Python pipenv

Summary: in this tutorial, you'll learn how to configure a project with a new virtual environment using the Python `pipenv` tool.

Creating a new project

First, create a new project folder e.g., `crawler` .

Second, navigate to the `crawler` folder and install the `requests` package using the `pipenv` command:

```
pipenv install requests
```

Output:

```
Creating a Pipfile for this project…
Installing requests…
Adding requests to Pipfile's [packages]…
Installation Succeeded
Pipfile.lock not found, creating…
```

```
Locking [dev-packages] dependencies…
Locking [packages] dependencies…
Locking...Building requirements...
Resolving dependencies...
Success!
Updated Pipfile.lock (fbd99e)!
Installing dependencies from Pipfile.lock (fbd99e)…
================================ 0/0 - 00:00:00
```

And you'll see that `pipenv` created two new files called `Pipfile` and `Pipfile.lock` . On top of that, it installed a virtual environment.

If you look at the project folder, you won't see the virtual environment folder.

To find the location of the virtual environment, you use the following command:

```
pipenv --venv
```

It'll return something like this on Windows:

```
C:\Users\<username>\.virtualenvs\crawler-7nwusESR
```

Note that the `<username>` is the username that you use to login to the Windows.

Third, create a new file called `app.py` in the project folder and add the following code to the file:

```
import requests

response = requests.get('https://www.python.org/')
print(response.status_code)
```

In this code, we imported the `requests` third-party module, use the `get()` function to make an HTTP request to the URL `https://www.python.org/` and display the status code (`200`).

Fourth, run the `app.py` file from the terminal by using the python command:

```
python app.py
```

It'll show the following error:

```
ModuleNotFoundError: No module named 'requests'
```

The reason is that Python couldn't locate the new virtual environment. To fix this, you need to activate the virtual environment.

Fifth, use the following command to activate the new virtual environment:

```
pipenv shell
```

If you run the `app.py` now, it should work correctly:

```
python app.py
```

Output:

```
200
```

The status code 200 means the HTTP request has been succeeded.

Sixth, use the `exit` command to deactivate the virtual environment:

```
exit
```

Resolving the Unresolved Import Warning in VS Code

If you're using VS Code, you may receive the unresolved import warning. The reason is that the VS code doesn't know which Python interpreter to use.

Therefore, you need to switch the Python interpreter to the one located in the new virtual environment:

First, click the current Python interpreter at the right bottom corner of the VS Code:

Second, select the Python interpreter from the list:

In addition, you need to change the `python.jediEnabled` parameter in the `settings.json` to `True` :

To open the settings.json file, you open the Command Palette with the keyboard shortcut `CTRL + SHIFT + P` on Windows or `CMD + SHIFT + P` on macOS:

And the change the value to True as follows:

After that, you should save the file and restart the VS Code for the change to take effect.