
venv_tools Documentation

Release 0+untagged.37.g6774553.dirty

James Tocknell

Jun 01, 2017

Contents

1	Usage	3
2	API	5
2.1	venv_tools	5
3	Indices and tables	7
	Python Module Index	9

venv_tools is a collection of BSD Licenced utilities for using venvs from Python.

```
from venv_tools import TemporaryVenv, Venv
from subprocess import call # or similar

with TemporaryVenv() as env_dir, Venv(env_dir):
    call(["pip", "install", "-r", "requirements.txt"])
    call(["python", "-m", "profile", "project"])
```

Contents:

CHAPTER 1

Usage

`venv_tools` contains two context managers, `venv_tools.Venv` and `venv_tools.TemporaryVenv`, which are designed to work together:

```
with TemporaryVenv() as env_dir, Venv(env_dir):
    call(["pip", "install", "-r", "requirements.txt"])
    call(["python", "-m", "profile", "project"])
```

Or, if you already have a `venv`:

```
with Venv(env_dir):
    call(["pip", "install", "-r", "requirements.txt"])
    call(["python", "-m", "profile", "project"])
```

Because `venv_tools` wraps around the `venv.EnvBuilder` API in Python 3.3 and above, it's possible to use more featureful subclasses, such as the example given [here](#).

venv_tools

A bunch of tools for using venvs (and virtualenvs) from python.

copyright

3. 2014 by James Tocknell.

license BSD, see LICENSE for more details.

class `venv_tools.TemporaryVenv` (*venv_builder=None, use_virtualenv=False, python_exe=None, **kwargs*)

Context manager around creating a temporary venv.

TemporaryVenv handles the creation and removal of a temporary venv. By default it will try to use the venv tools in the standard library, and fall back to using [virtualenv](#).

Note: The tool used to create the venv depends on the arguments given. *venv_builder* overrides *use_virtualenv* which overrides the defaults. If *path_to_python_exe* is given, then it is passed to the venv builder, which is chosen as above with the addition that the default will be a tool that supports using a specific python executable (most likely [virtualenv](#)).

Note: If you plan on using pip, you need the argument *with_pip*, as both *EnvBuilder* in the standard library, and the [virtualenv](#) builder included in *venv_tools* (to try to present a similar interface to *EnvBuilder*) default to not installing pip.

Warning: Creating or activating a venv inside a venv can be “interesting”, with the results varying between different python versions (and different venv tools). The safest method appears to be using [virtualenv](#), however this should not be relied upon. A warning will be raised if it is detected that this is running inside a venv.

Parameters

- **python_exe** (*str*) – The path to the python executable (relative or absolute), or the name of the executable on the system path.
- **use_virtualenv** (*bool*) – Use virtualenv instead of the default to create the venv.
- **venv_builder** (*venv.EnvBuilder or similar*) – An object which creates a venv. It must define a method *create* which takes one argument, *env_dir*, and creates a venv at that path. Any additional keywords passed to *Venv* will be passed to the object.

class `venv_tools.Venv` (*env_dir*, *venv_builder=None*, ***kwargs*)

Context manager around activating and deactivating a venv.

Venv sets a number of environment variables which are equivalent to running *bin/activate*. It can create a venv if *venv_builder* is given.

Warning: Creating or activating a venv inside a venv can be “interesting”, with the results varying between different python versions (and different venv tools). The safest method appears to be using virtualenv, however this should not be relied upon. A warning will be raised if it is detected that this is running inside a venv.

Parameters

- **env_dir** (*str*) – The absolute path to the venv.
- **venv_builder** (*venv.EnvBuilder or similar*) – An object which creates a venv. It must define a method *create* which takes one argument, *env_dir*, and creates a venv at that path. Any additional keywords passed to *Venv* will be passed to the object.

call_python_code (*code*)

Call some python code with the python interpreter associated with this virtualenv.

call_python_file (*filename*)

Call a python file with the python interpreter associated with this virtualenv.

call_python_module (*module_name*, **args*)

Call a python module with the python interpreter associated with this virtualenv.

env_dir

The path to the virtual environment

install_command

The command to install a python package in the virtualenv. Must be a format string with python and package.

install_package (*package*)

Install a python package into this virtualenv.

python_exe

Path to python interpreter

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

V

`venv_tools`, 5

C

`call_python_code()` (`venv_tools.Venv` method), 6
`call_python_file()` (`venv_tools.Venv` method), 6
`call_python_module()` (`venv_tools.Venv` method), 6

E

`env_dir` (`venv_tools.Venv` attribute), 6

I

`install_command` (`venv_tools.Venv` attribute), 6
`install_package()` (`venv_tools.Venv` method), 6

P

`python_exe` (`venv_tools.Venv` attribute), 6

T

`TemporaryVenv` (class in `venv_tools`), 5

V

`Venv` (class in `venv_tools`), 6
`venv_tools` (module), 5