



Текстовая расшифровка видео:

РЕЖИМ РАЗРАБОТКИ И ПРОЧИЕ СОВЕТЫ

План:

- Трюк с режимом разработки;
- Советы эксперта;
- Локальный репозиторий.

Трюк с режимом разработки

Трюки:

- `pip install -editable . ;`
- `reload()`.

Когда мы собираем код и собираемся делать из него пакет, мы хотим иметь возможность импортировать этот пакет из сторонних скриптов, если находимся внутри `Virtualenv`.

Как перестать постоянно переустанавливать пакет в новом коде?

Ответ: в Python есть режим разработки. Используем «`pip install -editable .`» для отладочного режима.



```
(slurm_3) meow-nofer@pikachu ~/Experiments/slurm/paperclip master ● ls
build dist docs Jenkinsfile MANIFEST.in paperclip paperclip.egg-info README.md requirements.txt
(slurm_3) meow-nofer@pikachu ~/Experiments/slurm/paperclip master ● pip install --editable .
Obtaining file:///home/meow-nofer/Experiments/slurm/paperclip
  Preparing metadata (setup.py) ... done
Collecting PyYAML==5.1.2
  Using cached PyYAML-5.1.2.tar.gz (265 kB)
  Preparing metadata (setup.py) ... done
Using legacy 'setup.py install' for PyYAML, since package 'wheel' is not installed.
Installing collected packages: PyYAML, paperclip
  Attempting uninstall: PyYAML
    Found existing installation: PyYAML 6.0
    Uninstalling PyYAML-6.0:
      Successfully uninstalled PyYAML-6.0
  Running setup.py install for PyYAML ... done
  Running setup.py develop for paperclip
Successfully installed PyYAML-5.1.2 paperclip-0.0.1
WARNING: You are using pip version 22.0.4; however, version 23.1.2 is available.
You should consider upgrading via the '/home/meow-nofer/.pyenv/versions/3.10.5/envs/slurm_3/bin/python3.10 -m pip install --upgrade pip' command.
(slurm_3) meow-nofer@pikachu ~/Experiments/slurm/paperclip master ●
```

В текущем проекте есть пакет setup.py. Вводим «pip install -editable .», он проверит присутствие всех зависимостей. Если мы посмотрим на зависимости в pip freeze, то увидим следующее:

```
(slurm_3) meow-nofer@pikachu ~/Experiments/slurm/paperclip master ● pip freeze
alabaster==0.7.13
Babel==2.12.1
certifi==2023.5.7
charset-normalizer==3.1.0
docutils==0.20
idna==3.4
imagesize==1.4.1
Jinja2==3.1.2
MarkupSafe==2.1.2
packaging==23.1
-e git+ssh://git@github.com/enchantner/paperclip.git@d734b88935ff05d454cfcec49eec78b5d96461c4#egg=paperclip
Pygments==2.15.1
PyYAML==5.1.2
requests==2.30.0
snowballstemmer==2.2.0
Sphinx==7.0.1
sphinx-argparse==0.4.0
sphinxcontrib-applehelp==1.0.4
sphinxcontrib-devhelp==1.0.2
sphinxcontrib-htmlhelp==2.0.1
sphinxcontrib-jsmath==1.0.1
sphinxcontrib-qthelp==1.0.3
sphinxcontrib-serializinghtml==1.1.5
urllib3==2.0.2
(slurm_3) meow-nofer@pikachu ~/Experiments/slurm/paperclip master ●
[0] 0:[tmux]* "pikachu" 15:43 15-мая-23
```

Pip определил, что мы находимся внутри Git-репозитория и попытался вставить адрес конкретного хэша, конкретного коммита в Git.

Как теперь будет работать импорт пакета?

В Python это решается специальными файлами с ссылками.

В данном примере мы находимся внутри Virtualenv. Рассмотрим, как все выглядит внутри Virtualenv.

Представим, что мы, условно, сделали «install -editable .». Вводим «~/

.pyenv/versions/3.10.5/envs/slurm_3». Внутри этого окружения пакеты ставятся в lib. Так мы можем увидеть все пакеты, которые стоят внутри этого окружения:

```
urllib3==2.0.2
(slurm_3) meow-nofer@pikachu ~/Experiments/slurm/paperclip master ● cd ~/.pyenv/versions/3.10.5/
(slurm_3) meow-nofer@pikachu ~/.pyenv/versions/3.10.5/envs/slurm_3 ls
bin include lib lib64 pyenv.cfg
(slurm_3) meow-nofer@pikachu ~/.pyenv/versions/3.10.5/envs/slurm_3 cd lib/python3.10/site-packages
(slurm_3) meow-nofer@pikachu ~/.pyenv/versions/3.10.5/envs/slurm_3/lib/python3.10/site-packages ls
alabaster Jinja2-3.1.2.dist-info sphinx
alabaster-0.7.13.dist-info markupsafe sphinx-7.0.1.dist-info
babel MarkupSafe-2.1.2.dist-info sphinxarg
Babel-2.12.1.dist-info packaging sphinx_argparse-0.4.0.dist-info
certifi packaging-23.1.dist-info sphinxcontrib
```

```
certifi-2023.5.7.dist-info paperclip.egg-link sphinxcontrib_applehelp-1.0.4.dist-info
charset_normalizer pip sphinxcontrib_devhelp-1.0.2.dist-info
charset_normalizer-3.1.0.dist-info pip-22.0.4.dist-info sphinxcontrib_devhelp-1.0.2-py3.8-nspkg.pth
_distutils_hack pkg_resources sphinxcontrib_htmlhelp-2.0.1.dist-info
distutils-precedence.pth __pycache__ sphinxcontrib_jsmath-1.0.1.dist-info
docutils pygments sphinxcontrib_jsmath-1.0.1-py3.7-nspkg.pth
docutils-0.20.dist-info Pygments-2.15.1.dist-info sphinxcontrib_qthelp-1.0.3.dist-info
easy-install.pth PyYAML-5.1.2-py3.10.egg-info sphinxcontrib_qthelp-1.0.3-py3.8-nspkg.pth
idna requests sphinxcontrib_serializinghtml-1.1.5.dist-info
idna-3.4.dist-info requests-2.30.0.dist-info sphinxcontrib_serializinghtml-1.1.5-py3.9-nspkg.pth
imagesize setuptools urllib3
imagesize-1.4.1.dist-info setuptools-58.1.0.dist-info urllib3-2.0.2.dist-info
imagesize.py snowballstemmer yaml
jinja2 snowballstemmer-2.2.0.dist-info
(slurm_3) meow-nofer@pikachu ~/.pyenv/versions/3.10.5/envs/slurm_3/lib/python3.10/site-packages
[0] 0:python* "pikachu" 15:44 15-мая-23
```

Также здесь есть пакет «paperclip», однако он представлен не каталогом, в котором лежат файлы, а лишь файлом. В этом файле находится путь, по которому интерпретатор будет импортировать этот пакет:

```
(slurm_3) meow-nofer@pikachu ~/.pyenv/versions/3.10.5/envs/slurm_3/lib/python3.10/site-packages cat paperclip.egg-link
/home/meow-nofer/Experiments/slurm/paperclip
(slurm_3) meow-nofer@pikachu ~/.pyenv/versions/3.10.5/envs/slurm_3/lib/python3.10/site-packages
[0] 0:python* "pikachu" 15:44 15-мая-23
```

Теперь, где бы мы не находились, мы можем запустить интерпретатор Python или сказать «import paperclip». Несмотря на то, что пакет формально не установлен внутри Virtualenv, интерпретатор знает, где его искать, что позволяет импортировать и модифицировать его код:

```
(slurm_3) meow-nofer@pikachu ~/.pyenv/versions/3.10.5/envs/slurm_3/lib/python3.10/site-packages cd
(slurm_3) meow-nofer@pikachu ~ python
Python 3.10.5 (main, Jul 19 2022, 13:06:39) [GCC 12.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import paperclip
>>>
[0] 0:python* "pikachu" 15:45 15-мая-23
```

Важно:

Когда код уже запущен, а пакет уже импортирован, вам необходимо перезапустить код или shell, чтобы проверить изменения. При этом не надо переустанавливать пакет.

Есть еще один динамический трюк с reload(), чтобы заново перечитать модуль с диска (не рекомендуется), его проще перезапустить.

Советы эксперта

- При создании Python-пакета используйте импорты от имени пакета.

Пример: «from my_package.config import config». Это позволит видеть уровень иерархии. Помимо этого, IDE легче с этим работать с точки зрения ссылок.

- С отладочным режимом легко упустить добавление файлов MANIFEST.in. Всегда проверяйте вручную.

- Файл `setup.py` в идеале должен лежать в корне проекта, иначе нельзя будет ставить с помощью `pip` напрямую из `git`. Однако можно хранить несколько проектов в одном репозитории.

Локальный репозиторий

Обсудим, как хранить и использовать пакеты в рамках команды.

Чаще всего в корпоративной инфраструктуре уже стоит решение для пакетов и Docker-контейнеров (JFrog Artifactory и т.д.).

В Python тоже есть способ сделать собственный внутренний Python-пакет, чтобы не заливать код, который вы пишете сразу в публичное пространство. В этом поможет [«Pypiserver»](#).

Pypiserver – простейший, примитивный сервер на Python, умеющий сервить пакеты для `pip` и хранить их в локальной файловой системе.

Чтобы ставить из него пакеты, вам необходимо прописать соответствующий конфиг. Нужно прописать адрес репозитория, а также, если у вас не настроен на сервере SSL-сертификат, указать «`trusted-host`»:

```
1. [global]
2. extra-index-url = http://myrepo.example.com:8088/simple/
3. trusted-host = myrepo.example.com
```

Вместо конфига можно писать аргументами командной строки.

Как загружать пакеты?

Если вы используете Poetry, можете использовать Poetry Publish. Тогда он соберется и загрузится.

Чаще для загрузки пакетов используется [Twine](#). **Twine** – специальный метапакет для Python, который нужен для заливки пакетов во внешний репозиторий. Однако для этого нужен отдельный инструмент из-за разных нюансов.

Как вам урок?



Далее >

Слёрм ©

[+7 \(495\) 248-05-80](tel:+7(495)248-05-80)

[Лицензия №ДЛ-1368 от 22.08.2019](#)

[Политика конфиденциальности](#)

[Публичная оферта](#)

