` implements the `install` method. It calls `should_run` to determine if the recipe should be run at all, and calls `mark_locked` once it is done.

```
from gs.recipe.base import Recipe


class SetupGSRecipe(Recipe):

   def get_script_command(self):
       'Get the command to do stuff'

    def install(self):
        if self.should_run():
            command = self.get_script_command()
            try:
                retcode = subprocess.call(command, shell=True)
                if retcode == 0:
                    self.mark_locked()
                    sys.stdout.write('GroupServer site created\n\n')
                else:
                    m = '{0}: Issue running\n\t{1}\nReturned {2}\n'
                    msg = m.format(self.name, command, retcode)
                    raise UserError(msg)
            except OSError as e:
                m = '{0}: Failed to run\n\t{1}\n{2}\n'
                msg = m.format(self.name, command, e)
                raise UserError(msg)

    return tuple()

def update(self):
    self.install()
```

CHAPTER 2

Changelog

1.0 (2014-05-15)

- Initial version.
- Forked off `gs.recipe.setupgs`

CHAPTER 3

Resources

- Code repository: https://source.iopen.net/groupserver/gs.group.recipe.base
- Questions and comments to http://groupserver.org/groups/development
- Report bugs at https://redmine.iopen.net/projects/groupserver

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Index