How to install packages using pip according to the requirements.txt file from a local directory?

Ask Question

755

217

Here is the problem

I have a requirements.txt that looks like:

```
BeautifulSoup==3.2.0
Django==1.3
Fabric==1.2.0
Jinja2==2.5.5
PyYAML==3.09
Pygments==1.4
SQLAlchemy==0.7.1
South==0.7.3
amqplib==0.6.1
anyjson==0.3
...
```

I have a local archive directory containing all the packages + others.

I have created a new virtualenv with

```
bin/virtualenv testing
```

upon activating it, I tried to install the packages according to requirements.txt from the local archive directory.

```
source bin/activate
pip install -r /path/to/requirements.txt -f file:///path/to/archive/
```

I got some output that seems to indicate that the installation is fine

```
Downloading/unpacking Fabric==1.2.0 (from -r ../testing/requirements.txt (line
3))
  Running setup.py egg_info for package Fabric
    warning: no previously-included files matching '*' found under directory
'docs/_build'
    warning: no files found matching 'fabfile.py'
Downloading/unpacking South==0.7.3 (from -r ../testing/requirements.txt (line 8))
  Running setup.py egg_info for package South
```

installed properly. I cannot import the package, and none is found in the site-packages directory of my virtualenv. So what went wrong?

python virtualenv pip

edited May 15 at 11:56

kvmahesh
1,869 ● 2 ● 8 ● 21

asked Aug 29 '11 at 3:53

kakarukeys
5,266 ● 6 ● 24 ● 39

3 Have you tried using `--no-index` so it does not even look at PyPI? – Hugo Tavares Sep 8 '11 at 5:15

1 You've snipped the error message. It's generally near the bottom. – bukzor May 5 '14 at 14:47

You could use the file paths instead of the package names, it means you'll have to change the requirements.txt file, but keep the pip install command the same. – asafge Nov 18 '14 at 8:41

4 **The most votted answer (Mike Lyons's one) does not even answer the question**. Could you please accept the codeape's answer which is the correct one? – Piotr Dobrogost Mar 25 '16 at 7:41

2 surprised nobody else linked but pip.pypa.io/en/stable/user_guide/#requirements-files – MrMesees Apr 25 '16 at 8:17

9 Answers

492

This works for me:

```
$ pip install -r requi
```

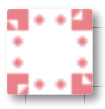
`--no-index` - Ignore package index (only looking at `--find-links` URLs instead).

`-f, --find-links <URL>` - If a URL or path to an html file, then parse for links to archives. If a local path or `file://` URL that's a directory, then look for archives in the directory listing.

edited Jan 26 '17 at 10:45

Paweł Prażak
2,268 ● 17 ● 34

answered May 3 '12 at 10:04



codeape
71k ● 20 ● 121 ● 147

6 **Information on `--no-index` from command pip help install** `--no-index` Ignore package index (only looking at `--find-links` URLs instead). **Information on `--find-links` from command pip help install** `-f` , `--find-links <url>` If a url or path to an html file, then parse for links to archives. If a local path or file:// url

that's a directory, then look for archives in the directory listing. – AWrightIV Apr 3 '15 at 18:54

1 // , This could be a very elegant solution, especially given the eternal struggle with vendorizing: bitprophet.org/blog/2012/06/07/on-vendorizing – Nathan Basanese Apr 4 '16 at 6:40

One caution with this is you may `pip install <some_module>` without using `requirements.txt` but that will not update `requirements.txt`. An alternative might be updating a docker such that it lists all the pip install commands that are run to install dependencies. – ampersands Feb 22 at 23:05

1065

I've read the above, realize this is an old question, but it's totally unresolved and still at the top of my google search results so here's an answer that works for everyone:

```
pip install -r /path/t
```

edited Dec 7 at 0:55

answered Mar 24 '13 at 0:47

Mike Lyons
11k ● 2 ● 9 ● 14

207 I realize this answer

doesn't address the original question, but it answers the question I had when I found this on Google... – Jonathan Dec 15 '13 at 5:50

8 @MikeLyons pretty sure the question hasn't in fact changed - there's no edit history. – Mark Amery Nov 6 '15 at 16:37

28 for those who like to know what they type, the -r in the command just means "install from a requirements file", and is a shortcut for --requirement – Florent Chatterji Oct 6 '16 at 14:04

1 @FlorentChatterji I've always wondered about that. Thanks! – rayryeng Mar 9 '17 at 22:59

1 When I want to upgrade a package (e.g. Django), I change the version in my requirements.txt file and then run `pip install -r /path/to/requirements.txt`. This detects the change, upgrades the package, and leaves everything else alone. – User May 1 '17 at 8:55

105 For virtualenv to install all files in the requirements.txt file.

1. cd to the directory where requirements.txt is located
2. activate your

virtualenv

3. run: `pip install -r requirements.txt` in your shell

answered Feb 26 '14 at 15:38

Aaron Lelevier
8,901 ● 5 ● 47 ● 75

3 "pyenv virtualenv pip install requirements.txt not working" for the sake of Google... – Jerad Jul 23 '16 at 4:17

46

I had a similar problem. I tried this:

```
pip install -U -r requi
```

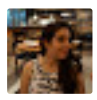
(-U = update if it had already installed)

But the problem continued. I realized that some of generic libraries for development were missed.

```
sudo apt-get install li
dev tcl8.6-dev tk8.6-de
```

I don't know if this would help you.

answered Jan 23 '16 at 22:08



enriquetaso
571 ● 4 ● 6

22

```
pip install -r requirem
```

For further details please check the help option.

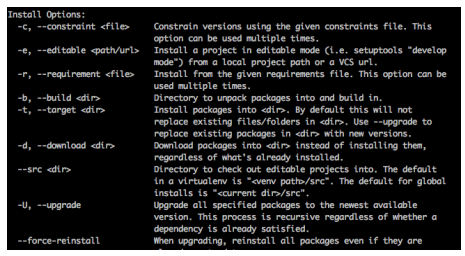
```
pip install --help
```

We can find the option '-r'

> -r, --requirement
> Install from the given

requirements file. This option can be used multiple times.

Further information on some commonly used pip install options: (This is the help option on pip install command)



```
Install Options:
  -c, --constraint <file>     Constrain versions using the given constraints file. This
                              option can be used multiple times.
  -e, --editable <path/url>   Install a project in editable mode (i.e. setuptools "develop
                              mode") from a local project path or a VCS url.
  -r, --requirement <file>    Install from the given requirements file. This option can be
                              used multiple times.
  -b, --build <dir>           Directory to unpack packages into and build in.
  -t, --target <dir>          Install packages into <dir>. By default this will not
                              replace existing files/folders in <dir>. Use --upgrade to
                              replace existing packages in <dir> with new versions.
  -d, --download <dir>        Download packages into <dir> instead of installing them,
                              regardless of what's already installed.
  --src <dir>                 Directory to check out editable projects into. The default
                              in a virtualenv is "<venv path>/src". The default for global
                              installs is "<current dir>/src".
  -U, --upgrade               Upgrade all specified packages to the newest available
                              version. This process is recursive regardless of whether a
                              dependency is already satisfied.
  --force-reinstall           When upgrading, reinstall all packages even if they are
```

Also the above is the complete set of options. Please use pip install --help for complete list of options.

edited Nov 9 '17 at 3:23

J4cK

20.2k ● 5 ● 30 ● 49

answered Sep 16 '16 at 17:23

Py_minion

1,070 ● 1 ● 7 ● 14

3 "Read the manual" answers are not helpful. – shrewmouse Jul 26 at 22:07

14

Often, you will want a fast install from local archives, without probing PyPI.

First, download the archives that fulfill your requirements:

```
$ pip install --downloa
```

Then, install using `–find-links` and `–no-index`:

```
$ pip install --no-inde
```

edited Nov 25 '15 at 17:06

mkobit

20.3k ● 6 ● 84 ● 99

answered Sep 22 '15 at 6:27

Jadav Bheda

2,643 ● 1 ● 17 ● 24

10

I work with a lot of systems that have been mucked by developers "following directions they found on the internet". It is extremely common that your `pip` and your `python` are not looking at the same paths/site-packages. For this reason, when I encounter oddness I start by doing this:

```
$ python -c 'import sys
['', '/usr/lib/python2.
'/usr/lib/python2.7/lib
'/usr/lib/python2.7/dis

$ pip --version
pip 9.0.1 from /usr/loc
```

That is a *happy system*.

Below is an *unhappy system*. (Or at least it's a blissfully ignorant system that causes others to be unhappy.)

```
$ pip --version
pip 9.0.1 from /usr/loc

$ python -c 'import sys
['',
'/usr/local/Cellar/pyth

'/usr/local/Cellar/pyth

'/usr/local/Cellar/pyth
darwin',
'/usr/local/Cellar/pyth
mac',
'/usr/local/Cellar/pyth
mac/lib-scriptpackages'
'/usr/local/Cellar/pyth
tk',
'/usr/local/Cellar/pyth
old',
```

```
'/usr/local/Cellar/pyth
dynload',
'/usr/local/lib/python2

$ which pip pip2 pip3
/usr/local/bin/pip
/usr/local/bin/pip3
```

It is *unhappy* because `pip` is (python3.6 and) using `/usr/local/lib/python3.6/site-packages` while `python` is (python2.7 and) using `/usr/local/lib/python2.7/site-packages`

When I want to make sure I'm installing requirements to the *right* python, I do this:

```
$ which -a python pytho
/usr/local/bin/python
/usr/bin/python
/usr/local/bin/python2
/usr/local/bin/python3

$ /usr/bin/python -m pi
```

You've heard, "If it ain't broke, don't try to fix it." The DevOps version of that is, "If you didn't break it and you can work around it, don't try to fix it."

edited Apr 8 '17 at 23:13

RedBlueThing
33.1k ● 16 ● 84 ● 113

answered Mar 23 '17 at 16:49

Bruno Bronosky
33.4k ● 4 ● 77 ● 80

5

Installing requirements.txt file inside virtual env with python 3:

I had the same issue. I was trying to install requirements.txt file

inside a virtual environament. I found the solution.

Initially, I created my virtual env in this way:

```
virtualenv -p python3 m
```

Activate the environment using:

```
source myenv/bin/activa
```

Now I installed the requirements.txt using:

```
pip3 install -r require
```

Installation was successful and I was able to import the modules.

edited Jul 24 at 6:24

answered Nov 10 '17 at 8:05

janu777

1,366 ● 4 ● 18

1 Doesn't answer the question – ivarec May 24 at 15:34

I faced this issue while installing the requirements file inside a virtual environment. I have now modified my response to address the issue – janu777 Jul 24 at 6:28

1 Using Anaconda Python 3.6 on Windows, I had to do `virtualenv -p python myenv` , `myenv\Scripts\activate.bat` , `pip install -r requirements.txt` – bdforbes Aug 25 at 6:13

1 Actually, on the contrary, this seems like the only answer that *does* answer the part of the question about installing locally defined deps in `requirements.txt` to a virtualenv. Unless I am missing something? Anyway, thanks! – davnicwil Sep 19 at 17:39

```
pip install -r /path/to
```

1

Here, `-r` is short form of `--requirement` and it asks the `pip` to install from the given `requirements` file.

`pip` will start installation only after checking the availability of all listed items in the `requirements` file and it won't start installation even if one `requirement` is unavailable.

One workaround to install the available packages is installing listed packages one by one. Use the following command for that. A red color warning will be shown to notify you about the unavailable packages.

```
cat requirements.txt |
```

answered Dec 11 at 5:08

Safwan
1,026 ● 11 ● 17

---

protected by Bhargav Rao ♦ Oct 24 '15 at 16:39

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?