

## telegram bots

### ملخص:

خدمة بوت تليجرام تعتبر واحدة من أقوى مميزات تطبيق تليجرام التي يمكننا الاستفادة منها، فمن خلال مجموعة نقرات بسيطة سوف نتمكن من القيام بالكثير من الأمور المميزة. والبوت هو عبارة عن أداة مبرمجة آليًا لتنفيذ مجموعة المهام عندما تتلقى مجموعة من الأوامر، مثلًا بوت في تلغرام يقوم بالترجمة سيطلب منا إدخال النص المراد ترجمته واختيار لغة الترجمة لكي يقوم بإنجاز المهمة.

A Telegram Bot هو برنامج يتصرف مثل شريك دردشة عادي مع وظائف إضافية. ينفذ مهام محددة مسبقًا بشكل مستقل ودون مشاركة المستخدم. مصطلح بوت مشتق من مصطلح الروبوت. في هذا البحث سنتعامل مع بوت معرفة حالة الطقس في أي مدينة حول العالم بسهولة وتوقعات الطقس لها في الأيام الثلاثة القادمة، يتم ذلك بالدخول إلى البوت، ثم الضغط على الطقس، ويمكننا تحديد حالة الطقس بدقة من خيار مشاركة الموقع الجغرافي مع البوت من خلال التطبيق.

كلمات مفتاحية: تليجرام, بوت ,حالة الطقس.

## 1. مقدمة

بوتات تليجرام أو "telegram bots" هي عبارة عن حسابات يتم تطويرها عن طريق المبرمجين باستخدام خدمات برمجية توفرها شركة تليجرام نفسها، وقد تم تطوير هذه الروبوتات الحالية بهدف تقديم خدمات بسيطة مثل سعر الدولار للريال أو حتى معرفة حالة الطقس أو التحميل من اليوتيوب وغيرها من الخدمات [1].

التنبؤ بالطقس هو تطبيق / تقنية من شأنها التنبؤ بحالة الغلاف الجوي لموقع ووقت معينين. من المهم تتبع تقرير الطقس للراحة اليومية. هناك بعض تطبيقات الجهات الخارجية المتاحة لتطبيق الهاتف المحمول لتتبع توقعات الطقس. ولكن سيكون من الأفضل إذا تم تغليف هذه التطبيقات في بعض منصات المراسلة. هذه المقالة سوف نركز على هذا الموضوع. الطقس بوت لتطبيق تليجرام سيساعدنا من خلال جعلنا لا نترك تطبيق المراسلة للبحث عن تقرير الطقس. يتم نشر Weather Bot في Telegram ويمكننا الحصول على تفاصيل الطقس الحالية والمتوقعة للموقع. وسوف توفر التفاصيل التالية عن الموقع التي يتم توفيرها.

درجة الحرارة الحالية

الحد الأقصى / دقيقة درجة الحرارة

رطوبة

سرعة الرياح

سيكون روبوت الطقس قادرا على توفير ما يلي: تقرير الطقس الحالي فيما يتعلق بموقع الخريطة وتقرير الطقس الحالي فيما يتعلق بالمدينة وتقرير الطقس المتوقع فيما يتعلق بموقع الخريطة وتوقعات الطقس تقرير فيما يتعلق بالمدينة

## 2. أهمية البحث وأهدافه:

باستخدام البوت، يجتاز المستخدم التفويض مرة واحدة فقط عند إضافة البوت إلى الدردشة. يمكن للعميل استخدام البوت بقدر ما هو ضروري، وعندما لا تكون هناك حاجة إليه، يقوم المستخدم بحظر البوت فقط. هذا كل شيء، سهل! لا يوجد إعادة تعيين كلمة المرور بعد الآن.

لا حاجة لتذكر كلمات المرور أو عمليات تسجيل الدخول المستخدمة. تؤدي إضافة بوت إلى الموقع أو التطبيق إلى زيادة عدد جمهور المستخدمين حيث إنها تجعل التواصل مع العملاء وتزويدهم بالمساعدة أسهل بكثير وسهل الاستخدام.

### 3. أهمية البوت:

البوت في كل مكان. يبدو أننا بالأمس فقط لم نكن نعلم بوجودها. الآن بالكاد يمكننا تخيل حياتنا بدونهم. لقد أصبحت شائعة على نطاق واسع بين العديد من المستخدمين النشطين للمراسلين نظرًا لأن لديهم نطاقات استخدام مختلفة - من المحتوى الترفيهي بما في ذلك الألعاب خطوة بخطوة وجمع نقاط المكافآت في المطاعم، إلى الحفاظ على خطة النظام الغذائي، وتتبع عمليات التسليم، وحتى سداد المدفوعات لخدمات مختلفة[2].

إن البوت تطبيق برمجي يقوم بتشغيل المهام الآلية (البرامج النصية) عبر الإنترنت. وبعبارة أخرى، إنه برنامج مكتوب لتقليد السلوك البشري من أجل أداء بعض المهام. تأخذ البوتات المدخلات وتقوم بمهام بسيطة ومتكررة، وهي أسرع بكثير من البشر.

هناك نوعان من الروبوتات:

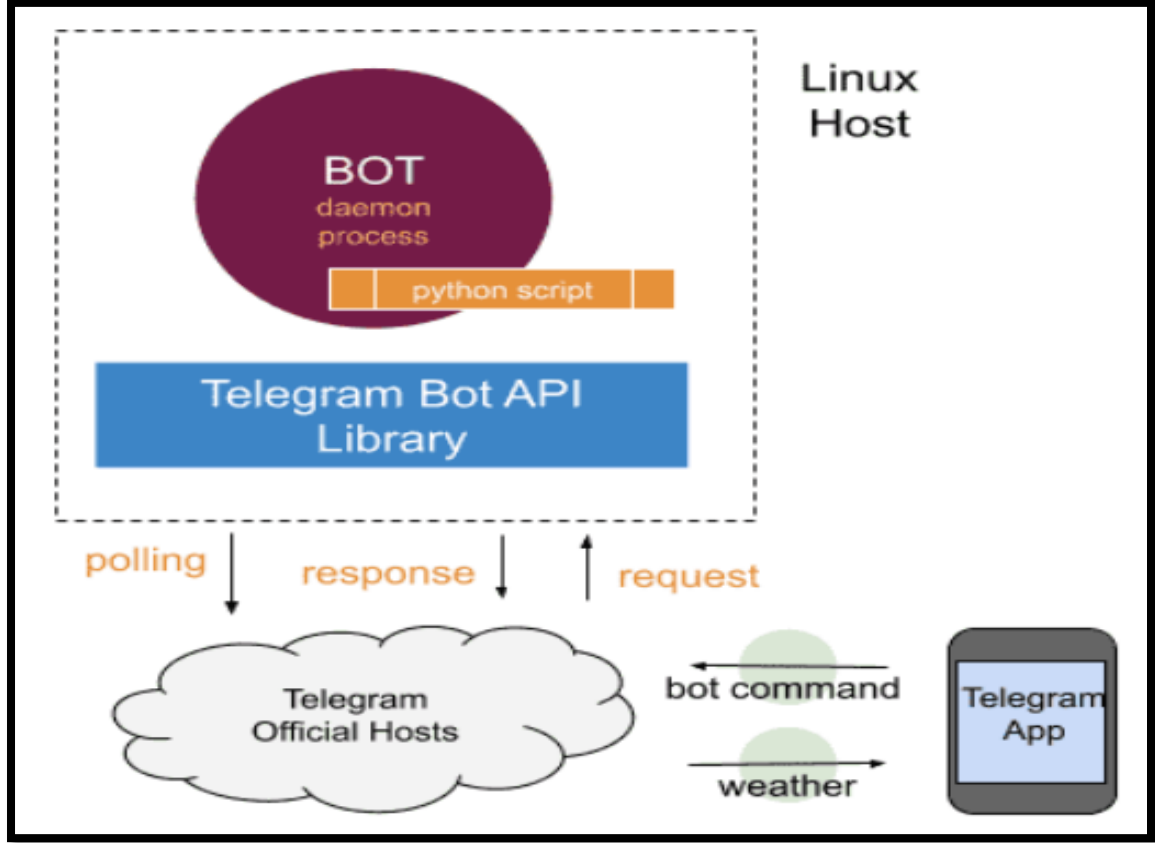
البوتات الجيدة: البوتات المفيدة للمؤسسات وكذلك الأفراد مثل ChatBots وبوتات الوسائط الاجتماعية وما إلى ذلك.

البوتات السيئة: هذه هي الروبوتات التي تستخدم لأداء أنشطة ضارة مثل البريد المزعج.

### 1.3. العلاقة بين بوت ومضيف تليجرام وتطبيق الهاتف المحمول:

نوضح العلاقة حول الكيانات كما في الشكل (1). يتم تنفيذ Bot بواسطة البرامج النصية Python مع مكتبة واجهة برمجة تطبيقات Telegram bot ، ويعمل كعميل خفي في مضيفات Linux. يواصل البوت استطلاع آراء المضيفين الرسميين في Telegram لطلبات الخدمة.

التلجرام تطبيق وبرنامج مصنف ضمن وسائل التواصل الاجتماعي لكنه مميز عنها، نظرا أنه يمكن أن يصبح وسيلة تعويضك وتجيب على كل الأسئلة وتنفذ كل الأوامر في غيابك، وهذا ما يقوم به نيابة عنك "البوت" أو المجيب الآلي أو الروبوت أو المجيب الذكي كما يحلو للبعض تسميته، لكن رغم اختلاف الأسماء فإن له نفس المهمة التي يبرمج عليها.



الشكل (1)

عندما تأتي الطلبات، تتعامل نصوص Python في البوت معها وترسل رسالة طقس مناسبة من خلال Telegram .api

أما واجهات برمجية التطبيقات [3] API ، وهو وسيط برمجي يسمح لتطبيقات بالتواصل مع بعضهما البعض. وبعبارة أخرى، يمكن أن تمنحك هذه البرامج إمكانية الوصول إلى العديد من التطبيقات مثل Telegram / Facebook / Twitter ولكن أيضا إلى قاعدة بيانات كبيرة أيضا. تعمل واجهات برمجية التطبيقات مع بنية دقيقة لتكون قابلة للتطوير والفهم بسهولة. وعادة ما يستخدمون بروتوكولات وأساليب HTTP أي GET ، POST.

أفضل طريقة لتحقيق ذلك هي البدء في اللعب باستخدام واجهة برمجة التطبيقات التي طورها Telegram؟ تعد واجهة برمجة تطبيقات Telegram بداية سهلة لتلقي إشعارات رائعة دون الحاجة إلى أطراف ثالثة.

## 2.3. البوت أهم من التطبيق:

هناك عدة أسباب [4]:

### 1. تصميم بسيط:

لا يمكن أن يكون للبوت تصميم ملون. ولكن، مقارنة بالعديد من التطبيقات ذات التصميمات المختلفة عندما يتعين عليك أن تتذكر أين وماذا تنقر، فإن الروبوتات أكثر سهولة؛ ويقدم اتصالاً بسيطاً عبر النصوص.

### 2. يحتوي البوت على حد أدنى من الإعلانات ويركز على احتياجات المستخدمين.

لا يتعين على المستخدم تثبيت مئات التطبيقات لكل خدمة إذا كان بإمكانك تلقي كل المساعدة اللازمة من الروبوت. هذا مفيد بشكل خاص للمطاعم والمتاجر. نادراً ما يتوقع العملاء إلى تثبيت التطبيقات من مجموعة من الأماكن التي زاروها. نتيجة لذلك، يفقد أصحاب الأعمال ملاحظات العملاء ويفقدون التواصل معهم. إذا كان لدى كل من هذه الأماكن بوت خاص به متاح في برامج مراسلة مختلفة، فسيكون ذلك أكثر ملاءمة وودية للمستخدمين. لا أحد يحب أن يملأ مساحة تخزين هواتفه بالتطبيقات غير الضرورية التي سيتم استخدامها مرة أو مرتين. ومع ذلك، يحتاج العملاء إلى التواصل مع أصحاب الخدمة وسوف يقدر على القيام بذلك من خلال برنامج المراسلة المفضل لديهم.

### 3. لا حاجة للتسجيل والتفويض وإعادة تسجيل الدخول المستمر:

باستخدام البوت، يجتاز المستخدم التفويض مرة واحدة فقط عند إضافة البوت إلى الدردشة. يمكن للعميل استخدام البوت بقدر ما هو ضروري، وعندما لا تكون هناك حاجة إليه، يقوم المستخدم بحظر البوت فقط. هذا كل شيء، سهل! لا يوجد إعادة تعيين كلمة المرور بعد الآن.

لا حاجة لتذكر كلمات المرور أو عمليات تسجيل الدخول المستخدمة. تؤدي إضافة بوت إلى الموقع أو التطبيق إلى زيادة عدد جمهور المستخدمين حيث إنها تجعل التواصل مع العملاء وتزويدهم بالمساعدة أسهل بكثير وسهل الاستخدام.

بذلك، حددنا المزايا الرئيسية للروبوتات وسنلقي نظرة على المشكلات التي يجب معالجتها في مرحلة ما قبل التطوير.

#### 4. متطلبات Telegram Bot:

1. حساب Telegram: إذا لم يكن لدينا تطبيق Telegram مثبتًا، فيمكننا تنزيله من متجر Google Play

أو Apple Store. بمجرد التنزيل، يتعين علينا إنشاء حساب باستخدام رقم الهاتف المحمول، تمامًا مثل

WhatsApp.

2. وحدة بايثون برقية بوت: لهذا المشروع، سنحتاج إلى وحدة تعرف باسم بايثون برقية بوت. تقدم هذه المكتبة

واجهة Python نقية لواجهة برمجة تطبيقات Telegram Bot. وهو متوافق مع إصدار بايثون

3.6.8+. بالإضافة إلى تنفيذ واجهة برمجة التطبيقات النقية، تتميز هذه المكتبة بالعديد من الفئات عالية

المستوى لجعل تطوير برامج الروبوت سهلاً وبسيطاً .

#### 5. التوصيات:

يعد روبوت Telegram إضافة مفيدة للاتصال الذكي، خاصة في المجالات المهنية. بالمقارنة مع خدمات المراسلة

الأكثر شعبية، يقدم Telegram مزايا إضافية في تبادل المعلومات مع الاستفسارات والإجراءات الخاصة للروبوت

المدرج في مقالتنا. ومع ذلك، مثل جميع البرامج، يتطلب Telegram عادة معرفة برمجة ملموسة للاستخدام المكثف.

خاصة للاستخدام في الصناعة واتصال الماكينة، يعد OPC Router أداة موصى بها تبسط هذه البرمجة المعقدة

بدقة.

#### 6. التطبيق العملي:

سنقوم ببناء بوت تلغرام يدعى Weather Bot يقوم بتقديم خدمتين هما الطقس الأسبوعي وطقس اليوم. كل

خدمة من الخدمات السابقة تتطلب إدخال المدينة المراد الحصول على الطقس فيها.

عندها يقوم البوت بإرسال معلومات الطقس المطلوب في المدينة المحددة.

## خطوات العمل:

### 1- إنشاء بوت تلغرام:

يتم ذلك عن طريق بوت موجود مسبقاً في تلغرام يدعى ب Bot Father.



يتم التخاطب مع البوت لإنشاء بوت خاص بنا وتحديد اسمه وصورته الشخصية. بعدها يرسل إلينا البوت رمز Token يتم استخدامه ضمن تطبيقنا للوصول للبوت الخاص بنا وإدارته.

### 2- الحصول على API يقدم خدمة أحوال الطقس:

تمت العملية من خلال موقع rapid api والذي يقدم خدمة open weather map تم تسجيل الاشتراك به والحصول على API Key يتم استخدامه للحصول على الخدمة لمدة 5 أيام فقط.

### 3- كتابة كود بايثون لإدارة البوت:

نقوم بكتابة كود يستخدم لاستقبال الرسائل من المستخدمين عن طريق البوت والاستجابة لها بالشكل المناسب.

### 4- تشغيل كود البايثون واختبار البوت:

يمكن تشغيل الكود أيضاً على استضافة وشراء خدمة الطقس لفترة أطول من 5 أيام ليعمل البوت بشكل مستمر.

## كود البوت:

### constants.py

```
import telegram

BOT_API_KEY = '5328...8' # https://t.me/python_weather_bot

BASE_URL = "https://community-open-weather-map.p.rapidapi.com"
RAPID_API_KEY = "8435e3b835mshcf8322310756be3ple3038jsn5b8c35f02c10"
RAPID_API_HOST = "community-open-weather-map.p.rapidapi.com"

KB_MAIN = telegram.ReplyKeyboardMarkup([
    [telegram.KeyboardButton('طقس اليوم')],
    [telegram.KeyboardButton('الطقس الاسبوعي')],
])

GOVERNORATES_AR = ["دمشق", "حلب", "اللاذقية", "حمص", "حماة", "طرطوس"]
GOVERNORATES_EN = ["Damascus", "Aleppo", "Latakia", "Homs", "Hama", "Tartous"]

KB_GOVERNORATES = telegram.ReplyKeyboardMarkup([ [telegram.KeyboardButton(g)] for g in
GOVERNORATES_AR ])

DAYS = ['الاثنين', 'الثلاثاء', 'الاربعاء', 'الخميس', 'الجمعة', 'السبت', 'الأحد']
```

### main.py

```
from datetime import datetime
import requests
from constants import *
from telegram.ext import *

clients_weather_type = {}

def start_command(update, context):
    chat_id = update.message.chat_id
    print('start from chat id: {}'.format(chat_id))
    bot_msg = "مرحبا، أدخل إحدى الخيارات التالية"
    context.bot.send_message(chat_id=chat_id,
                             text=bot_msg,
                             reply_markup=KB_MAIN)

def message_handler(update, context):
    user_msg = update.message.text
    chat_id = update.message.chat_id
    print(user_msg)

    weather_type = ""
    if user_msg == "طقس اليوم":
        weather_type = "current"
    if user_msg == "الطقس الاسبوعي":
        weather_type = "weekly"
```



```

if weather_type!="":
    clients_weather_type[chat_id] = weather_type
    bot_msg = "يرجى اختيار المحافظة"
    context.bot.send_message(chat_id=chat_id,
                             text=bot_msg,
                             reply_markup=KB_GOVERNORATES)

else:
    if user_msg in GOVERNORATES_AR:
        gov_ar = user_msg
        index = GOVERNORATES_AR.index(gov_ar)
        gov_en = GOVERNORATES_EN[index]

        # get weather type
        weather_type = clients_weather_type[chat_id] if chat_id in
clients_weather_type else "current"

        # get the weather and send the message
        if weather_type == "current":
            bot_msg = get_current_weather(gov_en)
        else:
            bot_msg = get_weekly_weather(gov_en)
    else:
        bot_msg = "اختيار خاطئ، أدخل إحدى الخيارات التالية"

    context.bot.send_message(chat_id=chat_id,
                             text=bot_msg,
                             reply_markup=KB_MAIN)

def get_current_weather(city):
    url = BASE_URL + "/weather?units=metric&q={}".format(city)
    res = requests.get(url, headers={'X-RapidAPI-Key': RAPID_API_KEY, 'X-RapidAPI-
Host': RAPID_API_HOST})
    status = res.status_code
    if (status == 200):
        res = res.json()
        msg = "{} {} \nClouds: {}% \nHumidity: {}% \nWind: {}m/s \nPressure:
{}hpa".format(
            res['weather'][0]['description'],
            res['main']['temp'],
            res['clouds']['all'],
            res['main']['humidity'],
            res['wind']['speed'],
            res['main']['pressure'],
        )
        return msg
    return "عذراً حصل خطأ تقني"

def get_weekly_weather(city):
    url = BASE_URL + "/climate/month?units=metric&q={}".format(city)
    res = requests.get(url, headers={'X-RapidAPI-Key': RAPID_API_KEY, 'X-RapidAPI-
Host': RAPID_API_HOST})
    status = res.status_code

```

```

if (status == 200):
    res = res.json()
    weather_days = res['list']
    msg = ""
    for i in range(7):
        weather = weather_days[i]
        dt = datetime.fromtimestamp(weather['dt'])
        day = DAYS[dt.weekday()]
        average_min = weather['temp']['average_min']
        average_max = weather['temp']['average_max']
        humidity = weather['humidity']
        wind_speed = weather['wind_speed']

        msg = msg + "{} {}° - {}°\nHumidity: {}%\nWind: {}m/s\n\n".format(day,
average_min, average_max, humidity, wind_speed)
    return msg
    return "عذراً حصل خطأ تقني"

def main():
    updater = Updater(BOT_API_KEY, use_context=True)
    dp = updater.dispatcher
    dp.add_handler(CommandHandler("start", start_command))
    dp.add_handler(MessageHandler(Filters.text,message_handler))
    updater.start_polling()
    updater.idle()

print('Bot Started...')
main()

# pip install python-telegram-bot
# pip install requests

```

## اختبار البوت:





## 7. المراجع:

- [1] <https://www.javatpoint.com/building-a-telegram-bot-using-python>
- [2] <https://djangostars.com/blog/how-to-create-and-deploy-a-telegram-bot/>
- [3] <https://telegrambots.github.io/book/>
- [4] <https://www.opc-router.com/what-is-a-telegram-bot/>