

Маршрут вебинара

Тестовое окружение



Python пакеты



deb пакеты



Рефлексия

Цели вебинара | После занятия вы сможете

1 Собирать и публиковать Python packages

2 Собирать и устанавливать deb packages

3 Эффективно распространять свой код

Смысл | Зачем вам это уметь

1 Настраивать тестовое окружение

2 Решать ряд проблем с Linux-окружением

The background of the slide is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a digital or technological feel. The text "Тестовое окружение" is centered in white, bold, sans-serif font.

Тестовое окружение

Тестовое окружение

Тестовое окружение - аппаратное и программное обеспечение и другие средства, необходимые для выполнения тестов

- Docker container
- Виртуальная машина
- Физический сервер
- Комбинации

Исходный код тестов

Где хранить исходный код тестов?

- Вместе с продуктом
- Отдельно от продукта
- Тесты разных уровней (приемочные, интеграционные) - вместе/отдельно от продукта?

Какой язык использовать для разработки тестов?

- Такой же как язык разработки
- Более удобный в зависимости от инструментария

Где запускать тесты?

Где запускать тесты?

- Локально
- Билд-агенты
- Изолированные окружения (контейнеры, виртуальные машины)

The background of the slide is an aerial photograph of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a web-like pattern. The text 'Python пакеты' is centered in the middle of the slide in a white, sans-serif font.

Python пакеты

PyPI и pip

PyPI (Python Package Index) - репозиторий, содержащий пакеты, которые разрабатываются Python community

pip - пакетный менеджер для Python, который позволяет устанавливать пакеты, которые не входят в стандартную библиотеку Python

Установка

```
$ python3 -m pip install requests
```

```
$ python3 -m pip install requests==2.24.0
```

```
$ python3 -m pip install -i https://custom-pypi.org/simple my_package
```

```
$ python3 -m pip install  
git+https://github.com/psf/requests.git#egg=requests
```

Distribution

Source distribution - формат распространения пакетов, включающий метаданные и исходные файлы, которые используются при установке пакета с помощью `pip` или при генерации `built distribution`

Built distribution - формат распространения пакетов, включающий метаданные и файлы, которые необходимо поместить в определенное место в системе для корректной установки пакета

Source distribution

source distribution - архив с исходными файлами, который используется для самостоятельной установки с помощью **pip** либо для сборки **wheel**

Представляет из себя **tar.gz**-архив

```
$ tar -tf six-1.15.0.tar.gz
```

Сборка

```
$ python3 setup.py sdist
```

Установка

```
$ python3 -m pip install six-1.15.0.tar.gz
```

Wheels

wheels - формат пакета, созданный для распространения модулей и скомпилированных расширений

Представляет из себя **zip**-архив

```
$ unzip -l six-1.15.0-py2.py3-none-any.whl
```

Сборка

```
$ python3 setup.py bdist_wheel
```

Установка

```
$ python3 -m pip install six-1.15.0-py2.py3-none-any.whl
```


Именование wheels

six-1.15.0-py2.py3-none-any.whl

- **six** - имя пакета
- **1.15.0** - версия пакета
- **py2.py3** - Python tag, поддержка Python 2 и Python 3
- **none** - [ABI](#) tag
- **any** - поддерживаются любые платформы

pure-Python wheel - Jinja2-3.0.0a1-**py3-none-any**.whl

universal wheel - chardet-3.0.4-**py2.py3-none-any**.whl

platform wheel - numpy-1.18.4-cp38-cp38-**win32**.whl

Семантическое версионирование

1.15.0

1 - "**Major**" номер версии. Увеличивается, когда API библиотеки или приложения меняется обратно несовместимым образом

15 - "**Minor**" номер версии. Увеличивается, когда в API добавляются новые функции без нарушения обратной совместимости. При увеличении "**Major**" принято сбрасывать "**Minor**" в 0

0 - "**Patch**" номер версии. Увеличивается при исправлении багов, рефакторинге и прочих изменениях, которые ничего не ломают, но и новых фичей не добавляют. При увеличении "**Major**" или "**Minor**" принято сбрасывать "**Patch**" в 0

wheels vs source distributions

wheels

- Устанавливаются быстрее, чем source distributions
- Имеют меньший размер, чем source distributions (six)
<https://pypi.org/project/six/#files>
- При установке wheels не запускается setup.py
- Для установки расширений не требуется компилятор

source distributions

- Можно воспользоваться если соответствующий wheel “сломан”
- Нужно добавить небольшой патч и затем установить пакет

Публикация Python package

Сборка пакета

setuptools, **wheel** - сборка пакета ([setup.py](https://pypi.org/project/setuptools/))

```
$ python3 setup.py sdist bdist_wheel
```

Проверка wheel ([check-wheel-contents](https://pypi.org/project/check-wheel-contents/))

```
$ check-wheel-contents dist/agrpkg-0.0.4-py3-none-any.whl
```

Загрузка пакета

twine - загрузка пакета на PyPI

```
$ python3 -m twine upload --repository testpypi dist/*
```


The image features a background of a dense city skyline, likely New York City, with numerous skyscrapers. The entire image is overlaid with a semi-transparent blue and teal gradient. A network of thin, light blue lines connects various points across the image, creating a digital or technological feel. The text "deb пакеты" is centered in the middle of the image in a white, sans-serif font.

deb пакеты

Пакет

Пакет - архивный файл, содержащий исполняемый двоичный файл, файл конфигурации и иногда информацию о зависимостях

Представляет из себя ar-архив

```
$ ar -t python3-six_1.15.0-2_all.deb
```

- **debian-binary** - текстовый файл, содержащий версию формата deb-пакета
- **control.tar** - tar-архив, содержащий информацию и скрипты установки пакета
- **data.tar** — tar-архив, содержащий дерево устанавливаемых файлов пакета

```
$ dpkg-deb -c python3-six_1.15.0-2_all.deb
```

Менеджер пакетов

Менеджер пакетов позволяет устанавливать, удалять или обновлять любой пакет или группы пакетов, выполняя автоматически все необходимые для этого процедуры (доставку пакетов из удаленных репозиториев, вычисление зависимостей и установку требуемых по ним пакетов, удаление замещаемых пакетов и т. п.)

- **Yum** и **RPM** - Red Hat (Fedora, RHEL, openSUSE)
- **dpkg** - Debian (Ubuntu)
- **pacman** - Arch Linux
- **brew** - OSX

Сборка deb пакетов

fpm - инструмент, который позволяет собирать пакеты для различных систем

```
$ sudo apt-get update
```

```
$ sudo apt-get install ruby-dev build-essential
```

```
$ sudo gem install fpm
```

Сборка python-пакета

```
$ fpm -s python -t deb setup.py
```


Добавление в PATH

Не всегда нужно устанавливать пакеты, иногда достаточно поместить исполняемый файл в директорию, который указан в **PATH**

```
$ echo $PATH
```

```
$ sudo curl -L
```

```
https://github.com/docker/compose/releases/download/1.27.4/docker-compose  
-`uname -s` -`uname -m` -o /usr/local/bin/docker-compose
```

```
$ ll /usr/local/bin/docker-compose
```

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

```
$ docker-compose --version
```

The background of the slide is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue and teal gradient. A network of thin, light blue lines connects various points across the gradient, creating a digital or technological aesthetic. The word "Рефлексия" is centered in the middle of the slide in a large, white, sans-serif font.

Рефлексия

Цели вебинара | После занятия вы сможете

1 Собирать и публиковать Python packages

2 Собирать и устанавливать deb packages

3 Эффективно распространять свой код


Материалы

Документация

- [Python Packaging User Guide](#)
- [Writing the Setup Script](#)
- [setuptools](#)
- [distutils – Building and installing Python modules](#)
- [What Are Python Wheels and Why Should You Care?](#)

Видео

- [The Black Magic of Python Wheels](#)



Заполните, пожалуйста,
опрос о занятии по ссылке в чате



Подготовка тестового окружения



Андрей Гридяев

Ведущий инженер по автоматизации тестирования
andrey.gridyaev@gmail.com