**Flask 安装**

###### pip 安装flask

1、安装 Flask 最便捷的方式是使用虚拟环境。虚拟环境是 Python 解释器的一个私有副本,在这个环境中你可以安装私有包,而且不会影响系统中安装的全局 Python 解释器。虚拟环境非常有用,可以在系统的 Python 解释器中避免包的混乱和版本的冲突。为每个程序单独创建虚拟环境可以保证程序只能访问虚拟环境中的包,从而保持全局解释器的干净整洁,使其只作为创建(更多)虚拟环境的源。使用虚拟环境还有个好处,那就是不需要管理员权限。虚拟环境使用第三方实用工具 virtualenv 创建。输入以下命令可以检查系统是否安装了virtualenv :

$ virtualenv --version

如果结果显示错误,你就需要安装这个工具。Python 3.3 通过 venv 模块原生支持虚拟环境,命令为 pyvenv 。 pyvenv 可以替代 virtualenv 。不过要注意,在 Python 3.3 中使用 pyvenv 命令创建的虚拟环境不包含 pip ,你需要进行手动安装。Python 3.4 改进了这一缺陷, pyvenv 完全可以代替 virtualenv 。

大多数 Linux 发行版都提供了 virtualenv 包。例如,Ubuntu 用户可以使用下述命令安装它:

$ sudo apt-get install python-virtualenv

如果你的电脑是 Mac OS X 系统,就可以使用 easy\_install 安装 virtualenv :

$ sudo pip install virtualenv

2、现在你要新建一个文件夹,用来保存示例代码(示例代码可从 GitHub 库中获取)。我们在前言的“如何使用示例代码”一节中说过,获取示例代码最简便的方式是使用 Git 客户端直接从 GitHub 下载。下述命令从 GitHub 下载示例代码,并把程序文件夹切换到“1a”版本,即程序的初始版本:

$ git clone https://github.com/miguelgrinberg/flasky.git

$ cd flasky

$ git checkout 1a

3、下一步是使用 virtualenv 命令在 flasky 文件夹中创建 Python 虚拟环境。这个命令只有一个必需的参数,即虚拟环境的名字。创建虚拟环境后,当前文件夹中会出现一个子文件夹,名字就是上述命令中指定的参数,与虚拟环境相关的文件都保存在这个子文件夹中。按照惯例,一般虚拟环境会被命名为 venv :

$ virtualenv venv

New python executable in venv/bin/python2.7

Also creating executable in venv/bin/python

Installing setuptools............done.

Installing pip...............done.

现在,flasky 文件夹中就有了一个名为 venv 的子文件夹,它保存一个全新的虚拟环境,其

中有一个私有的 Python 解释器。在使用这个虚拟环境之前,你需要先将其“激活”。如果

你使用 bash 命令行(Linux 和 Mac OS X 用户),可以通过下面的命令激活这个虚拟环境:

$ source venv/bin/activate

4、虚拟环境被激活后,其中 Python 解释器的路径就被添加进 PATH 中,但这种改变不是永久性的,它只会影响当前的命令行会话。为了提醒你已经激活了虚拟环境,激活虚拟环境的命令会修改命令行提示符,加入环境名:

(venv) $

当虚拟环境中的工作完成后,如果你想回到全局 Python 解释器中,可以在命令行提示符下

输入 deactivate

5、大多数 Python 包都使用 pip 实用工具安装,使用 virtualenv 创建虚拟环境时会自动安装

pip 。激活虚拟环境后, pip 所在的路径会被添加进 PATH 。

执行下述命令可在虚拟环境中安装 Flask:

(venv) $ pip install flask

执行上述命令,你就在虚拟环境中安装 Flask 及其依赖了。要想验证 Flask 是否正确安装,

你可以启动 Python 解释器,尝试导入 Flask:

(venv) $ python

>>> import flask

>>>

如果没有看到错误提醒,那恭喜你。

###### easy\_install安装flask

1. sudo easy\_install virtualenv

2、一旦你装上了virtualenv,请调出shell然后创建你自己的环境变量。我通常会创建 一个包含 env 文件夹的项目文件夹:

$ mkdir myproject

$ cd myproject

$ virtualenv env

New python executable in env/bin/python

Installing setuptools............done.

现在,无论何时你想在一个项目上工作,你只需要激活相应的环境。在OS X和Linux上

,执行以下操作:

$ . env/bin/activate

(注意脚本名称和点号之间的空格。该点意味着这个脚本应该运行在当前shell的上下

文。 如果这条命令不能在你的shell中正常工作,请试着把点号替换为 source )

3、easy\_install Flask //是大写的F

4、如果你想要使用最新版本的Flask,有两种方法:你可以使用 easy\_insall 拉出开发版

本, 或者让它来操作一个git检索。无论哪种方式,推荐你使用virtualenv。

在一个新的Virtualenv中获得git检索,并运行在在开发模式下$ git clone http://github.com/mitsuhiko/flask.git

Initialized empty Git repository in ~/dev/flask/.git/

$ cd flask

$ virtualenv env

$ . env/bin/activate

New python executable in env/bin/python

Installing setuptools............done.

$ python setup.py develop

...

Finished processing dependencies for Flask

这将引入依赖关系和激活Git的头作为在Virtualenv中当前的版本。然后你只需 要

pull origin 来获得最新的版本。

git

如果你不想用git来得到最新的开发版,可以改用下面的命令:

$ mkdir flask

$ cd flask

$ virtualenv env

$ . env/bin/activate

New python executable in env/bin/python

Installing setuptools............done.

$ easy\_install Flask==dev

...

Finished processing dependencies for Flask==dev

注：1. 在import flask后，提示

*>>> import flask*

*Traceback (most recent call last):*

*File "<stdin>", line 1, in <module>*

*File "flask/\_\_init\_\_.py", line 21, in <module>*

*from .app import Flask, Request, Response*

*File "flask/app.py", line 26, in <module>*

*from . import json, cli*

*File "flask/cli.py", line 17, in <module>*

*import click*

*ImportError: No module named click*

这是缺少click module ，执行pip install click就可以了。

2. flask安装到系统全局

这样也可以,但是我确实不推荐它。只需以root权限运行 easy\_install

$ sudo easy\_install Flask