[**stevedore动态加载模块**](https://www.cnblogs.com/CaesarLinsa/p/8571297.html)

stevedore动态加载模块，stevedore使用setuptools的entry points来定义并加载插件。entry point引用的是定义在模块中的对象，比如类、函数、实例等，只要在import模块时能够被创建的对象都可以，但stevedore的优良之处在于，其可以动态调用所有插件的某一相同方法，将返回结果存入list中。这对于处理具有相同行为的类，起到极大的便利。

**如何构建插件？**

先定义一个基类，所有继承该基类的子类既为各个独立的插件。

import abc

import six

# 定义基类，只包含format方法，文件路径 plugins/

@six.add\_metaclass(abc.ABCMeta)

class FormatterBase(object):

@abc.abstractmethod

def format(self, data):

pass

 定义一个插件Simple，实现format方法

1 from plugins import plugin\_base

2 # 文件路径 plugins/simple.py

3 class Simple(plugin\_base.FormatterBase):

4

5 data = {'ca':'cangyue', "name":"caesar"}

6

7 def format(self, data):

8 lines = []

9 for name, value in sorted(data.items()):

10 line = '{name}={value}'.format(

11 name = name,

12 value = value

13 )

14 lines.append(line)

15 return ''.join(lines)

 编写setup.py文件，用于发布包。 包名demo，版本0.1， 组名stevedore.formatter  两个插件simple和plain都对应Plugins.simple路径中的Simple类。文件路径与plugins同路径，注意plugins中有空\_\_init\_\_.py文件，

表示是一个python包。

1 from setuptools import setup, find\_packages

2 setup(

3 name = "demo",

4 version = "0.1",

5 packages = find\_packages(),

6 entry\_points={

7 'stevedore.formatter': [

8 'simple = plugins.simple:Simple',

9 'plain = plugins.simple:Simple',

10 ],

11 },

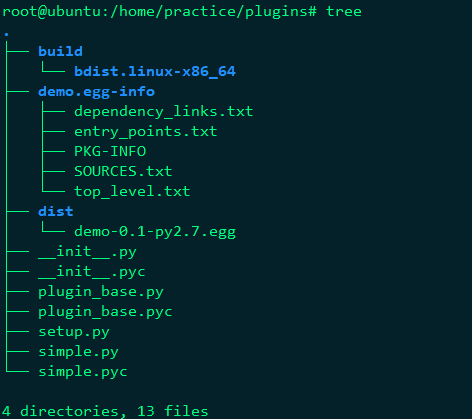
)

 使用命令发布包 python setup.py install 安装，安装后会在egg文件夹下生成entry\_points.txt文件，描述组名和插件

[stevedore.formatter]

simple = plugins.simple:Simple

plain = plugins.simple:Simple



**如何调用插件中方法？**

1.加载某一插件，并调用其中方法

1 from stevedore import driver

2 mgr = driver.DriverManager(

3 namespace='stevedore.formatter',

4 name='simple',

5 invoke\_on\_load=True

6 ) # 加载组名或者说命名空间为stevedore.formetter下name为simple的插件

7 >>> st = mgr.driver.format({'caesa':'sss'})

8 >>> data = mgr.driver.data # mgr.driver实例化simple对象并调用其data属性

9 >>> print data

10 {'ca': 'cangyue', 'name': 'caesar'}

 2.加载命名空间下所有插件，并调用所有插件的format方法

1 from stevedore import extension

2 mgr = extension.ExtensionManager(

3 namespace='stevedore.formatter',

4 invoke\_on\_load=True) # 加载stevedore.formatter命名空间下所有插件

5

6 def d(ext):

7 return ext.obj.data

8

9 >>> dat = mgr.map(d) #对所有插件调用d方法，d方法中ext为插件的extension对象，调用其obj即可实例化该对象

10 >>> print dat

11 [{'ca': 'cangyue', 'name': 'caesar'}, {'ca': 'cangyue', 'name': 'caesar'}]

12

13

14 def m(ext):

15 data = ext.obj.data #获取插件对象的data属性值

16 return ext.obj.format(data) 调用对象的format方法，将属性值传入

17

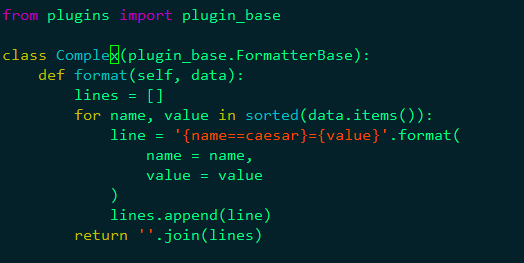
18 >>> ret = mgr.map(m)

19 >>> print ret

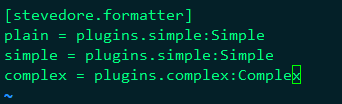
20 ['ca=cangyuename=caesar', 'ca=cangyuename=caesar']

**特记：**

当想重新开发一个插件时，需要在编写完成后，重新安装，否则不生效。比如鄙人想重新开发一个complex，在相应路径下添加complex.py



在entry\_point中添加插件



但重新执行时，加载的仍然是两个插件

