

Python之包管理工具快速入门

每日一个Linux、Python干货



关注的人都加薪了

在Python环境中已经有很多成熟的包，可以通过安装这些包来扩展我们的程序。

例如，很多时候Python开发人员都会去PyPI网站去查找自己想要使用的包，然后进行安装。PyPI（Python Package Index）是获得第三方Python软件包以补充标准库的一个站点。

在安装Python包的过程中，经常涉及到distutils、setuptools、distribute、setup.py、easy_install、easy_install和pip等等。

突然对Python的包管理凌乱了，这些工具关系是什么，应该怎么去选择使用？

下面就简单介绍下Python包管理相关的一些内容。

Python包管理工具

这么多的Python包管理工具，刚开始接触Python时会有点不知如何选择。其实，当查过Python的官方文档之后，慢慢就理清了这些工具之间的关系了，也就没有这么混乱了。

下面就分别对Python中的包管理工具进行简单的介绍。

distutils

distutils 是 python 标准库的一部分，这个库的目的是为开发者提供一种方便的打包方式，同时为用户提供方便的安装方式。

我们经常使用的setup.py就是基于distutils实现的，然后通过setup.py就可以进行打包或者安装了。

看一个简单的例子，找一个目录创建三个文件foo.py、bar.py和setup.py，其中setup.py的内容如下：

```
from distutils.core import setup

setup(
    name='fooBar',
    version='1.0',
    author='Will',
    author_email='wilber@sh.com',
    url='http://www.cnblogs.com/wilber2013/',
    py_modules=['foo', 'bar'],
)
```

然后，在该目录中运行 `python setup.py sdist`，会得到以下输出，同时生成了一个“fooBar-1.0.zip”包。

```
C:\packageManagement>python setup.py sdist
running sdist
running check
warning: sdist: manifest template 'MANIFEST.in' does not exist (using default file list)
warning: sdist: standard file not found: should have one of README, README.txt

writing manifest file 'MANIFEST'
creating fooBar-1.0
copying files to fooBar-1.0...
copying bar.py -> fooBar-1.0
copying foo.py -> fooBar-1.0
copying setup.py -> fooBar-1.0
creating dist
creating 'dist\fooBar-1.0.zip' and adding 'fooBar-1.0' to it
adding 'fooBar-1.0\bar.py'
adding 'fooBar-1.0\foo.py'
adding 'fooBar-1.0\PKG-INFO'
adding 'fooBar-1.0\setup.py'
removing 'fooBar-1.0' (and everything under it)
C:\packageManagement>
```

使用者就可以解压缩这个包然后执行 `python setup.py install` 进行安装，然后就可以使用foo、bar这两个模块了：

```
C:\packageManagement\dist\fooBar-1.0>ls
PKG-INFO bar.py foo.py setup.py

C:\packageManagement\dist\fooBar-1.0>python setup.py install
running install
running build
running build_py
creating build
creating build\lib
copying foo.py -> build\lib
copying bar.py -> build\lib
running install_lib
running install_egg_info
Writing C:\Python27\Lib\site-packages\fooBar-1.0-py2.7.egg-info
C:\packageManagement\dist\fooBar-1.0>
```

关于更过如何编写setup.py的内容，请自行参阅Python官方文档中的setuptools部分。

setuptools 和 distribute

setuptools 是对 distutils 的增强，尤其是引入了包依赖管理。我们可以通过 `ez_setup.py` 来安装 setuptools。

至于 distribute，它是 setuptools 的一个分支版本。分支的原因是有一部分开发者认为 setuptools 开发太慢。但现在，distribute 又合并回了 setuptools 中，所以可以认为它们是同一个东西。

前面看到 `setup.py` 可以创建一个压缩包，而 setuptools 使用了一种新的文件格式（.egg），可以为 Python 包创建 egg 文件。setuptools 可以识别 .egg 文件，并解析、安装它

easy_install

当安装好 setuptools/distribute 之后，我们就可以直接使用 easy_install 这个工具了：

1. 从 PyPI 上安装一个包：当使用 `easy_install package` 命令后，easy_install 可以自动从 PyPI 上下载相关的包，并完成安装，升级
2. 下载一个包安装：通过 `easy_install package.tgz` 命令可以安装一个已经下载的包
3. 安装 egg 文件：通过 `easy_install package.egg` 可以安装一个 egg 格式的文件

通过 `easy_install --help` 命令可以获取该命令相关的帮助提示：

```

C:\Users\tianwl>easy_install --help
Global options:
--verbose (-v)    run verbosely (default)
--quiet (-q)      run quietly (turns verbosity off)
--dry-run (-n)    don't actually do anything
--help (-h)       show detailed help message
--no-user-cfg     ignore pydistutils.cfg in your home directory

Options for 'easy_install' command:
--prefix          installation prefix
--zip-ok (-z)     install package as a zipfile
--multi-version (-m) make apps have to require() a version
--upgrade (-U)    force upgrade (searches PyPI for latest versions)
--install-dir (-d) install package to DIR
--script-dir (-s) install scripts to DIR
--exclude-scripts (-x) Don't install scripts
--always-copy (-a) Copy all needed packages to install dir
--index-url (-i)   base URL of Python Package Index
--find-links (-f)  additional URL(s) to search for packages
--build-directory (-b) download/extract/build in DIR; keep the results
--optimize (-O)   also compile with optimization: -O1 for "python -O",
                  -O2 for "python -OO", and -OO to disable
                  [default: -O0]
--record          filename in which to record list of installed
                  files
--always-unzip (-Z) don't install as a zipfile, no matter what
--site-dirs (-S)  list of directories where .pth files work
--editable (-e)   Install specified packages in editable form
--no-deps (-N)    don't install dependencies
--allow-hosts (-H) pattern(s) that hostnames must match
--local-snapshots-ok (-l) allow building eggs from local checkouts
--version         print version information and exit
--no-find-links   Don't load find-links defined in packages being
                  installed
--user            install in user site-package
                  'C:\Users\tianwl\AppData\Roaming\Python\Python27\
                  site-packages'

usage: easy_install-script.py [options] requirement_or_url ...
or: easy_install-script.py --help

```

根据上面的分析，可以看到setuptools/distribute和easy_install之间的关系：

*setuptools/distribute 都扩展了 distutils，提供了更多的功能

*easy_install是基于setuptools/distribute的一个工具，方便了包的安装和省级

pip

pip是目前最流行的Python包管理工具，它被当作easy_install的替代品，但是仍有大量的功能建立在setuptools之上。

easy_install 有很多不足：安装事务是非原子操作，只支持 svn，没有提供卸载命令，安装一系列包时需要写脚本。pip 解决了以上问题，已经成为新的事实标准。

pip的使用非常简单，并支持从任意能够通过 VCS 或浏览器访问到的地址安装 Python 包：

*安装： `pip install SomePackage`

***卸载：** `pip uninstall SomePackage`

文章的下面部分就重点介绍一下pip相关的内容。

使用pip

在大家使用Python中，推荐使用pip进行Python包管理，pip的安装和使用都比较方便。

pip安装

pip的安装有两种常用的方式：

1. 下载get-pip.py文件，然后执行 `python get-pip.py` 进行安装（如果没有安装setuptools，那么get-pip.py会帮忙安装）
2. 现在pip源码包，然后通过setup.py进行安装

pip常用命令

对于pip，最常用的肯定还是 `pip --help` ，通过帮助文档，就可以大概知道如何使用命令和参数。

```
C:\Users\tianui>pip --help
Usage:
  pip <command> [options]

Commands:
  install          Install packages.
  uninstall        Uninstall packages.
  freeze           Output installed packages in requirements format.
  list             List installed packages.
  show             Show information about installed packages.
  search           Search PyPI for packages.
  wheel            Build wheels from your requirements.
  help            Show help for commands.

General Options:
  -h, --help          Show help.
  --isolated          Run pip in an isolated mode, ignoring
                     environment variables and user configuration.
  -v, --verbose       Give more output. Option is additive, and can be
                     used up to 3 times.
  -U, --version       Show version and exit.
  -q, --quiet         Give less output.
  --log <path>       Path to a verbose appending log.
  --proxy <proxy>     Specify a proxy in the form
                     user:passwd@proxy.server:port.
  --retries <retries> Maximum number of retries each connection should
                     attempt (default 5 times).
  --timeout <sec>     Set the socket timeout (default 15 seconds).
  --exists-action <action> Default action when a path already exists:
                     <s>witch, <i>gnore, <u>ipe, <b>ackup.
  --trusted-host <hostname> Mark this host as trusted, even though it does
                     not have valid or any HTTPS.
  --cert <path>       Path to alternate CA bundle.
  --client-cert <path> Path to SSL client certificate, a single file
                     containing the private key and the certificate
                     in PEM format.
  --cache-dir <dir>   Store the cache data in <dir>.
  --no-cache-dir      Disable the cache.
  --disable-pip-version-check Don't periodically check PyPI to determine
                     whether a new version of pip is available for
                     download. Implied with --no-index.
```

pip常用命令集合：

使用	命令
从 PyPI 安装软件包	pip install SomePackage
卸载软件包	pip uninstall SomePackage
查看以安装软件包	pip list
查看可升级软件包	pip list --outdated
升级软件包	pip install --upgrade

	Som ePac kage
查看 软件 包安 装了 哪些 文件 及路 径等 信息	pip sho w -- files Som ePac kage
安装 软件 包的 指定 版本 号	pip insta ll Som ePac kage # lates t versi on pip insta ll Som ePac kage ==1. 0.4 # spec ific versi on pip insta ll 'So meP acka ge>

	<code>=1.0 .4' # mini mu m versi on</code>
根据 依赖 文件 安装 软件 包	<code>pip free ze > requ irem ents. txt # 使 用 pip 导出 依赖 文件 列表 pip insta ll -r requ irem ents. txt # 根 据依 赖文 件列 表, 自动 安装 对应 的软 件包</code>

总结

本文介绍了Python的各个包管理工具之间的关系，相信通过这篇文章的介绍，就不会再对distutils、setuptools、distribute、setup.py、easy_install、

easy_install和pip等等名词感觉混乱了。

大致了解这些Python包管理工具后，大家应该也清楚了怎么选择和使用。

本文没有涉及如何制作并发布一个Python包，感兴趣的同学可以去Python官网上查查。

作者：包子

原文链接：<https://www.cnblogs.com/liangqihui/p/6895108.html>

[《Linux云计算及运维架构师高薪实战班》2018年09月17日即将开课中，120天冲击Linux运维年薪30万，改变速约~~~~](#)

*声明：推送内容及图片来源于网络，部分内容会有所改动，版权归原作者所有，如来源信息有误或侵犯权益，请联系我们删除或授权事宜。

- END -



Python福利包

主讲人：上市公司十年开发经理

福利1：15册Python入门书籍

福利2：30集Python入门视频

福利3：50个Python商业项目源代码



长按识别二维码，即刻获取



每天精选技术干货，十万Linux人订阅

◀ **Linux人充电第一站**

长按识别二维码 关注马哥Linux运维

更多Linux好文请点击 [【阅读原文】](#) 哦



[阅读原文](#)