# Виртуальное окружение Python (venv)

Все сторонние пакеты устанавливаются менеджером PIP глобально. Проверить это можно просто командой pip show <имя\_пакета>

root@purplegate:~# pip3 show pytest

Name: pytest

Version: 5.3.2

Summary: pytest: simple powerful testing with Python

Home-page: https://docs.pytest.org/en/latest/

Author: Holger Krekel, Bruno Oliveira, Ronny Pfannschmidt, ...

License: MIT license

Location: /usr/local/lib/python3.8/site-packages

Requires: more-itertools, pluggy, py, wcwidth, attrs, packaging

Required-by:

Location – путь до ваших глобальных пакетов.

Допустим у нас есть два проекта: "Project A" и "Project B". Оба проекта зависят от библиотеки Simplejson. Проблема возникает, когда для "Project A" нужна версия Simplejson 3.0.0, а для проекта "Project B" – 3.17.0. Python не может различить версии в глобальном каталоге site-packages – в нем останется только та версия пакета, которая была установлена последней.

Решение данной проблемы – создание виртуального окружения (virtual environment).

Основная цель виртуального окружения Python – создание изолированной среды для python-проектов.

Это означает, что каждый проект может иметь свои собственные зависимости, независимо от других проектов.

## Настройка виртуального окружения

Один из самых популярных инструментов для создания виртуального окружения – virtualenv. Однако в данной статье мы будем рассматривать более свежий инструмент [venv](https://docs.python.org/3/library/venv.html" \t "_blank).

Устанавливать venv не нужно – он входит в стандартную библиотеку Python

## Создание

Для создания виртуального окружения, перейдите в директорию своего проекта и выполните:

python -m venv venv

Флаг -m указывает Python-у запустить venv как исполняемый модуль. venv/ - название виртуального окружения (где будут храниться ваши библиотеки)

В результате будет создан каталог venv/ содержащий копию интерпретатора Python, стандартную библиотеку и другие вспомогательные файлы.

Новые пакеты будут устанавливаться в venv/lib/python3.x/site-packages/

## Активация

Чтобы начать пользоваться виртуальным окружением, необходимо его активировать:

* venv\Scripts\activate.bat - для Windows;
* source venv/bin/activate - для Linux и MacOS:

source выполняет bash-скрипт без запуска дополнительного bash-процесса.

Проверить успешность активации можно по приглашению оболочки. Она будет выглядеть так:

(venv) root@purplegate:/var/test#

Также новый путь до библиотек можно увидеть выполнив команду:

python -c "import site; print(site.getsitepackages())"

Интересный факт: в виртуальном окружении вместо команды python3 и pip3, можно использовать python и pip

## Автоматическая активация

В некоторых случаях, процесс активации виртуального окружения может показаться неудобным.

На практике, для автоматической активации перед запуском скрипта, создают скрипт-обертку на bash

#!/usr/bin/env bash

source $BASEDIR/venv/bin/activate

python $BASEDIR/my\_app.py

Теперь можно установить права на исполнение и запустить нашу обертку:

chmod +x myapp/run.sh

./myapp/run.sh

## Деактивация

Закончив работу в виртуальной среде, вы можете отключить ее, выполнив консольную команду:

deactivate

## Альтернативы venv

На данный момент существует несколько альтернатив для venv:

* [pipenv](https://pythonchik.ru/osnovy/pipenv-menedzher-zavisimostej-python) - это pipfile, pip и virtualenv в одном флаконе;
* pyenv - простой контроль версий Питона;
* poetry - новый менеджер для управления зависимостями;
* autoenv - среды на основе каталогов;
* pew - инструмент для управления несколькими виртуальными средами, написанными на чистом Python;
* rez - интегрированная система конфигурирования, сборки и развертывания пакетов для программного обеспечения.

Стоит ли использовать виртуальное окружение в своей работе – однозначно да. Это мощный и удобный инструмент изоляции проектов друг от друга и от системы. С помощью виртуального окружения можно использовать даже разные версии Python!

Однако рекомендуем присмотреться к более продвинутым вариантам, например к pipenv или poetry.

Последнее изменение: Четверг, 11 Февраль 2021, 16:29