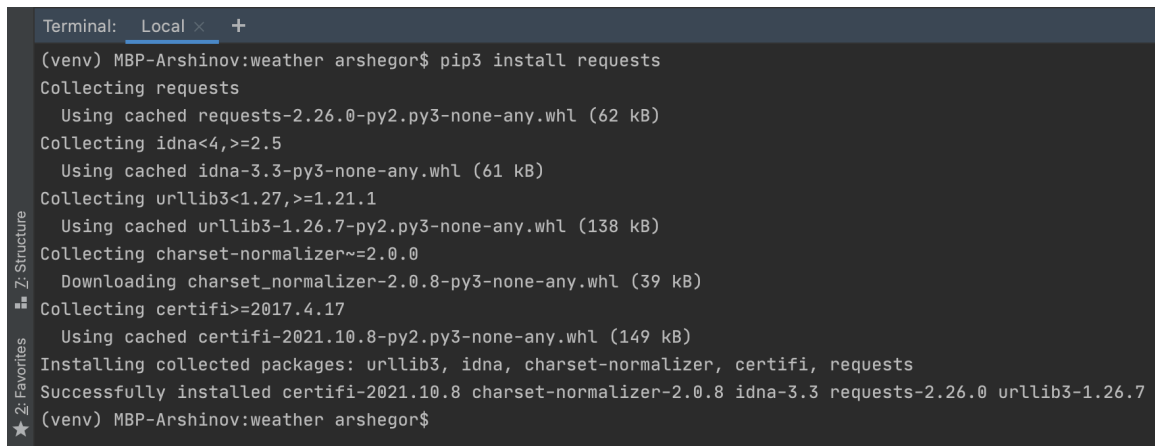


## Лабораторная работа №2: Создание приложения с метео-информацией

---

1. Создаем новый проект в PyCharm
2. Устанавливаем библиотеку requests
  - В терминале PyCharm вводим следующую команду:  
**pip3 install requests**



```
Terminal: Local x +
(venv) MBP-Arshinov:weather arshegor$ pip3 install requests
Collecting requests
  Using cached requests-2.26.0-py2.py3-none-any.whl (62 kB)
Collecting idna<4,>=2.5
  Using cached idna-3.3-py3-none-any.whl (61 kB)
Collecting urllib3<1.27,>=1.21.1
  Using cached urllib3-1.26.7-py2.py3-none-any.whl (138 kB)
Collecting charset-normalizer~=2.0.0
  Downloading charset_normalizer-2.0.8-py3-none-any.whl (39 kB)
Collecting certifi>=2017.4.17
  Using cached certifi-2021.10.8-py2.py3-none-any.whl (149 kB)
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2021.10.8 charset-normalizer-2.0.8 idna-3.3 requests-2.26.0 urllib3-1.26.7
(venv) MBP-Arshinov:weather arshegor$
```

Библиотека requests является стандартным инструментом для составления HTTP-запросов в Python. Простой и аккуратный API значительно облегчает трудоемкий процесс создания запросов.

3. Регистрируемся на сайте [openweathermap.org](https://openweathermap.org)
4. В файле main.py импортируем библиотеку requests

**import requests**

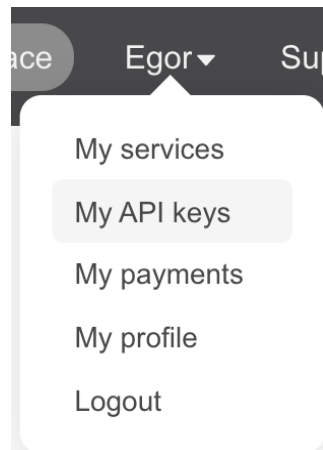
5. Далее запоминаем в переменные город, который нас интересует и APPID

city = "Moscow,RU"

appid = "Ваш APPID"

### Получение APPID:

- Заходим в **My API keys**



- Копируем ключ

Key	Name	
7080b85d03b618cd74a84a0fa43249d3	Default	

### 6. Отправляем запрос на сервис и получаем данные

```
res = requests.get("http://api.openweathermap.org/data/2.5/weather",  
    params={'q': s_city, 'units': 'metric', 'lang': 'ru', 'APPID': appid})  
data = res.json()
```

GET является одним из самых популярных HTTP методов. Метод GET указывает на то, что происходит попытка извлечь данные из определенного ресурса. Для того, чтобы выполнить запрос GET, используется `requests.get()`.

Параметр `q` используется для указания города.

Параметр `units` используется для указания системы измерений (нам подойдет метрическая).

Параметр `lang` используется для указания языка отображения данных.

Параметр `APPID` необходимо указать, чтобы сервис не отклонил наш запрос, а принял нас как зарегистрированных пользователей.

JSON построен на двух структурах:

- Набор пар «имя-значение». Они могут быть реализованы как объект, запись, словарь, хеш-таблица, список

«ключей-значений» или ассоциативный массив.

- Упорядоченный список значений. Его реализуют в виде массива, вектора, списка или последовательности.

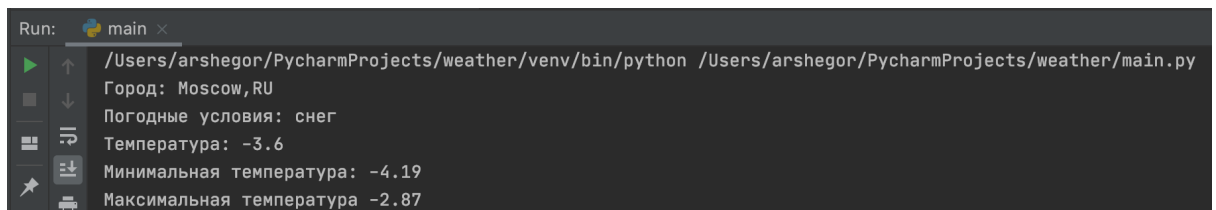
Для сохранения результатов используем переменную data. Так как сервис отдает нам информацию в формате json, нам необходимо использовать метод json() для декодирования информации полученной от сервиса.

## 7. Выводим информацию в удобном для восприятия виде

```
print("Город:", s_city)
print("Погодные условия:", data['weather'][0]['description'])
print("Температура:", data['main']['temp'])
print("Минимальная температура:", data['main']['temp_min'])
print("Максимальная температура", data['main']['temp_max'])
```

Так как в переменной data хранится словарь json, то данные из него можно брать подобно обычному словарю Python.

## 8. Проверяем работу программы



```
Run: main x
/Users/arshegor/PycharmProjects/weather/venv/bin/python /Users/arshegor/PycharmProjects/weather/main.py
Город: Moscow, RU
Погодные условия: снег
Температура: -3.6
Минимальная температура: -4.19
Максимальная температура -2.87
```

## 9. Узнаем прогноз погоды на неделю

```
res = requests.get("http://api.openweathermap.org/data/2.5/forecast",
                    params={'q': s_city, 'units': 'metric', 'lang': 'ru', 'APPID': appid})
data = res.json()
print("Прогноз погоды на неделю:")
for i in data['list']:
    print("Дата <", i['dt_txt'], "> \r\nТемпература <",
          '{0:3.0f}'.format(i['main']['temp']), "> \r\nПогодные условия <",
          i['weather'][0]['description'], ">")
    print("_____")
```

Чтобы узнать прогноз погоды на неделю нам необходимо отправлять запрос не на путь "/weather", а на путь "/forecast"

**Домашнее задание:**

- Изучить документацию на сайте <https://openweathermap.org/api>
- Вывести в текущем и недельном прогнозе скорость ветра и видимость.