

管理第三方包

对第三方包的管理主要包含查找、安装和卸载三个部分的操作。

安装

使用 `pip install <包名>` 命令可以安装指定的第三方资源包。

```
pip install ipython # 安装ipython包
```

使用 `install` 命令下载第三方资源包时，默认是从 `pythonhosted` 下载，由于各种原因，在国内下载速度相对来说比较慢，在某些时候甚至会出现连接超时的情况，我们可以使用国内镜像来提高下载速度。

临时修改

如果只是想临时修改某个第三方资源包的下载地址，在第三方包名后面添加 `-i` 参数，再指定下载路径即可，格式为 `pip install <包名> -i <国内镜像路径>`

```
pip install ipython -i https://pypi.douban.com/simple
```

永久修改

除了临时修改 `pip` 的下载源以外，我们还能永久改变 `pip` 的默认下载路径。

在当前用户目录下创建一个 `pip` 的文件夹，然后再在文件夹里创建 `pip.ini` 文件并输入一下内容：

```
[global]
index-url=https://pypi.douban.com/simple
[install]
trusted-host=pypi.douban.com
```

常见国内镜像

- 阿里云 <https://mirrors.aliyun.com/pypi/simple/>
- 中国科技大学 <https://pypi.mirrors.ustc.edu.cn/simple/>
- 豆瓣(douban) <https://pypi.douban.com/simple/>
- 清华大学 <https://pypi.tuna.tsinghua.edu.cn/simple/>
- 中国科学技术大学 <https://pypi.mirrors.ustc.edu.cn/simple/>

卸载

使用 `pip install <包名>` 命令可以用来卸载指定的第三方资源包。

```
pip uninstall ipython # 卸载ipython包
```

查找

使用 `pip list` 或者 `pip freeze` 命令可以来管理第三方资源包。这两个命令的功能一致，都是用来显示当前环境里已经安装的包，区别在于 `pip list` 会列出所有的包，包括一些无法uninstall的包；而 `pip freeze` 只会列出我们安装的第三方包。

总结

开发中，我们通常会使用很多第三方的资源包，我们在将程序部署到服务器的时候，不仅要把代码上传到服务器，同时还需要把代码里用到的第三方资源包告诉服务器。那么这里就有两个问题：

1. 当我们电脑上运行很多个项目，每个项目使用的第三方资源包又不一致时，怎样将代码和它使用到的第三方资源包放在一起呢？答:虚拟环境
2. 怎样将自己代码使用到的第三方资源包告诉给服务器？

1. 使用 `pip freeze > requirements.txt` 命令，将代码里使用到的第三方资源包以及版本号写入到 `requirements.txt` 文件，在部署时，同时将 `requirements.txt` 文件上传到服务器。
2. 服务器在拿到代码以后，首先运行 `pip install -r requirements.txt` 命令，将文件里列出的所有第三方框架先安装到服务器，然后才能运行代码。

使用pycharm管理第三方包

除了使用pip 命令管理第三方资源包以外，我们还能使用pycharm来对第三方包进行管理。

🔍

- > Appearance & Behavior
 - Keymap
- > Editor
 - Plugins
- > Version Control
- ▼ Project: 01-代码
 - Project Interpreter
 - Project Structure
- > Build, Execution, Deployment
- > Languages & Frameworks
- > Tools

Project: 01-代码 > Project Interpreter

⚙ For current project

Project Interpreter: Python 3.7 C:\Users\chris\AppData\Local\Programs\Python\Python37\python.exe

Package	Version	Latest version
Pygments	2.5.2	2.5.2
astroid	2.2.5	▲ 2.3.3
autopep8	1.4.4	▲ 1.5
backcall	0.1.0	
colorama	0.4.3	
cycler	0.10.0	
decorator	4.4.1	
ipython	7.12.0	
ipython-genutils	0.2.0	
isort	4.3.21	
jedi	0.16.0	
kiwisolver	1.1.0	
lazy-object-proxy	1.4.2	
matplotlib	3.1.3	
mccabe	0.6.1	
numpy	1.18.1	
parso	0.6.1	
pickleshare	0.7.5	
pip	19.2.3	

+
-
▲
▼
🔍

千锋Python人工智能学院