



链滴

Deploy Springboot With Python

作者: [liumeijian](#)

原文链接: <https://ld246.com/article/1557498450002>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

python 自动全量，增量部署脚本

运行在python3+下 命令 python3 deploy.py app springboot

-----> app (应用部署目录) springboot (应用zip包名称)

必须在用户目录下新建Releases目录，把此脚本放到下面命名为deploy.py

项目目录 ~/app/application/ 其中app要先自己手动创建

自用 仅做参考 可以结合<http://www.liumj.cn/articles/2019/04/20/1555736649588.html>

```
# Program: springboot项目部署
```

```
import datetime
```

```
import os
```

```
import re
```

```
import shutil
```

```
import stat
```

```
import sys
```

```
import tarfile
```

```
import zipfile
```

```
# Home目录
```

```
HomePath = os.path.expanduser('~')
```

```
#config 可配置项
```

```
# 我自己的环境 其他环境去掉下面这句
```

```
# HomePath = HomePath + "/PycharmProjects/insteadShell"
```

```
# 是否备份
```

```
isbackUp = False
```

```
# 增量部署默认只会覆盖jar/war/lib下的包，需要强制覆盖的配置需要额外添加
```

```
# 强制需要覆盖的文件 例子:force_instead_file= ["data"]
```

```
force_instead_file= []
```

```
backUpDirName = '/backup'
```

```
allFileToDeploy = []
```

```
dirName = ""
```

```
dirPath = ""
```

```
HomePath = HomePath + "/"
```

```
ReleasePath = os.path.expanduser(HomePath + 'Releases/')
```

```
# 移动文件
```

```
def movefile(srcfile, dstfile):
```

```
    fpath, fname = os.path.split(dstfile) # 分离文件名和路径
```

```
    if not os.path.exists(fpath):
```

```
        os.makedirs(fpath) # 创建路径
```

```
    shutil.move(srcfile, dstfile) # 移动文件
```

```
# 备份
```

```
def makeBakDir(application_path):
```

```
    if not os.path.exists(application_path + backUpDirName):
```

```

    os.makedirs(application_path + backUpDirName)
return application_path + backUpDirName

# tar 打包
def tar(fname, fpath):
    t = tarfile.open(fname + ".tar", "w")
    for root, dir, files in os.walk(fpath):
        for file in files:
            fullpath = os.path.join(root, file)
            if backUpDirName not in fullpath and "/log/" not in fullpath and "/logs/" not in fullpat
and "/nohup.out" not in fullpath:
                t.add(fullpath)
    t.close()

# unzip 解压zip包
def un_zip(zipFileName, target):
    f = zipfile.ZipFile(ReleasePath + zipFileName + ".zip", 'r')
    for file in f.namelist():
        f.extract(file, target)
    f.close()

def exit(errorMsg):
    print(errorMsg)
    sys.exit(1)

##### 获取所有需要部署的项目添加到变量 allFileToDeploy
def allFileToDeployM():
    global allFileToDeploy
    global dirName
    global dirPath
    # 所有在Release目录下的zip文件
    allfiles = os.listdir(ReleasePath)
    allZipFiles = [re.sub(r".zip", "", f) for f in allfiles if re.search('.zip$', f)]
    if len(sys.argv) <= 1:
        exit("参数错误 无法部署")

# 全部署 python3 deploy.py
if len(sys.argv) == 2:
    if not os.path.isdir(HomePath + sys.argv[1]):
        exit("请输入正确的应用部署目录")
    dirName = sys.argv[1]
    dirPath = HomePath + sys.argv[1]
    allFileToDeploy = allZipFiles
    # asw = input('是否在目录:%s 部署所有应用(y:n)\n'%(dirName))
    # if (asw != 'y'):
    #     exit("停止部署")

# 部署目录和项目 python3 deploy.py app id-center
if len(sys.argv) >= 3:
    if not os.path.isdir(HomePath + sys.argv[1]):
        exit("请输入正确的应用部署目录")
    else:
        dirName = sys.argv[1]

```

```

dirPath = HomePath + sys.argv[1]
for i in range(len(sys.argv)):
    if (i >= 2):
        if sys.argv[i] in allZipFiles:
            allFileToDeploy.append(sys.argv[i])
        else:
            print("不存在应用:" + sys.argv[i] + ".zip")

print("所有即将部署的应用:", allFileToDeploy)
if len(allFileToDeploy) == 0:
    exit("没有需要部署的项目")
# asw = input("是否开始部署(y:n)\n")
# if asw == 'y':
#     print("开始部署")
# else:
#     exit("停止部署")

# 开始部署 备份
def deployApplication():
    for application in allFileToDeploy:
        print("开始部署目录:", dirName, "下的:", application, "项目")
        # 全量部署
        dir_path = HomePath + dirName + '/'
        application_path = dir_path + application
        if not os.path.isdir(application_path):
            un_zip(application, dir_path)
            os.chdir(application_path)
            os.chmod(application_path + "/start.sh", stat.S_IRWXU)
            os.system(application_path + "/start.sh > nohup.out 2>&1 &")
        else:
            # 开始备份
            os.chdir(application_path)
            os.chmod(application_path + "/stop.sh", stat.S_IRWXU)
            os.system(application_path + "/stop.sh")
            os.chdir(makeBakDir(application_path))
            if isbackUp:
                otherStyleTime = datetime.datetime.now().strftime("%Y%m%d%H%M%S")
                tar(application + otherStyleTime, application_path)
            # 增量部署
            un_zip(application, HomePath + 'Releases/')
            application_release_path = HomePath + 'Releases/' + application + '/'
            if os.path.isfile(application_release_path + application + '.war'):
                if os.path.isfile(application_path + application + '.war'):
                    os.remove(application_path + "/" + application + ".war")
                    shutil.copy(application_release_path + application + '.war',
                                application_path + "/" + application + ".war")
            else:
                if os.path.isfile(application_path + application + '.jar'):
                    os.remove(application_path + "/" + application + ".jar")
                    shutil.copy(application_release_path + application + '.jar',
                                application_path + "/" + application + ".jar")
                if os.path.isdir(application_path + "/lib"):
                    shutil.rmtree(application_path + "/lib", True)

```

```
    shutil.copytree(application_release_path + '/lib', application_path + "/lib")
    # 补充剩余空缺文件
listdir = os.listdir(application_path)
re_listdir = os.listdir(application_release_path)
for x in re_listdir:
    if x not in listdir or x in force_instead_file:
        if os.path.isdir(application_release_path + x):
            if os.path.isdir(application_path + "/" + x):
                shutil.rmtree(application_path + "/" + x)
            shutil.copytree(application_release_path + '/' + x, application_path + "/" + x)
        else:
            shutil.copy(application_release_path + '/' + x, application_path + "/" + x)
if os.path.isfile(application_path + "/start.sh"):
    os.chdir(application_path)
    os.chmod(application_path + "/start.sh", stat.S_IRWXU)
    os.system(application_path + "/start.sh > nohup.out 2>&1 &")
```

```
def main():
    allFileToDeployM()
    deployApplication()
    print("部署成功")
```

```
if __name__ == '__main__':
    main()
```