

Lesson Description - Using Third-Party Packages in Your Scripts

Now that we know how to install third-party code, it's time to learn how to use it in our scripts.

Python Documentation For This Video

- [The requests package](#)
- [The argparse.ArgumentParser class](#)
- [The os.getenv function](#)
- [The sys.exit function](#)

Creating a Weather Script

We're going to write up the start of a script that can provide us with weather information using data from openweathermap.org. For this video, we're going to be installing another package called `requests`. This is a nice package for making web requests from Python and one of the most used Python packages. You will need to get your API key from [OpenWeatherMap](https://openweathermap.org/api) to follow along with this video.

Let's start off by activating the `experiment` virtualenv that we created in the previous video. Install the package and set an environment variable with an API key:

```
$ source ~/venvs/experiment/bin/activate
(experiment) $ pip install requests
(experiment) $ export OWM_API_KEY=[YOUR API KEY]
```

Create a new script called `weather`:

`~/bin/weather`

```
#!/usr/bin/env python3.6

import os
import requests
import sys

from argparse import ArgumentParser
```

```

parser = ArgumentParser(description='Get the current weather
information')
parser.add_argument('zip', help='zip/postal code to get the weather
for')
parser.add_argument('--country', default='us', help='country zip/
postal belongs to, default is "us"')

args = parser.parse_args()

api_key = os.getenv('OWM_API_KEY')

if not api_key:
    print("Error: no 'OWM_API_KEY' provided")
    sys.exit(1)

url = f"http://api.openweathermap.org/data/2.5/weather?
zip={args.zip},{args.country}&appid={api_key}"

res = requests.get(url)

if res.status_code != 200:
    print(f"Error talking to weather provider: {res.status_code}")
    sys.exit(1)

print(res.json())

```

Notice that we were able to use the `requests` package in the same way that we would any package from the standard library.

Let's try it out:

```

(experiment) $ chmod u+x ~/bin/weather
(experiment) $ weather 45891
(experiment) ~ $ weather 45891
{'coord': {'lon': -84.59, 'lat': 40.87}, 'weather': [{'id': 801,
'main': 'Clouds', 'description': 'few clouds', 'icon': '02d'}],
'base': 'stations', 'main': {'temp': 282.48, 'pressure': 1024,
'humidity': 84, 'temp_min': 282.15, 'temp_max': 283.15},
'visibility': 16093, 'wind': {'speed': 1.5, 'deg': 210}, 'clouds':
{'all': 20}, 'dt': 1517169240, 'sys': {'type': 1, 'id': 1029,
'message': 0.0043, 'country': 'US', 'sunrise': 1517143892, 'sunset':
1517179914}, 'id': 0, 'name': 'Van Wert', 'cod': 200}

```

Making `weather` Work Regardless of the Active Virtualenv

Currently, our `weather` script will only work if the `experiment` virtualenv is active since no other Python has `requests` installed. We can get around this by changing the shebang to point to the specific python within our virtualenv:

Make this script work regardless of active python by using this as the shebang:

```
#!/home/$USER/venvs/experiment/python
```

You'll need to substitute in your actual username for `$USER`. Here's what the script looks like on a cloud server with the username of `user`:

`~/bin/weather`

```
#!/home/user/venvs/experiment/bin/python

import os
import requests
import sys

from argparse import ArgumentParser

parser = ArgumentParser(description='Get the current weather
information')
parser.add_argument('zip', help='zip/postal code to get the weather
for')
parser.add_argument('--country', default='us', help='country zip/
postal belongs to, default is "us"')

args = parser.parse_args()

api_key = os.getenv('OWM_API_KEY')

if not api_key:
    print("Error: no 'OWM_API_KEY' provided")
    sys.exit(1)

url = f"http://api.openweathermap.org/data/2.5/weather?
zip={args.zip},{args.country}&appid={api_key}"

res = requests.get(url)

if res.status_code != 200:
    print(f"Error talking to weather provider: {res.status_code}")
    sys.exit(1)
```

```
print(res.json())
```

Now if we deactivate and use the script it will still work:

```
(experiment) $ deactivate
$ weather 45891
{'coord': {'lon': -84.59, 'lat': 40.87}, 'weather': [{'id': 801,
'main': 'Clouds', 'description': 'few clouds', 'icon': '02d'}],
'base': 'stations', 'main': {'temp': 282.48, 'pressure': 1024,
'humidity': 84, 'temp_min': 282.15, 'temp_max': 283.15},
'visibility': 16093, 'wind': {'speed': 1.5, 'deg': 210}, 'clouds':
{'all': 20}, 'dt': 1517169240, 'sys': {'type': 1, 'id': 1029,
'message': 0.0035, 'country': 'US', 'sunrise': 1517143892, 'sunset':
1517179914}, 'id': 0, 'name': 'Van Wert', 'cod': 200}
```

Take is as a challenge to build on this example to make a more full-featured weather CLI.