

Buildout Rocks

By: 逝雾x

Buildout是什么？

- Java中的Maven
- 基于配置文件
- 优美地进行依赖管理
- develop egg
- repeatable
- buildout can do everything!

为什么要使用Buildout

- 场景一
- `install_requires=['cornice', 'gunicorn', 'CouchDB', 'cryptacular', 'pyelasticsearch', 'PasteScript', 'tokenlib', 'requests<2.0', 'upyun']`
- 看起来很乱
- `.egg` format?
- `pip install --egg package_name?`
- `pip freeze` (过于精确, 而不是需求)

为什么要使用Buildout

- 场景二
- `python setup.py develop`
- may get non-final version
- Starting with v1.4, pip will only install stable versions as specified by PEP426 by default. If a version cannot be parsed as a compliant PEP426 version then it is assumed to be a pre-release.
- `pip install -r requirement.txt`
- `pip install --egg -r requirement.txt` (python setup install again)

为什么要使用Buildout

- 场景三
- 升级依赖
- `pip install -U package_name ?`
- `pip freeze --local | grep -v '^-e' | cut -d = -f 1 | xargs pip install -U ?`
- `pip install -U -r requirement.txt ?`

为什么要使用Buildout

- 场景四
- 移除依赖
- `python setup.py install --record files.txt`
- `cat files.txt | xargs rm -rf`
- `pip uninstall package_name` (不会递归删除)
- `virtualenv --clear $VIRTUAL_ENV`

为什么要使用Buildout

- 场景五
- replicate
- 完全一样

Buildout

- `pip install zc.buildout`
- `buildout init`
- 放弃
- `mkvirtualenv --no-setuptools env`
- `python bootstrap.py`

buildout.cfg

最小配置

```
[buildout]
```

```
parts =
```

依赖管理

```
[buildout]
```

```
parts = sample
```

```
[sample]
```

```
recipe = zc.recipe.egg
```

```
eggs = pyramid
```

```
interpreter = py
```

```
dependent-scripts = true
```

Buildout

- buildout = parts + recipe
- create a recipe
- buildout.cfg VS .installed.cfg
- os.options.created()
- bin/buildout annotate

buildout.cfg

- `${part_name:option_key}`
- `<= 父part`
- `[part_name:conditions]`
- `option_key += option_value`
- `extends`
- `$HOME/.buildout/default.cfg`

uninstall

```
def uninstall_method(name, options):  
    pass
```

[zc.buildout.uninstall]

make sure use same name

不需要关注创建的文件和文件夹

删除之前仍然可访问(做备份)

develop package

- [buildout]
- develop = package_dir
- parts = sample
- [sample]
- eggs = package_name

Extensions

- create a extension
- `def ext(buildout)`
- `zc.buildout.extension`

Buildout

- Download utility
- download cache download-cache
- configuration cache
- bin/buildout setup setup.py command
- bin/buildout -D
- meta recipe(buildout.parse)

API

- `zc.buildout.easy_install.install`
- `zc.buildout.easy_install.scripts`

直击GAE

- 利用buildout创建pyramid应用

Q&A

- 如果能帮到各位, 不胜荣幸

感谢

- 排名不分先后
- PyCon
- Beijing GDG
- Mark Huang