
כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

מאת OGRose



הקדמה

Whatsapp היא תוכנה להעברת מסרים מידיים שמשתמשת באינטרנט לטלפונים סלולאריים. ניתן לשלוח באמצעותה הודעות טקסט, תמונות, סרטונים, מיקום, אודיו, לקיים שיחות טלפון מעל האינטרנט והקמת קבוצות בין חברים שבה כל הודעה שנשלחת, נשלחת אל כל חברי הקבוצה.

היא האפליקציה השנייה במספר ההורדות ב-Android (לא כולל אפליקציות של גוגל שמגיעות כברירת מחדל עם מערכת ההפעלה). עם בסיס משתמשים עצום של 900 מיליון משתמשים פעילים, וגרסה שקיימת לכמעט כל סוג של מכשיר סלולארי, ניתן להניח שקורא כתבה זאת משתמש ב-Whatsapp.

Whatsapp הוקמה בשנת 2009, ומאז נמצאת בעלייה מתמדת במספר משתמשים. החברה נרכשה על ידי Facebook ב-22 מיליארד דולר בפברואר 2014.

במאמר זה אסביר כיצד האפליקציה עובדת, את פרוטוקול התקשורת שלה ואיך ניתן לכתוב, בעזרת סקריפט פייתון, בוט שמתפקד כמשתמש באפליקציה.



על מנת שתוכלו לממש את הבוט ולהבין את תוכן המאמר תצטרכו:

- ידע בפייתון
- סביבת עבודה מתאימה (מומלצת מערכת לינוקס)
 - לדעת איך להשתמש ב-Bash
- הבנה בסיסית ברשתות
- הבנה בסיסית ב-XML
- חיבור לאינטרנט

בנוסף לזאת, תצטרכו מספר טלפון שתוכלו לקבל אליו הודעת SMS אחת, אך על נושא זה ארחיב בהמשך.

ארכיטקטורת Whatsapp

Whatsapp, ללא כל הפיצ'רים המיוחדים, היא בסופו של דבר, תוכנה למסרים מידיים. ניתן להקביל אותה ל-ICQ, MSN, AOL Instant Messenger וכד'. כל התוכנות האלה עובדות מעל פרוטוקול התקשורת - XMPP.

XMPP (Extensible Messaging and Presence Protocol) הוא פרוטוקול שמשמש להעברת מסרים מידיים על גבי XML, כלומר כל ההודעות נשלחות ומתקבלות בפורמט XML-לי. הפרוטוקול עובד מעל TCP/IP.

על מנת שנוכל להבין כיצד הפרוטוקול לשליחת וקבלת מסרים מידיים עובד אנחנו צריכים ראשית להכיר כמה מושגי יסוד:

- Stream - הכרזה על חיבור פתוח שנוצר ונשלח לפני התחלת התקשורת עצמה בין שני הצדדים, והוא זה שמעיד על התחלת וסיום ההתקשרות. ה-Stream כולל בתוכו מידע נוסף לגבי התקשורת (כתובת השרת, גרסת הפרוטוקול וכד'). על מנת לפתוח חיבור עלינו להשתמש בתגית ה-XML:

```
<stream:stream  
...>
```

על מנת לסיים את ההתקשרות עלינו להשתמש בתגית:

```
</stream:stream>
```



- Stanza - אלמנט XML שלם שהשרת או הלקוח שולח על Stream פתוח. קיימים סוגים רבים של Stanza-ות אך בבסיס הפרוטוקול קיימים שלושה:

- `<presence/>` - מספק הודעה על מצב הנוכחות של המשתמש (זמין, עסוק וכד')
- `<message/>` - מאפשר שליחת הודעה מישות אחת לאחרת
- `<iq/>` - בעזרת תקשורת עם השרת מקבל וקובע מידע על כל אלמנט בשיחה ומסביב לה

כל סוג Stanza אחר שעובר בפרוטוקול עובר מבוסס על אחת משלושת סוגי ה-Stanza-ות האלו. על כל Stream יכולות להישלח Stanza-ות ללא הגבלה בצורה דו-כיוונית.

על מנת שנוכל לראות דוגמה מהעולם האמיתי על יצירת חיבור ותקשורת XMPP בסיסית, אנחנו לא נשתמש בשרתי Whatsapp מכיוון שאלו מאפשרים שליחה וקבלת מידע בצורה מוצפנת בלבד. במידה ונרצה להתנסות עם הנושא באמצעות חיבור ללא תוכנה כדי לראות את תעבורת השרת אנחנו נשתמש בשרתי Google Talk (אפליקציית מסרים מיידים שגם מבוססת על פרוטוקול XMPP).

על אף סגירת השירות על ידי גוגל לפני כמה חודשים, השרתים נשארו פתוחים על מנת שאפליקציות צד שלישי יוכלו להמשיך לתקשר ביניהן. כדי לפתוח חיבור עם השרת אנחנו נשתמש בתוכנה Putty ונתחבר אל שרתי גוגל ב-`talk.google.com` באמצעות פורט 5222 - הפורט הלא מוצפן של הפרוטוקול, וכך נוכל לראות את ההודעות שלנו עוברות ב-Plaintext.

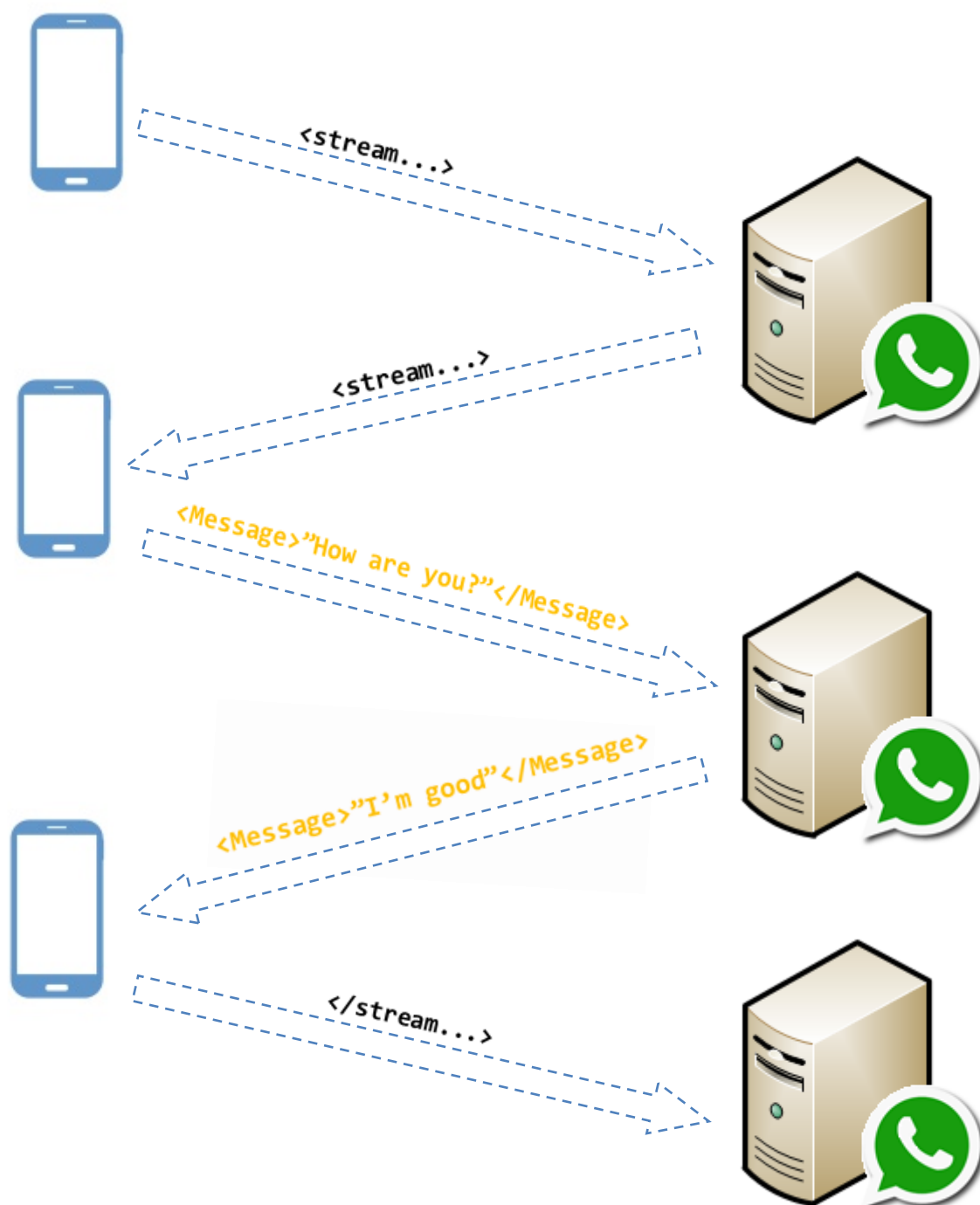
מצורפת דוגמה להסנפה של מידע שנשלח והתקבל מול השרת.

10.0.0.7	64.233.166.125	XMPP/XML	75 XML
10.0.0.7	64.233.166.125	XMPP/XML	72 STREAM > og5440@gmail.com
64.233.166.125	10.0.0.7	XMPP/XML	199 STREAM < og5440@gmail.com
64.233.166.125	10.0.0.7	XMPP/XML	295 FEATURES
10.0.0.7	64.233.166.125	XMPP/XML	105 STARTTLS
64.233.166.125	10.0.0.7	XMPP/XML	104 PROCEED

ניתן לראות שתחילה אנחנו יוזמים את החיבור דרך הכתובת הלוקאלית (10.0.0.7) לשרתי Google talk. לאחר מכן אנחנו מקבלים בקשה לאוטנטיקציה מול השרת, ולאחר שאנחנו מספקים אותה, בקשה לאחת נוספת. אנחנו שולחים אותה ומאותו רגע אנו יכולים להתחיל לשלוח ולקבל Stanza-ות מהשרת. (יש לציין שאין באמת אפשרות לעשות זאת מכיוון שגוגל דורשים שלב אוטנטיקציה נוסף, אך על מנת להבין איך ההתחברות עובדת זה מספיק).

ניתן לסכם בפשטנות את שלב ההתחברות מול השרת בעזרת התרשים הבא:

- Client •
- Server •
- Stanza •

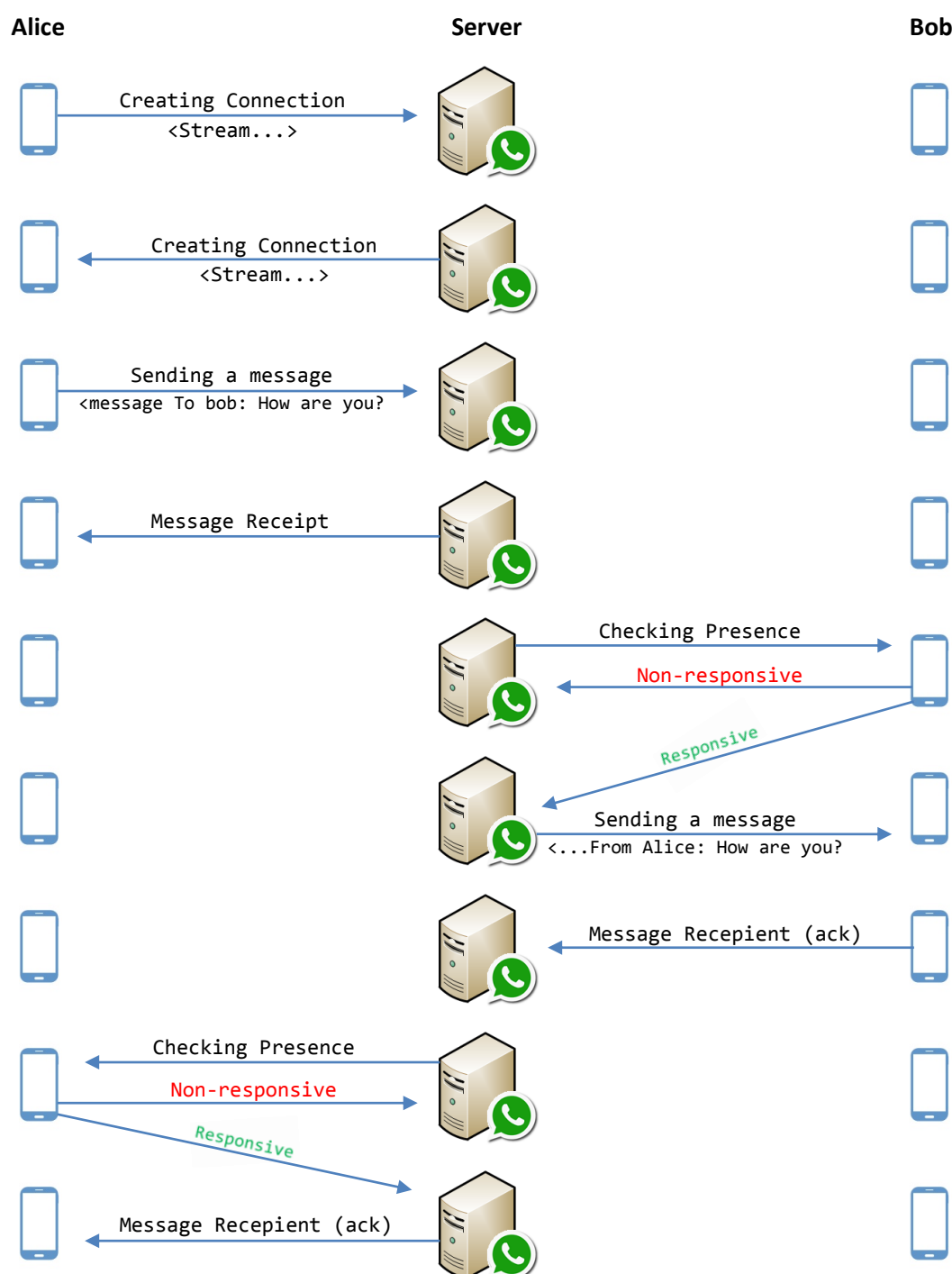


כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il

ב-Whatsapp מנגנון ההתחברות ושליחת ההודעות הוא מורכב יותר מכיוון שהוא עובד מעל Customized XMPP, במימוש זה צומצמו מספר הבתים של ה-Metadata על מנת לצמצם את ה-Overhead של הפרוטוקול וכל התקשורת בו מוצפנת.

אנסה לסכם את התהליך באמצעות תרשים מחזור-החיים של הודעה העוברת ב-Whatsapp:



כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



כמה הערות לגבי התרשים:

- כל פעם שנבדקת הנוכחות היא תיבדק עד שהמקבל של ההודעה יגיב לבדיקת הנוכחות, וכאשר הוא יגיב ההודעה תישלח אליו.
- ההודעה תישמר למשך 30 יום על שרתי Whatsapp עד שתימחק.
- כאשר הודעה נשלחת היא נמחקת מהשרת.

מהתרשים הושמטו הרבה שלבים במחזור החיים של ההודעה (לדוגמה, פתיחת Stream מול Bob כאשר הוא מגיב, סגירת ה-stream, שימוש מול קבוצה ועוד...). יש לציין שכל ישות שנשלחת היא Stanza בפני עצמה: ההודעות, ה-Receipt, בדיקת הנוכחות וה-Ack.

עכשיו שאנחנו מבינים: מהי התוכנה בבסיסה, מעל איזה פרוטוקול היא עובדת, איך הפרוטוקול הזה בנוי, איך אנחנו מתחברים לשרתים ומהו מחזור החיים של ההודעה, אנחנו יכולים להתחיל להשתמש בידע הזה על מנת לכתוב לה בוט.

כתיבת בוט ל-Whatsapp

על מנת לכתוב בוט אנחנו הולכים להשתמש בספריית Yowsup, ספרייה ל-Python שהכרנו כבר קודם בתהליך ההרשמה כשהשתמשנו ב-CLI (Command line interface) שלה. הספרייה תאפשר לנו לבצע כל פעילות שאנחנו מבצעים באפליקציה הרגילה דרך סקריפט.

על מנת להשתמש בספרייה עלינו להשתמש במכונה שיש בה Python מותקן בגרסה 2.6+ או 3.0+ לצורך ההדגמה אני אריץ Lubuntu (הפצת לינוקס) גרסה 15.04 על vmware workstation. ניתן גם להוריד ולהשתמש בגרסאות Windows של הספרייה.

על מנת להתקין את החבילה קודם נתקין את ה-Dependencies שלה:

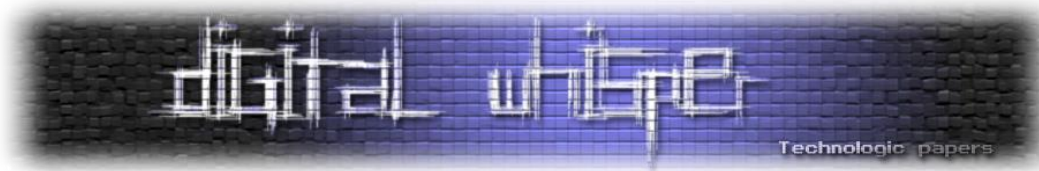
```
sudo apt-get git
git clone https://github.com/tgalal/yowsup
sudo apt-get install python-dateutil
sudo apt-get install python-setuptools
sudo apt-get install python-dev
sudo apt-get install libevent-dev
sudo apt-get install ncurses-dev
```

לאחר הרצת פקודות אלה אנחנו מוכנים להתקין את החבילה עצמה, עלינו להריץ את סקריפט ההתקנה של הספרייה (נמצא בתיקייה שעשינו אליה git clone):

```
sudo python setup.py install
```

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



הערה: קיימת בעיה בהתקנת הספרייה, כאשר אנחנו מריצים את סקריפט ההתקנה (setup.py) הוא נתקע כאשר הוא מנסה להוריד ולהתקין את הספרייה pillow, כדי לתקן זאת הייתי צריך להוריד גרסה ישנה יותר של הספרייה באמצעות pip. יש להריץ את השורות הבאות במידה ואתם נתקעים בהתקנה:

```
sudo apt-get install python-pip  
sudo pip install pillow==2.7.0
```

הרשמה ל-Whatsapp

בחלק זה אתאר כיצד ניתן להירשם ל-Whatsapp (גם באמצעות מכשיר ישן שאינו תומך באפליקציה). מומלץ מאוד להשתמש בשלב זה בפיתוח הבוט, כתיבת בוטים נוגדת את ה-TOS של Whatsapp, ובמידה ומספר יסומן כבוט, החברה תתן Ban למספר, אני ממליץ להשתמש במדריך על מנת להימנע ממצב כזה.

תהליך ההרשמה ל-Whatsapp מורכב משלושה שלבים:

- שליחת הודעת ביקוש הרשמה ל-Whatsapp
- קבלת הודעת אוטנטיקציה
- שליחת הודעה שמראה על האוטנטיקציה

השלב הראשון אמור לקרות בטלפון הסלולרי שלנו. מכיוון שאיננו רוצים לקשר את המספר האמיתי שלנו לבוט אנחנו נצטרך להשיג מספר אחר, וזאת על מנת לקבל SMS אחד שיאפשר לנו להשתמש בשירות.

על מנת להשיג מספר סלולרי ישנן כמה אפשרויות:

- להשתמש במספר ישן/מספר גיבוי של טלפון שעדיין יש לנו גישה אליו.
- להשתמש במספר שמקושר ל-Sim שיש לנו גישה אליו (מספר מכרטיס טוקמן שנקנה בחו"ל לדוגמה)
- קניית מספר חדש
- לאחר בירורים, התוכנית הזולה ביותר שהצלחתי למצוא היא לקנות ב-Cofix סים בחמישה שקלים ולהירשם לתוכנית ה-Light של Walla mobile למשך חודש.

כל שיטה שבה אתם מצליחים להשיג מספר שעוד לא רשום ל-Whatsapp ויכול לקבל SMS תעבוד טוב. לאחר שיש לנו את המכשיר המדובר, ניתן להתחיל את ההרשמה עצמה, אותה נבצע באמצעות ספריית Yowsup.

נתחיל בביצוע הפקודה:

```
Yowsup-cli registration --requestcode sms --phone 972XXXXXX -cc 972 --mcc XXX -mnc XXX
```

על מנת להשיג את ה-MCC וה-MNC הרלוונטיים נשתמש בטבלה הבאה:

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



https://en.wikipedia.org/wiki/Mobile_country_code

(לדוגמא שנרצה לקבל את ה-SMS לחברת סלקום אנחנו נשתמש ב-MNC 2, וב-MCC 425).

השורה הזאת תשלח אל המספר שנתנו את הודעת האונטיקציה עם הקוד שעלינו להזין על מנת לסיים את תהליך ההרשמה. במידה וקיבלנו את הקוד 123456 אנחנו נשתמש בספרייה שנית על מנת לסיים את תהליך ההרשמה עם השורה הבאה:

```
Yowsup-cli registration -register 123456 -phone 972XXXXXXX -cc 972
```

יש ציין שבמידה ואנחנו לא שולחים את ההודעה ממספר ישראלי ולא מצפים לקבל תשובה למספר ישראלי אנחנו נשנה את ה-CC על פי הקודים שנמצאים ברשימה הזאת: <https://countrycode.org/>

לאחר ששלחנו את הפקודה הזאת אנחנו נקבל את ההודעה הבאה ב-console שממנו ביצענו את הפקודה:

```
status: ok
kind: free
pw: [REDACTED]
price: 3.90 ₪
price_expiration: [REDACTED]
currency: ILS
cost: 3.90
expiration: [REDACTED]
login: [REDACTED]
type: new
```

תהליך ההרשמה נגמר ואנחנו קיבלנו את המידע שרצינו. מההודעה הבאה עלינו לשמור את השדה: pw. על מנת להתחבר אנחנו נשתמש בשדות login עם המספר שהזנו, ובסיסמא שקיבלנו באמצעות תהליך ההרשמה הזה.

שליחת הודעה ראשונה

לאחר סיום שלבים אלה הספרייה אמורה להיות מותקנת ועלינו ליצור קובץ קונפיגורציה על מנת שנוכל להתחבר לשרתי Whatsapp עם המשתמש שנוצר קודם. יש ליצור קובץ ריק ולהכניס אליו את הפרטים הבאים:

- cc=COUNTRYCODE (972: ישראל)
- phone=PHONENUMBER
- password=XXXXXXXXXXXXXXXXXXXXXXX

לאחר מכן נשמור את הקובץ בשם שנבחר (לדוגמא CONFIG_FILE).

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



על מנת שנוכל לבדוק שההתקנה עבדה כמו שצריך ושהמשתמש רשום כראוי, נריץ כעת שליחת הודעה פשוטה לטלפון שנבחר באמצעות שימוש ב-CLI המובנה בספרייה. כדי להיכנס אל ה-CLI נריץ:

```
Yowsup-cli demos --yowsup --config CONFIG_FILE
```

במידה והספרייה הותקנה כראוי אנחנו נכנסו עכשיו אל ה-CLI ונוכל באמצעותו לשלוח הודעות ולקנפג את המשתמש שלנו. על מנת להשתמש בממשק עלינו להתחבר תחילה. כדי להתחבר נזין את הפקודה:

```
Yowsup Cli client
=====
Type /help for available commands

[offline]:/L
Auth: Logged in!
[connected]:
```

[בהתחברות הראשונה, ההתחברות תיקח כמה שניות ולא תהיה מיידי]

לאחר מכן אנחנו נרצה לשלוח הודעה תוך שימוש בממשק, ועל מנת לעשות זאת נשתמש בפקודה:

```
/message send *SEND_TO_NUMBER* "TestMessage"
```

```
[connected]:
[connected]:/message send [redacted] "TestMessage"
[connected]:
Sent: [redacted]
```

במידה וההודעה הגיעה אל המספר שהזנתם - הצלחתם להתקין את החבילה, מזל טוב!, ומה עכשיו? כל מה שנשאר הוא לכתוב את הרובוט שתוצו לממש.

הבנת ארכיטקטורת Yowsup

Yowsup מורכב מ-Stack של מספר שכבות (layer), שניתנות להחלפה. כל שכבה היא ערוץ דו-כיווני (ערוץ שניתן לשלוח ממנו, ולקבל אליו מידע). כל שכבה תתקשר עם השכבה שמתחתיה ומעליה, ולכל שכבה יש תפקיד ספציפי בשינוי המידע שעובר דרכה לפורמט שמתאים לשכבה הבאה. השכבה העליונה ביותר היא הראשונה בשליחת המידע ואחרונה בקבלתו בעוד שהשכבה התחתונה ביותר היא הראשונה בקבלת המידע והאחרונה בשליחתו.

על כל שכבה לממש לפחות שתי פונקציות: send ו-receive. הפונקציה send שולחת data כלשהו לשכבה שמתחתיה, והפונקציה receive שולחת מידע כלשהו לשכבה שמעליה.

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

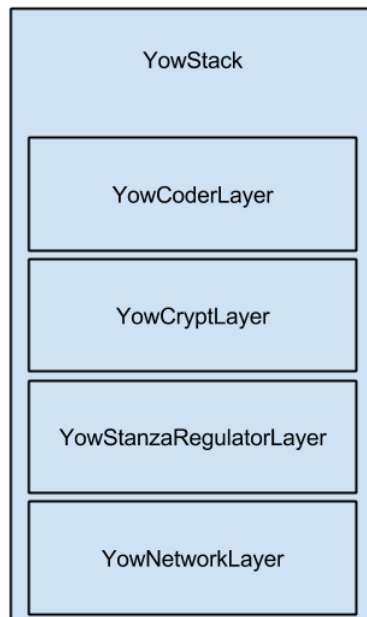
www.DigitalWhisper.co.il

לדוגמה שכבה שמעבירה את המידע דרכה:

```
class PassthroughLayer(YowLayer):
    def send(self, data):
        # Sends the data to the layer below
        self.toLower(data)

    def receive(self, data):
        # Sends the data to the layer above
        self.toUpper(data)
```

ל-Yowsup ישנן כמה core layers שדרושות לפעולת הספרייה והן מתוארות בתרשים הבא:



- **YowNetworkLayer** - אחראית על קריאת מידע משרת ה-Whatsapp, ושליחת המידע אליו חזרה.
- **YowStanzaRegulatorLayer** - השכבה הזאת מקבלת מידע מהשכבה מתחתיה עד שיש לה מספיק בתים כדי ליצור Stanza. היא מעבירה לשכבה שמעליה מערך בתים שמייצג בדיוק Stanza אחת. המידע שהיא מקבלת מהשכבה שמעליה נשלח לשכבה מתחתיה ללא שינוי.
- **YowCryptLayer** - השכבה הזאת מפענת את המידע שהיא מקבלת מהשכבה מתחתיה ושולחת אותה לשכבה מעליה. כשהיא שולחת מידע למטה היא מקבלת מידע, מצפינה אותו, ושולחת.
- **YowCoderLayer** - מקבלת מערך בתים של מידע מפוענח ויוצרת ממנו אובייקט בשם ProtocolTreeNode, אותו היא מעבירה למעלה. היא גם מצפה לקבל אובייקט כזה חזרה שאותו היא יכולה להפוך חזרה למערך בתים.

אנחנו יכולים לבחור להרכיב את ה-Stack בכל צורה שנבחר, אך עלינו לשמור על מודל ה-Stack שמוצג על ידי Yowsup, ולהשתמש ב-Core layers.

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



מעל ה-Core layers האלו קיימות שכבות נוספות שהספרייה מספקת:

- **YowProtocolLayer** - שכבה זאת מצפה לקבל ProtocolTreeNode ולהעלות לשכבה שמעליה ProtocolEntity.
- **ProtocolEntity** - הוא אובייקט שמאפשר גישה פייתונית ישירה לתכונות של ההודעה בלי התעסקות מיותרת ב-XML - לדוגמה: ניתן לגשת לתוכן ההודעה באמצעות הפונקציה `getBody()`.
- **YowInterfaceLayer** - שכבה שמצפה לקבל אובייקט מסוג ProtocolEntity מהשכבה מתחתיה. שכבה זאת מוסיפה פונקציות נוחות נוספות וזאת השכבה שבדרך כלל יש לרשת ממנה כאשר ממשים בוט משלנו.

בנוסף לקונספט השכבות עלינו לרדת לעומקו של מושג נוסף:

- **YowLayerEvent** - השכבות גם מסוגלות לשלוח ולקבל Event-ים אחת מהשניה.

```
self.emitEvent(YowLayerEvent) #sends event to upper layers
self.broadcastEvent(YowLayerEvent) # sends event to lower layers
```

[על מנת לנצל את ה-Event-ים האלה עלינו לממש פונקציית OnEvent בשכבות הרצויות]

- **Yowsup - Parallel Layers** גם תומך בשכבות מקבילות. הרעיון הוא להציב כמה שכבות באותה רמה ב-Stack, כל השכבות יקבלו את אותו סוג של אובייקט מלמעלה וכולן יישלחו אותו אובייקט למטה, רק שכל שכבה מטפלת בסוגי Stanzas שונים. לדוגמה ב-Stack הבא:
 - **YowAuthenticatorLayer, YowMessagesProtocolLayer, YowGroupsProtocolLayer, YowReceiptProtocolLayer, YowPresenceProtocolLayer YowCoderLayer YowCryptLayer YowStanzaRegulatorLayer YowNetworklayer**
- כל השכבות המודגשות שייכות לאותה הרמה ב-Stack והן מקבילות אחת לשנייה (לכולן אותו Input ואותו Output).

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



- **ProtocolEntityCallback** - כאשר אנחנו יורשים מ-YowInterfaceLayer אנחנו יכולים להשתמש ב-Decorator הזה. ה-Decorator הזה מקבל String וכאשר השכבה מקבלת מידע, במידה ופונקציית ה-getTag() של האובייקט שעלה ממנה מחזיר את אותו ה-String הפונקציה תקרא אוטומטית עם האובייקט שהגיע אל השכבה. לדוגמה ה-Snippet הבא:

```
@ProtocolEntityCallback("message")
def onMessage(messageProtocolEntity):
    """ do stuff"""
```

על מנת לקבל הסבר מלא על הארכיטקטורה (בהסבר לא נכללו כל הפרטים, רק אלו שהרגשתי שהיו הכרחיים לכתיבת בוט בסיסי):

<https://github.com/tgalal/yowsup/wiki/Architecture>

כתיבת בוט ריק

בשלב זה אנחנו נכתוב בוט ריק - כזה בלי לוגיקה, רק כדי לבנות שלד לתוכנית שלנו. נתחיל בכתיבת השכבה שלנו שתשב מעל כל ה-Stack שנבנה. השכבה הזאת לא תעשה כלום, ולאחר מכן, בבוט אמיתי, נוכל לצוק לתוכה לוגיקה משלנו.

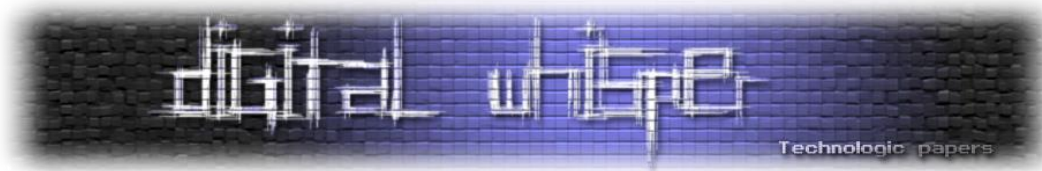
ניתן לראות שהשכבה מקבלת הודעה באמצעות ה-Decorator - ProtocolEntityCallback. השכבה לא עושה כלום עם ההודעה שהיא מקבלת. השכבה הזאת יכולה לשמש כמעין "שלד" לבוטים שמקבלים הודעות ומשיבים אליהן:

```
"""
ExampleLayer - Used as an example. Does nothing.
"""
from yowsup.layers.interface import YowInterfaceLayer, ProtocolEntityCallback

class ExampleLayer(YowInterfaceLayer):
    """
    ExampleLayer- Does nothing
    """
    # For the example we are going to use a program that responds to messages.
    @ProtocolEntityCallback("message")
    def onMessage(self, message):
        """
        This function is called when a message is sent, and the function
        gets the message object as an argument
        """
        """
        INSERT YOUR LOGIC HERE
        """
    pass
```

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



לאחר שבנינו את השכבה הריקה, אנחנו נצטרך לבנות את ה-Stack שיאפשר לנו לכלול את השכבה בהרצת הבוט:

```
#####
Creates the stack and executes the stack main loop.
#####

from ExampleLayer import ExampleLayer
from yowsup.layers.protocol_messages import YowMessagesProtocolLayer
from yowsup.layers.network import YowNetworkLayer
from yowsup.layers.coder import YowCoderLayer
from yowsup.stacks import YowStack
from yowsup.common import YowConstants
from yowsup.layers import YowLayerEvent
from yowsup.stacks import YowStack, YOWSUP_CORE_LAYERS
from yowsup import env

def build_stack():
    stack_layers = (
        ExampleLayer, # The example layer that we just built
        YowMessagesProtocolLayer, # The layer that will handle the messages
    ) + YOWSUP_CORE_LAYERS # The core layers (As explained earlier)

    # Creating the stack
    stack = YowStack(stack_layers)
    # In the following lines we will be replacing the "props" used to create the objects with
    # actual values so that the stack will be connected to the actual whatsapp servers.
    stack.setProp(YowAuthenticationProtocolLayer.PROP_CREDENTIALS,
        ("YOUR_PHONE_NUMBER", "YOUR_PASSWORD"))
    stack.setProp(YowNetworkLayer.PROP_ENDPOINT, YowConstants.ENDPOINTS[0])
    stack.setProp(YowCoderLayer.PROP_DOMAIN, YowConstants.DOMAIN)
    stack.setProp(YowCoderLayer.PROP_RESOURCE, env.CURRENT_ENV.getResource())

    return stack

def main():
    #####
    The programs main function.
    #####

    # Building our stack
    stack = build_stack()
    # Broadcasting an event that we are connected to the whatsapp servers
    stack.broadcastEvent(YowLayerEvent(YowNetworkLayer.EVENT_STATE_CONNECT))
    # Starting the main loop of the sending & receiving data from the whatsapp servers.
    stack.loop()

if __name__ == "__main__":
    main()
```

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



בעזרת שני המודולים האלה אנחנו יכולים להריץ את התוכנה הראשית (stack.py) על מנת לגרום לבוט שלנו לעבוד. שתי דוגמאות הקוד שסופקו פה, אינן עובדות, מכיוון שחסר בהן מידע וקוד. הדוגמאות הללו נועדו להמחשת שלד הקוד בלבד, ואינן מכילות קוד הכרחי להרצה.

בחלק הבא ניתן דוגמה לבוט שעובד, ניתן להעתיק אותו ולהשתמש בו (לאחר שינוי פרטי האותנטיקציה מול Whatsapp).

דוגמה לבוט

בוט שמבצע פקודות Shell פשוטות ומחזיר את התוצאה שלהן:

MessageInformation.py:

```
"""
MessageInformation - filters out the relevant message information we need
                    for MyCommandsLayer.
"""

class MessageInformation(object):
    """
    Generates and contains the relevant message information from a MessageProtocolEntity
    """
    def __init__(self, message):
        self.content = message.getBody()
        self.sender = message.getFrom()
```

myLayer.py:

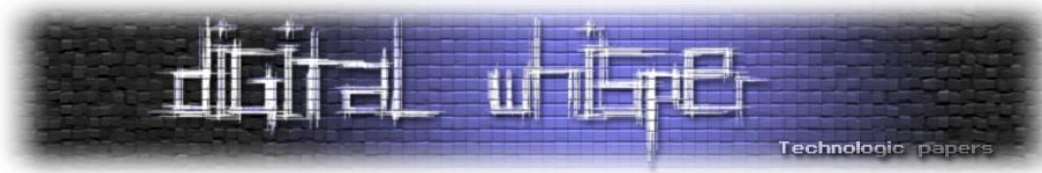
```
"""
MyCommandsLayer - Generates a layer that executes commands,
                  and responds with their output.
"""

import subprocess
from yowsup.layers.interface import YowInterfaceLayer, ProtocolEntityCallback
from yowsup.layers.protocol_messages.protocolentities import TextMessageProtocolEntity
from yowsup.layers.protocol_receipts.protocolentities import OutgoingReceiptProtocolEntity
from yowsup.layers.protocol_acks.protocolentities import OutgoingAckProtocolEntity
from MessageInformation import MessageInformation

# A dictionary containing all the commands we know how to perform.
VALID_COMMANDS = {
    "time" : "date",
    "fortune" : "fortune",
    "identity" : "rig",
    "pi" : "pi"
}
```

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



```
class MyCommandsLayer(YowInterfaceLayer):
    """
    MyCommandsLayer- Executes certain commands and returns their output.
    """

    @ProtocolEntityCallback("message")
    def onMessage(self, messageProtocolEntity):
        """
        Response in the case a text message was received.
        """

        # In this function we will deal with any text messages that we receive.

        # Gnerating the relevant message infomation from the bigger object.
        incomingMessageInformation = MessageInformation(messageProtocolEntity)

        # Creating the message we want to send back.
        outgoingMessage = TextMessageProtocolEntity(
            self.__process_message(incomingMessageInformation),
            to=incomingMessageInformation.sender
        )

        # Creating the receipt to send to the whatsapp servers.
        receipt = self.__get_receipt(messageProtocolEntity)

        # Transferring both of them to the lower layers so that they will be sent.
        self.toLower(receipt)
        self.toLower(outgoingMessage)

    @ProtocolEntityCallback("receipt")
    def onReceipt(self, entity):
        """
        Response in the case a receipt was received.
        """

        # To every message a receipt from the whatsapp server is sent we need to send an ack
        # so that the whatsapp server stops sending us the same message.
        self.toLower(OutgoingAckProtocolEntity(entity.getId(),
            "receipt",
            entity.getType(),
            entity.getFrom()))

    def __get_receipt(self, messageProtocolEntity):
        """
        Generates a receipt from a messageProtocolEntity
        """

        return OutgoingReceiptProtocolEntity(messageProtocolEntity.getId(),
            messageProtocolEntity.getFrom(),
            'read',
```

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



```
messageProtocolEntity.getParticipant())
```

```
def __process_message(self, messageInformation):
```

```
    """
```

Processes the message, executes the command, if it is available.
returns an appropriate output.

```
    """
```

```
    content = messageInformation.content
```

```
    # If it is a command message
```

```
    if content.startswith("!"):
```

```
        # If we know how to execute the command
```

```
        if content[1:] in VALID_COMMANDS:
```

```
            # Execute it and return the output string.
```

```
            command_output = self.__execute_command(VALID_COMMANDS[content[1:]])
```

```
            return command_output[0] # Comamnd output is a tuple, we only want the string.
```

```
        else:
```

```
            # If we don't know how to execute it.
```

```
            return "Please enter a valid command"
```

```
    else:
```

```
        # If the input wasn't a command.
```

```
        return "Please input a command"
```

```
def __execute_command(self, command):
```

```
    """
```

Executes the command and reutrns its output.

```
    """
```

```
    return subprocess.Popen(command, stdout=subprocess.PIPE).communicate()
```

Stack.py:

```
    """
```

Creates the stack and executes the stack main loop.

```
    """
```

```
from MyLayer import MyCommandsLayer
```

```
from yowsup.layers.auth import YowAuthenticationProtocolLayer
```

```
from yowsup.layers.protocol_messages import YowMessagesProtocolLayer
```

```
from yowsup.layers.protocol_receipts import YowReceiptProtocolLayer
```

```
from yowsup.layers.protocol_acks import YowAckProtocolLayer
```

```
from yowsup.layers.protocol_iq import YowIqProtocolLayer
```

```
from yowsup.layers.network import YowNetworkLayer
```

```
from yowsup.layers.axolotl import YowAxolotlLayer
```

```
from yowsup.layers.coder import YowCoderLayer
```

```
from yowsup.stacks import YowStack
```

```
from yowsup.common import YowConstants
```

```
from yowsup.layers import YowLayerEvent
```

```
from yowsup.stacks import YowStack, YOWSUP_CORE_LAYERS
```

```
from yowsup import env
```

כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il



```
CREDENTIALS = ("YOUR_PHONE_NUMBER", "YOUR_PASSWORD")

def build_stack():
    stack_layers = (
        MyCommandsLayer,
        (YowAuthenticationProtocolLayer, YowMessagesProtocolLayer,
        YowReceiptProtocolLayer, YowIqProtocolLayer, YowAckProtocolLayer),
        YowAxolotlLayer
    ) + YOWSUP_CORE_LAYERS

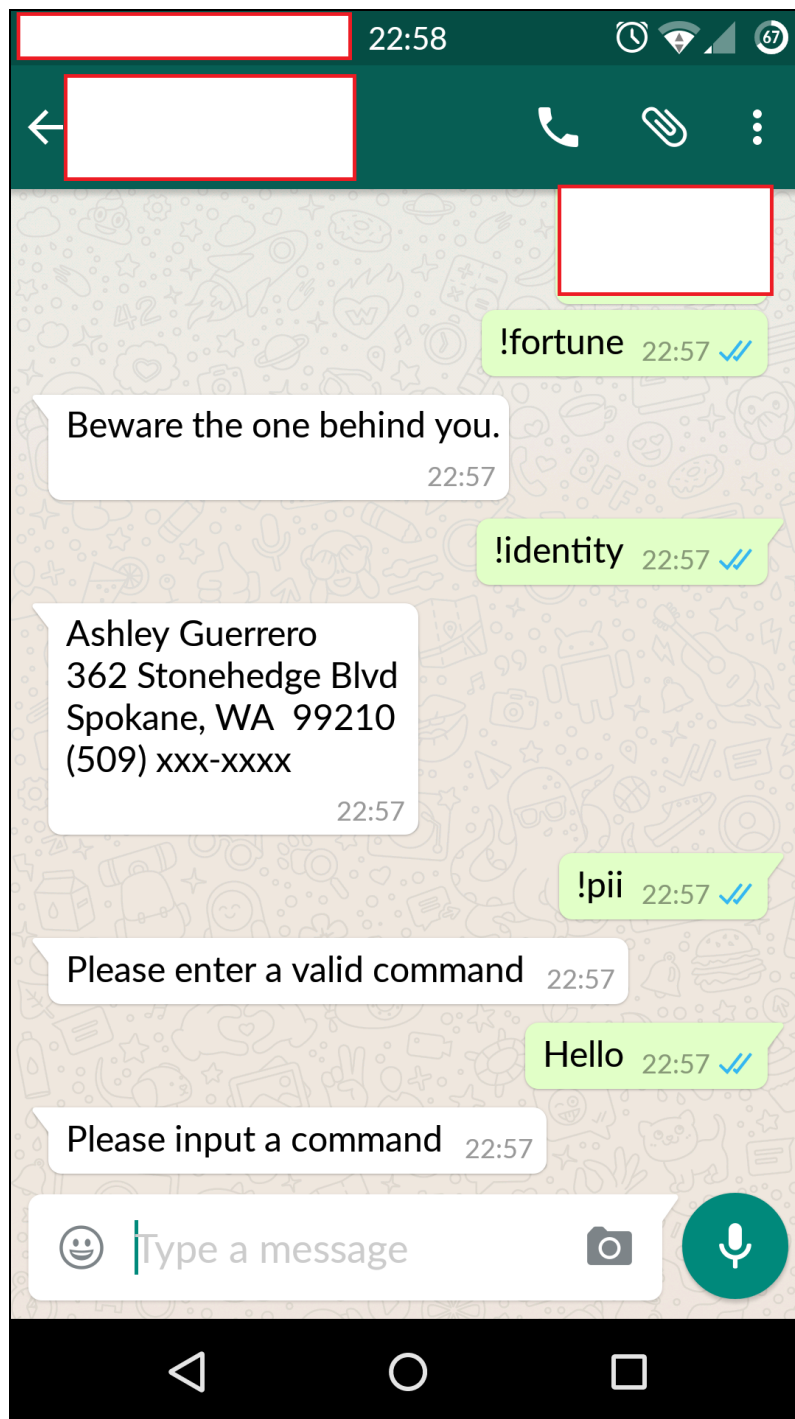
    stack = YowStack(stack_layers)
    stack.setProp(YowAuthenticationProtocolLayer.PROP_CREDENTIALS, CREDENTIALS)
    stack.setProp(YowNetworkLayer.PROP_ENDPOINT, YowConstants.ENDPOINTS[0])
    stack.setProp(YowCoderLayer.PROP_DOMAIN, YowConstants.DOMAIN)
    stack.setProp(YowCoderLayer.PROP_RESOURCE, env.CURRENT_ENV.getResource())

    return stack

def main():
    """
    The programs main function.
    """
    stack = build_stack()
    stack.broadcastEvent(YowLayerEvent(YowNetworkLayer.EVENT_STATE_CONNECT))
    stack.loop()

if __name__ == "__main__":
    main()
```

דוגמאות לשימוש ב-CommandBot:



כל מה שרציתם לדעת על Whatsapp ומעולם לא העזתם לשאול

www.DigitalWhisper.co.il

סיכום

לאחר קריאת המאמר הזה אני מקווה שהקורא מבין יותר טוב איך מערכת מדהימה כזאת עובדת. אנחנו לפעמים מסתכלים על הדברים האלה כבנאליים, אך בסופו של יום, הכוח שהאפליקציה הזאת נותנת לנו הוא עצום. והדרך שבה האפליקציה שינתה את הדרך שבה העולם מסתכל על תקשורת בין אנשים ובין קבוצות היא מדהימה לדעתי.

אני מקווה שנהנתם לקרוא את המאמר, ושאתם רוצים לפתח בוט כזה בעצמכם. הכוח שניתן עם היכולת הזאת הוא עצום. ניתן לכתוב בוט שיעשה כמעט כל דבר: הוא יכול לשמש לכם דרך לשלוח ולקבל חדשות בצורה אוטומטית, הוא יכול לחבר בין שתי קבוצות, הוא יכול לשלוח הודעות לרשימת תפוצה ענקית, הוא יכול להשתמש בכל פיצ'ר שקיים באפליקציה, ולעשות כל מה שעולה בדעתכם, במידה ואתם יכולים לממש זאת בפייטון.

נהנתי מאוד לכתוב את המאמר ואני אשמח, באמת, לקבל כל הערה ושאלה לגבי תוכן המאמר, סגנון הכתיבה, ובקשה להרחבות על נושאים מסויימים. **אבקש במיוחד לפנות אליי במידה ואתם מוצאים שיטה להשיג מספר שאפשר להירשם איתו בחינם.**

ניתן לפנות אליי למייל:

Og5440@gmail.com

