import pyowm

import telebot

from pyowm.utils.config import get\_default\_config

from pyowm.commons.exceptions import NotFoundError

**Импортируем необходимые библиотеки**:

* pyowm - библиотека для работы с OpenWeatherMap API.
* telebot - библиотека для создания Telegram-ботов.
* get\_default\_config - функция из pyowm, используемая для получения конфигурации по умолчанию.
* NotFoundError - исключение, которое будет выбрасываться, если указанный город не найден.

# Установите API ключи

owm = pyowm.OWM('a68c3f780bb8d0c4f861a73f10e70ef0')

bot = telebot.TeleBot("7014379328:AAF9vcLnmnMw1pNk4OSsmnzl00QbGQedE6M", parse\_mode=None)

**Создаём экземпляры для работы с API**:

* owm - экземпляр для работы с OpenWeatherMap API, используя указанный ключ API.
* bot - экземпляр Telegram-бота, используя указанный токен.

# Укажите язык (опционально, здесь указано на русском)

config\_dict = get\_default\_config()

config\_dict['language'] = 'ru'

**Устанавливаем язык для API**:

* config\_dict - получаем конфигурацию по умолчанию и устанавливаем язык на русский ('ru').

# Получите менеджер для работы с погодой

mgr = owm.weather\_manager()

**Создаём менеджер погоды**:

* mgr - менеджер, который позволяет взаимодействовать с погодными данными.

# Обработчик команды /start

@bot.message\_handler(commands=['start'])

def send\_welcome(message):

bot.send\_message(message.chat.id, "Привет! Я бот, который поможет узнать погоду. ")

bot.send\_message(message.chat.id, "Пожалуйста, напиши название города, чтобы узнать текущую погоду.")

**Создаём обработчик команды /start**:

* Декоратор @bot.message\_handler(commands=['start']) - указывает, что функция send\_welcome будет обрабатывать сообщения с командой /start.
* Функция send\_welcome отправляет два сообщения приветствия пользователю, когда тот вводит команду /start.

# Обработчик текстовых сообщений

@bot.message\_handler(content\_types=['text'])

def send\_echo(message):

try:

observation = mgr.weather\_at\_place(message.text)

w = observation.weather

temp = w.temperature('celsius')['temp']

wind\_speed = w.wind()['speed']

status = w.detailed\_status

clouds = w.clouds

rain = w.rain

snow = w.snow

**Создаём обработчик текстовых сообщений**:

* Декоратор @bot.message\_handler(content\_types=['text']) - указывает, что функция send\_echo будет обрабатывать все текстовые сообщения.
* try - пытаемся получить данные о погоде в указанном пользователем городе.
* observation - получаем наблюдение за погодой в указанном месте.
* w - объект, содержащий данные о текущей погоде.
* Из объекта w извлекаем:
* temp - текущую температуру в градусах Цельсия.
* wind\_speed - скорость ветра.
* status - подробный статус погоды (например, 'ясно').
* clouds - облачность в процентах.
* rain - данные о дожде (если есть).
* snow - данные о снеге (если есть).

answer = f"Температура в {message.text}: {temp}°C\n"

answer += f"Порыв ветра: {wind\_speed} м/с\n"

answer += f"Облачность: {clouds}%\n"

answer += f"Статус: {status}\n"

**Формируем ответ**:

* Создаём строку ответа с информацией о температуре, скорости ветра, облачности и статусе погоды.

if temp > 25:

answer += "На улице жарко, надевайте легкую одежду и не забудьте солнцезащитные очки."

elif 15 <= temp <= 25:

answer += "На улице тепло, подойдет легкая одежда, возможно, кофточка или легкая куртка."

elif 5 <= temp < 15:

answer += "На улице прохладно, надевайте куртку и что-нибудь теплое."

else:

answer += "На улице холодно, обязательно надевайте теплую одежду, шапку и перчатки."

**Добавляем рекомендации по одежде**:

* В зависимости от температуры, добавляем к ответу советы по одежде.

if wind\_speed > 10:

answer += " Также учтите, что сильный ветер, может понадобиться ветровка или шарф."

**Добавляем рекомендации по ветру**:

* Если скорость ветра превышает 10 м/с, добавляем совет о необходимости ветровки или шарфа.

if rain:

answer += " Похоже, будет дождь, возьмите зонт или дождевик."

if snow:

answer += " Ожидается снег, оденьтесь потеплее и возьмите зимнюю обувь."

**Добавляем рекомендации по осадкам**:

* Если ожидается дождь, добавляем совет взять зонт.
* Если ожидается снег, добавляем совет одеться потеплее и взять зимнюю обувь.

bot.send\_message(message.chat.id, answer)

except NotFoundError:

bot.send\_message(message.chat.id, "Такого города не существует.")

**Отправляем ответ пользователю**:

* Отправляем сформированный ответ пользователю.
* Если город не найден (NotFoundError), отправляем сообщение о том, что такого города не существует.

bot.polling(non\_stop=True)

**Запускаем бота**:

* bot.polling(non\_stop=True) - запускаем бота в режиме непрерывного опроса новых сообщений.