From conda create requirements.txt for pip3

Asked 4 years, 2 months ago Modified 1 month ago Viewed 181k times

155

60

Hi I usually use conda to manage my environments, but now I am on a project that needs a little more horsepower than my laptop. So I am trying to use my university's workstations which have new Intel Xeons. But I don't have admin rights and the workstation does not have conda so I am forced to work with virtualenv and pip3.

How do I generate a `requirements.txt` from conda that will work with `pip3` and `venv`?

```
conda list -e > requirements.txt
```

does not generate a compatible file:

```
= is not a valid operator. Did you mean == ?
```

The `conda` output is:

```
# This file may be used to create an environment using:
# $ conda create --name <env> --file <this file>
# platform: osx-64
certifi=2016.2.28=py36_0
cycler=0.10.0=py36_0
freetype=2.5.5=2
icu=54.1=0
libpng=1.6.30=1
matplotlib=2.0.2=np113py36_0
mkl=2017.0.3=0
numpy=1.13.1=py36_0
openssl=1.0.2l=0
pip=9.0.1=py36_1
pyparsing=2.2.0=py36_0
pyqt=5.6.0=py36_2
python=3.6.2=0
python-dateutil=2.6.1=py36_0
pytz=2017.2=py36_0
qt=5.6.2=2
readline=6.2=2
scikit-learn=0.19.0=np113py36_0
scipy=0.19.1=np113py36_0
setuptools=36.4.0=py36_1
sip=4.18=py36_0
six=1.10.0=py36_0
sqlite=3.13.0=0
tk=8.5.18=0
wheel=0.29.0=py36_0
xz=5.2.3=0
zlib=1.2.11=0
```

I thought I would just manually change all `=` to `==` but the there are two `=` in the conda output. Which one to change? Surely there is an easier way?

EDIT: `pip freeze > requirements.txt` gives:

```
certifi==2016.2.28
cycler==0.10.0
matplotlib==2.0.2
matplotlib-venn==0.11.5
numpy==1.13.1
pyparsing==2.2.0
python-dateutil==2.6.1
pytz==2017.2
scikit-learn==0.19.0
scipy==0.19.1
six==1.10.0
```

python-3.x pip conda

Share Edit Follow Flag

edited Jun 9, 2018 at 19:50

asked Jun 9, 2018 at 19:35

ITA
2,460 2 16 30

7 you can use 'pip freeze' on your computer – Evgeny Jun 9, 2018 at 19:41

But the outputs are different, I mean the conda command has more packages than output of `pip freeze` . – ITA Jun 9, 2018 at 19:45

that looks a bit strange, but usually you can edit the rewuiremrnts.txt by hand as last resort. another option in pipenv, a wrapper around pip and virtualenv – Evgeny Jun 9, 2018 at 19:53

1 Nope, none if it is working. I find it so strange that this can't be done. Yeah maybe its uncommon for people to switch between environment managers, but still ... the need *is* conceivable. – ITA Jun 9, 2018 at 20:10

2 I'm not sure, I don't see a full answer; in the sense you have explained what the tools do, but there is no solution as far as I see in *how* to replicate a conda environment successfully with virtualenv of pipenv. – ITA Jun 17, 2018 at 15:11

7 Answers

Sorted by:
Trending sort available

Highest score (default)

As the comment at the top indicates, the output of

252

```
conda list -e > requirements.txt
```

can be used to create a `conda` virtual environment with

```
conda create --name <env> --file requirements.txt
```

but this output isn't in the right format for `pip`.

If you want a file which you can use to create a `pip` virtual environment (i.e. a `requirements.txt` in the right format) you can install `pip` within the `conda` environment, then use pip to create `requirements.txt`.

```
conda activate <env>
conda install pip
pip freeze > requirements.txt
```

Then use the resulting `requirements.txt` to create a `pip` virtual environment:

```
python3 -m venv env
source env/bin/activate
pip install -r requirements.txt
```

When I tested this, the packages weren't identical across the outputs (`pip` included fewer packages) but it was sufficient to set up a functional environment.

For those getting odd path references in requirements.txt, use:

```
pip list --format=freeze > requirements.txt
```

Share Edit Follow Flag

edited Feb 22 at 5:36

answered Sep 8, 2019 at 20:11

rmwenz

2,664 1 7 7

15 This is the best answer and should be the accepted answer, because it provides specific commands to use to accomplish / answer the original SO question. – Rich Lysakowski PhD Mar 23, 2020 at 3:30

24 For those following this thread and getting odd path references in `requirements.txt`, use `pip list --format=freeze > requirements.txt` instead. Ref: stackoverflow.com/a/62886215/13095028 – tnwei Feb 5, 2021 at 2:17

1 @JohnnyUtah careful there. Creating a totally empty environment won't include `pip` and so your recommendation would end up installing all those packages *outside* the intended target environment. Besides, the OP clearly is asking about moving to a system that doesn't have Conda in the first place, hence the turn to `pip` and `venv`. – merv Apr 12, 2021 at 3:36

1 Ah good point @merv. Could you explicitly install `pip` after creating and activating the Conda environment and before running `pip install -r requirements.txt`? – JohnnyUtah Apr 14, 2021 at 15:48

3 If you are using conda the best option is to use `conda env export > environment.yml` , commit your environment.yml and then `conda env create -f environment.yml` . – Playing With BI Nov 3, 2021 at 13:39

I

36

In a conda environment with simply calling

```
pip freeze
```

I got:

```
ipykernel @ file:///C:/ci/ipykernel_1607454116140/work/dist/ipykernel-5.3.4-py3-
none-any.whl
ipython @ file:///D:/bld/ipython_1612487184680/work
...
```

Wanted format:

```
ipykernel==5.3.4
ipython==7.20.0
...
```

In an activated conda environment I had to use

```
pip list --format=freeze
```

to get the correct format for generating a requirements file for people who prefer to use pip with virtual environments.

Share Edit Follow Flag

answered Apr 30, 2021 at 14:14

Pascal
530 5 15

6 Works perfectly! Just use `pip list --format=freeze > requirements.txt` .To store package list as requirements.txt file in the project directory. – Pam Cesar Jun 1, 2021 at 4:02

27

Following the discussion, I'd like to mention that you can actually see some separation of `pip` and `conda` roles.

`pip` is a standard package manager, it does one thing and does it well. `requirements.txt` can be generated in one environment and installed by `pip` in a new environment.

Now there is `conda` output: you rightfully capture their comment which says *'we generated this list of libraries to work with conda'*. Note that python itself is in the conda list and (properly) not in `requirements.txt`. `conda` replicates own installation, that is why its list of libraries is longer, and has python itself.

`pip` produces a list of packages that were installed on top of standard library to make the package you wrote work. Hope it helps to distinguish between the two.

Also pipenv is a newer tool, that can do both virtual environment and package management for you.

Share Edit Follow Flag

edited Feb 28, 2020 at 1:00

Martijn Pieters ♦
981k 271 3866 3224

answered Jun 10, 2018 at 10:54

Evgeny
3,867 2 17 35

14 pipenv is still a tool on top of the Python base, just like pip is. It will not handle requirements *outside* of Python. That's what conda is for, to handle that larger picture, see What is the difference between pip and conda? – Martijn Pieters ♦ Jul 19, 2018 at 10:17

1 While this answer provides some additional useful information, it does not provide actual commands to use with conda and pip. – Rich Lysakowski PhD Mar 23, 2020 at 3:29

22

Just in case someone is looking to generate requirements.txt from an existing project in conda, use following

- Go to your project environment `conda activate <env_name>`
- `conda list` *gives you list of packages used for the environment*
- `conda list -e > requirements.txt` *save all the info about packages to your folder*
- `conda env export > <env_name>.yml`
- `pip freeze`

Share Edit Follow Flag

answered Aug 13, 2019 at 15:02

Aishwat Singh
4,147 2 25 45

6 The OP wanted a usable requirements.txt file. How does this procedure help them get that? – mabraham Feb 27, 2020 at 9:13

4 This answer is very useful because it provides the exact package list and commands needed to create the yaml file to recreate the conda environment, not just the list of pip packages. – Rich Lysakowski PhD Mar 23, 2020 at 9:27

Aishwat thanks for sharing. can you explain how does this magic happen? – SKSKSKSK Dec 2, 2020 at 10:28

activate the conda env

`conda activate flask-test`

get the path of the conda env and copy it

`conda list`

append the copied path with `lib\site-packages` and use it in pip with `--path` option

`pip freeze --path C:\Users\username\Miniconda3\envs\flask-test\lib\site-packages > requirements.txt` on Linux the path is like `/home/username/miniconda3/envs/flask-app/lib/python3.8/site-packages/`

2

Share Edit Follow Flag

edited Aug 10, 2020 at 3:47

answered Aug 3, 2020 at 13:13

Edgar Manukyan

913 8 18

As mentioned in the comments above, the correct full answer is:

```
pip list --format=freeze > requirements.txt
```

0

Share Edit Follow Flag

answered May 13 at 0:19

MattG

4,700 4 29 45

I made some simple python script to convert conda requirements.txt for pip3. Just copy below code.

```
f = open(r"requirements.txt", "r").read()
f1 = f.split("\n")
for line in f1:
    if "=" in line:
       a = line.split("=")[0:-1]
       print(a[0]+"==" + a[1])
```

0

then just copy the printed output;

Sample output:

```
_libgcc_mutex==0.1
_openmp_mutex==4.5
blas==1.0
brotli==1.0.9
bzip2==1.0.8
```

Share Edit Follow Flag

answered Jul 17 at 7:47

tensordt

1