

**הפרויקט שלכם יקיים או ייפול  
על תביסיסי בדיקות  
אוור שרפ', יווץ לSOCTים**

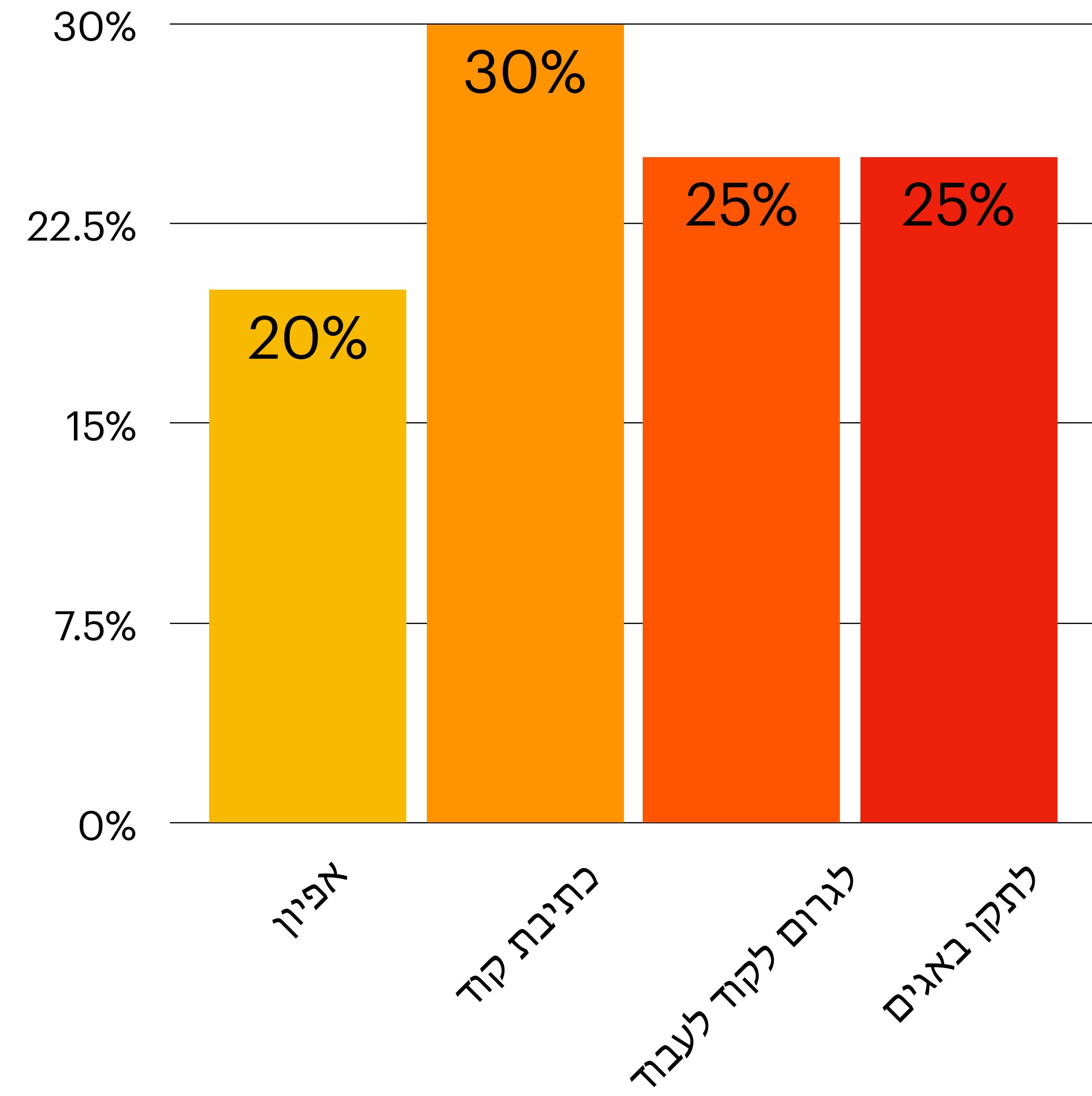
**אוגוסט פינגוון 2023**

**אהובים לתוכנת?**

**אהובים לכתוב בדיקות?**

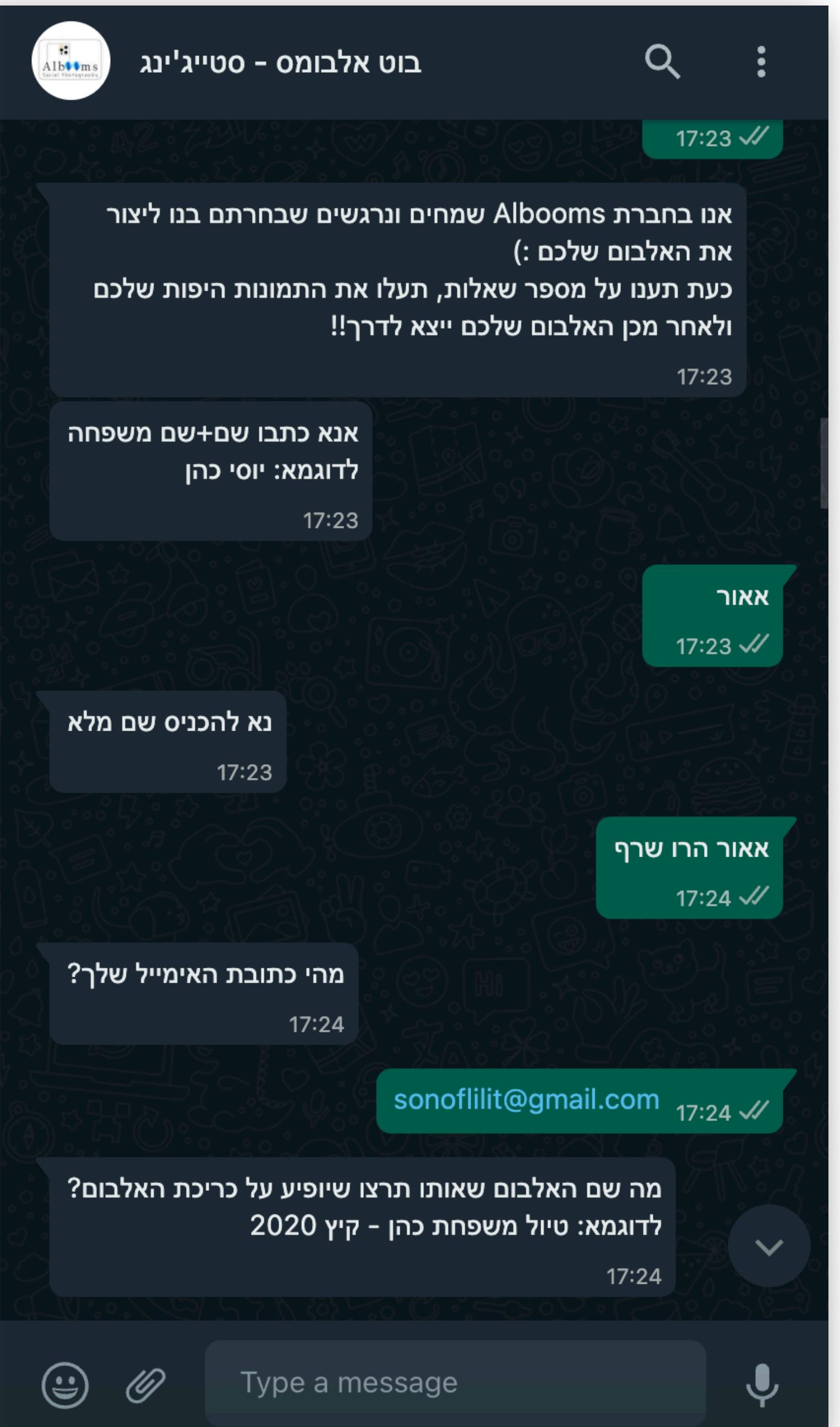
**אהובים לדבג?**

## כמה זמן פיתוח מושקע ב...

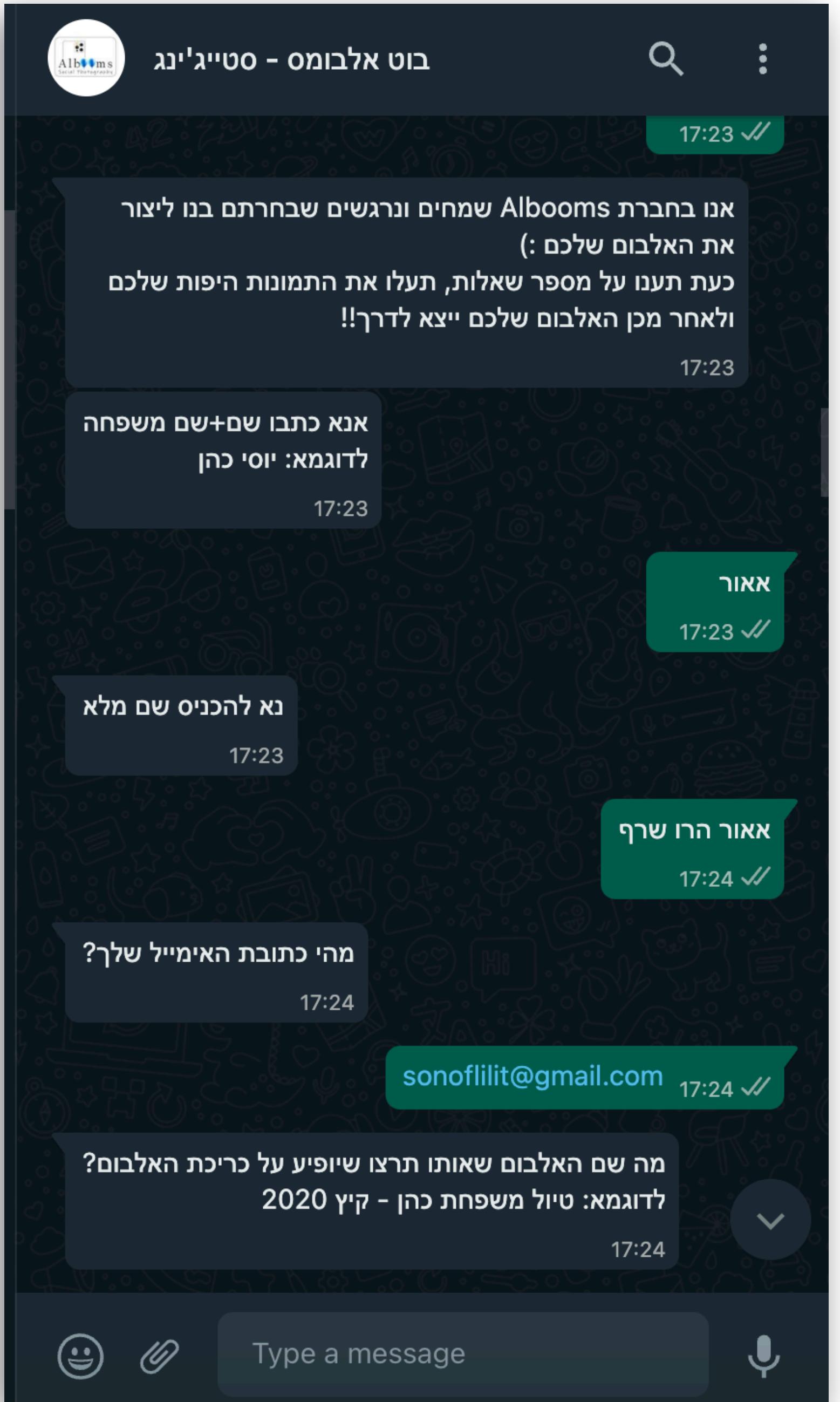
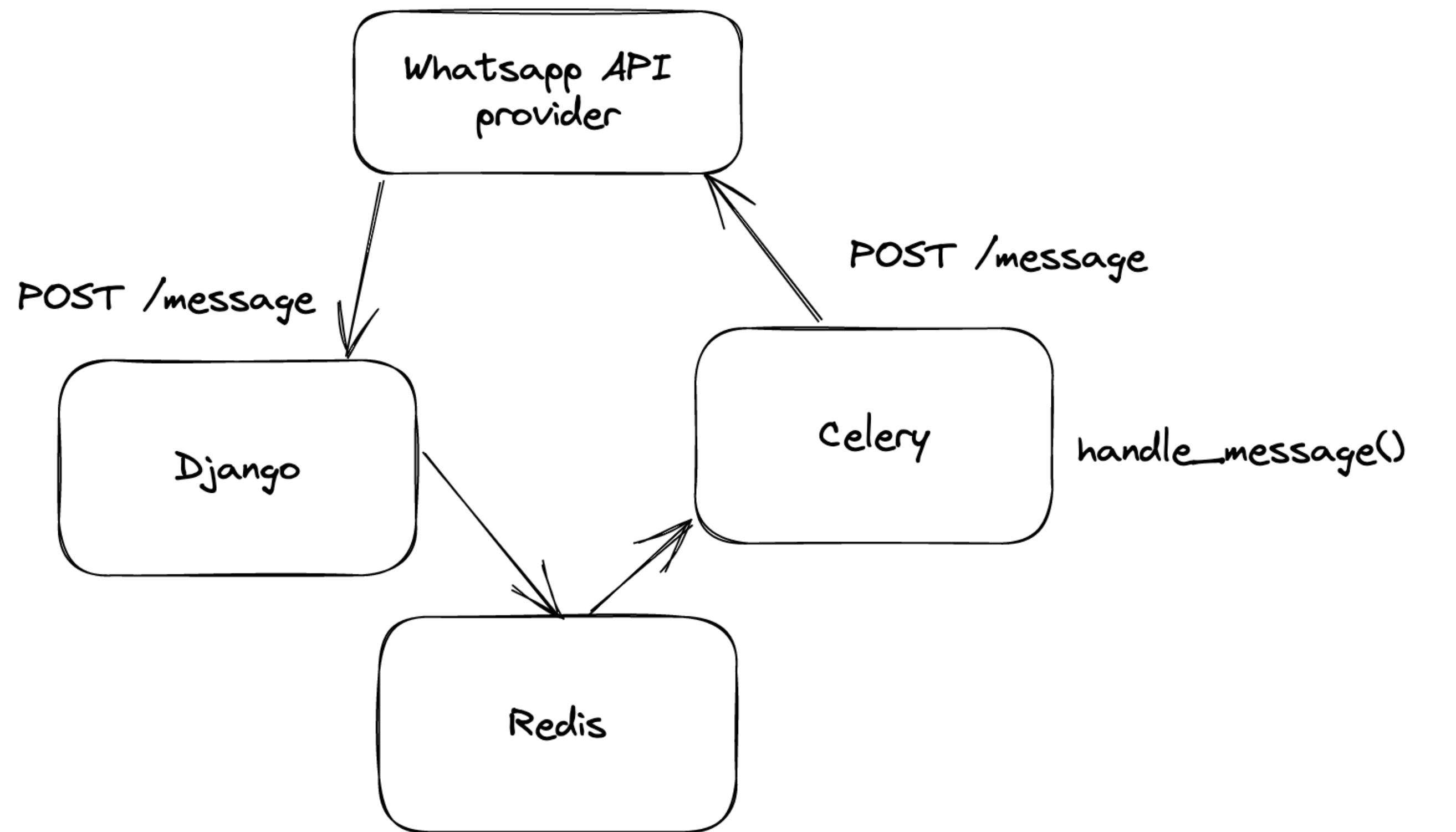


**אפשר יותר טוב**

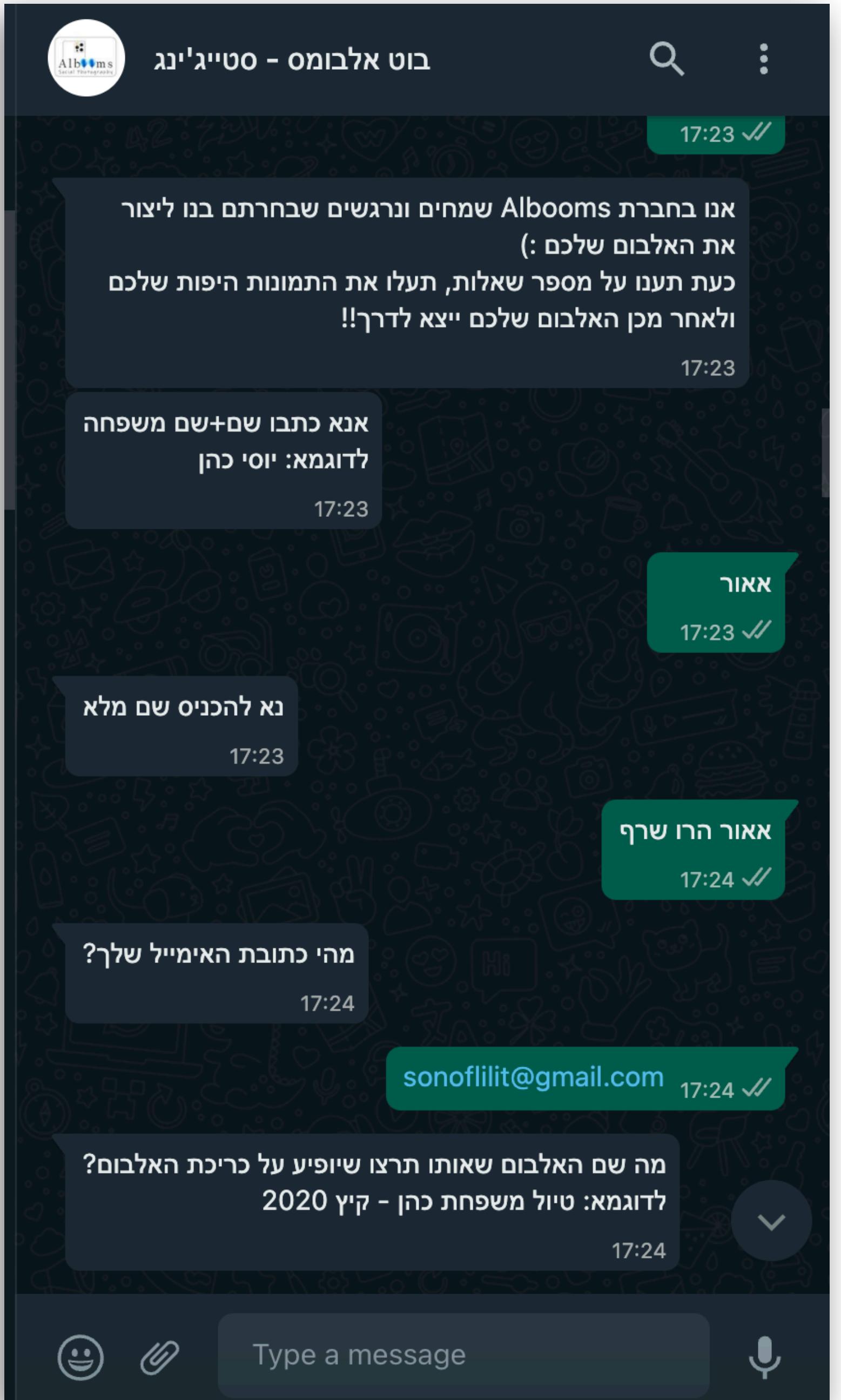
# בוט וווטסאפ שמורך דברים



# **בוט וווטסאפ שماובר דברים**



# בוט ווטסאפ שמורר דברים



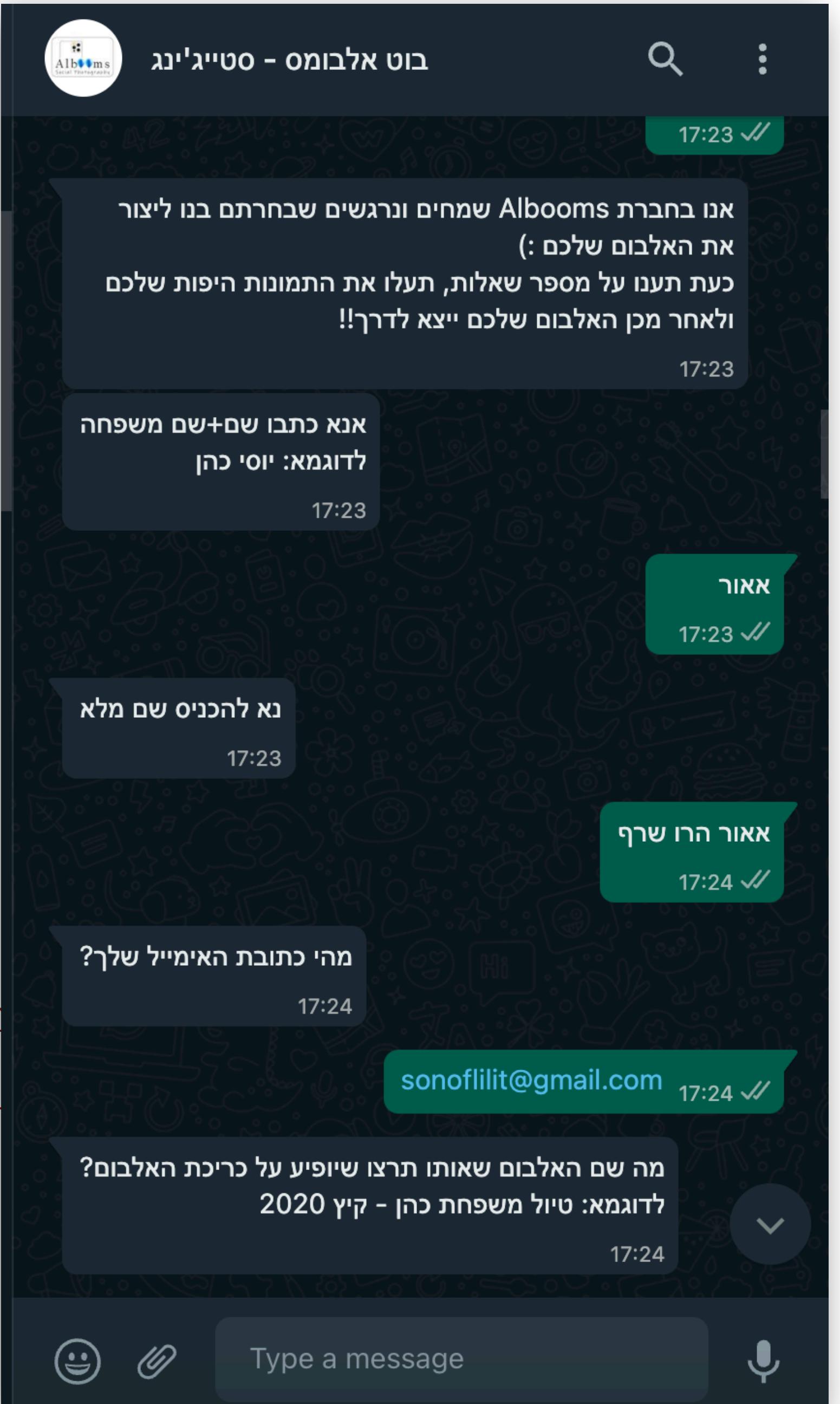
```

def send(text):
    return handle_message({
        "from": NUMBER, "id": "somebase64id", "timestamp": timestamp,
        "text": {"body": text}, "type": "text"})
def receive(send_message_mock, text):
    data = send_message_mock.call_args['data']
    assert json.loads(data)['body']['text'] == text

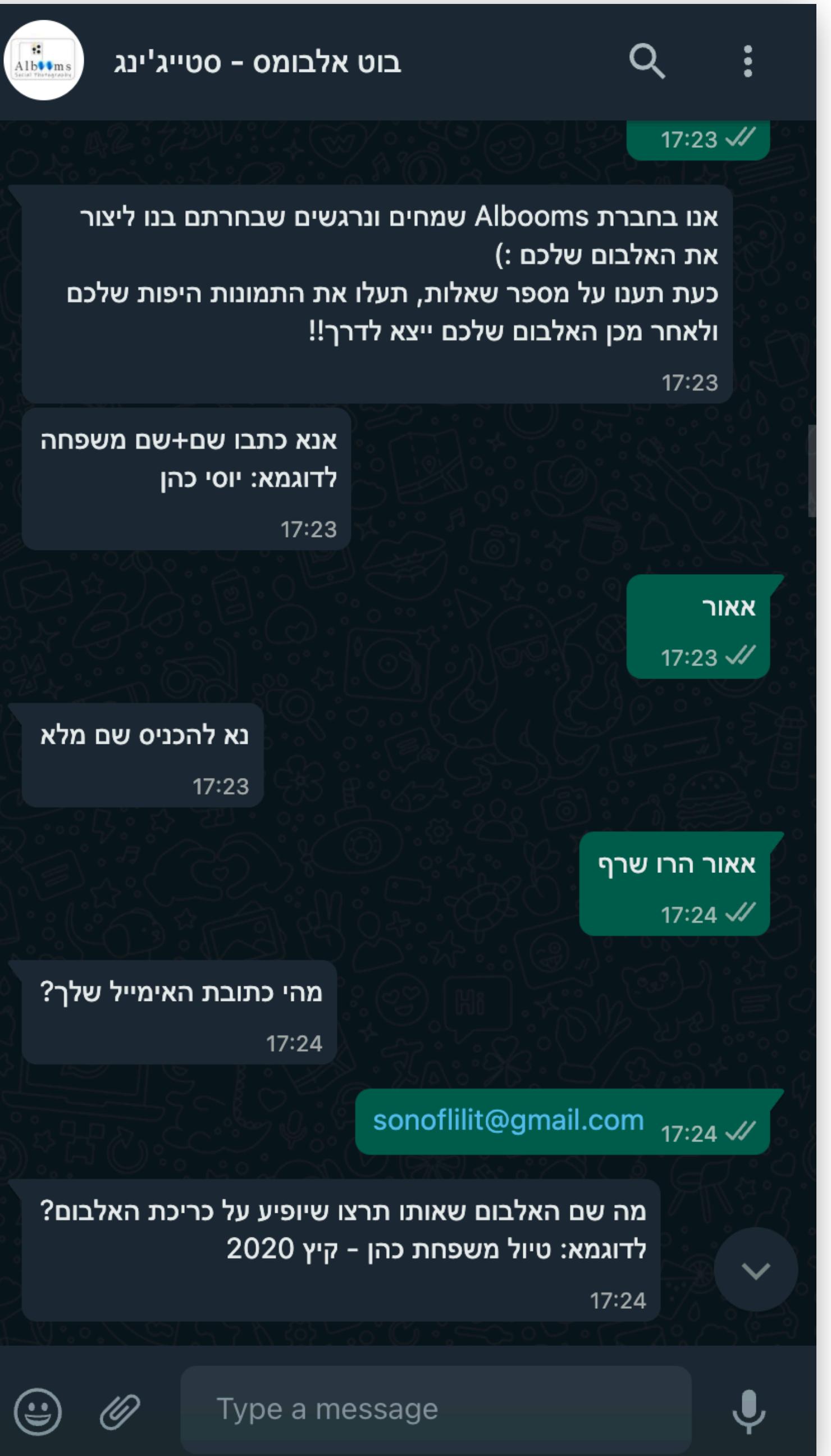
@patch('whatsapp_business_api_is.messages.send_message')
def test_form(send_message_mock):
    user = WaUser.objects.create(number=NUMBER, ...)
    event = Event.objects.create(client=user)

    send("גַם אָנִי רֹצֶה אֱלָבּוּם")
    receive(send_message_mock, "אָנוּ בְּחַבְרָת Albooms בְּנָנוּ")
    receive(send_message_mock, "אָנוּ אֲנָא כְּתָבָן שְׁמַעַשׂ מִשְׁפָּחָה")
    send("אוֹר")
    receive(send_message_mock, "נָא לְהַכְּנֵיס שְׁמַעַשׂ")
    send("אוֹר הַרוֹשָׁף")
    receive(send_message_mock, "?מָה כְּתוּבָה הַמִּיִּיל שְׁלֵזֶק")

```



# אפשר יותר טוב



- להקמת אירוע

- B: - |

חרתם בנו להקים את האירוע הבא שלכם \*Albooms\* אנו בחברת עת תענו על מספר שאלות ולאחר מכן מכוון האירוע שלכם י יצא לדרך

- B: - |

אנא כתבו שם+שם משפחה

לדוגמא: יוסי כהן

- אוור

- B: נא להכניס שם מלא:

- אוור הרו שרפ

- B: מה כתובת המייל שלך?

- sonoflilit@gmail.com

- B: מה גודל האלבום הרצוי?

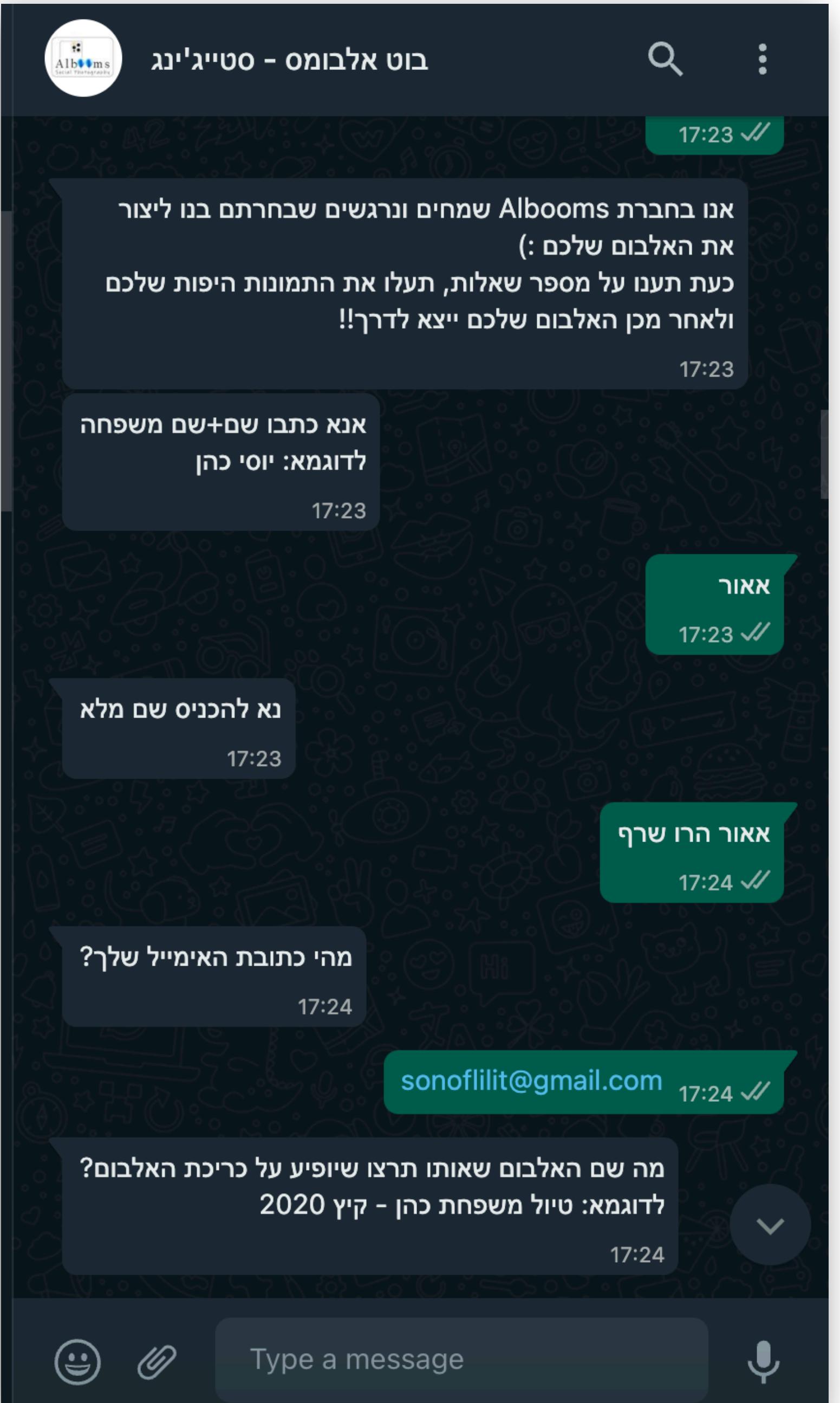
**buttons:**

- אלבום קטן

- אלבום בינוני

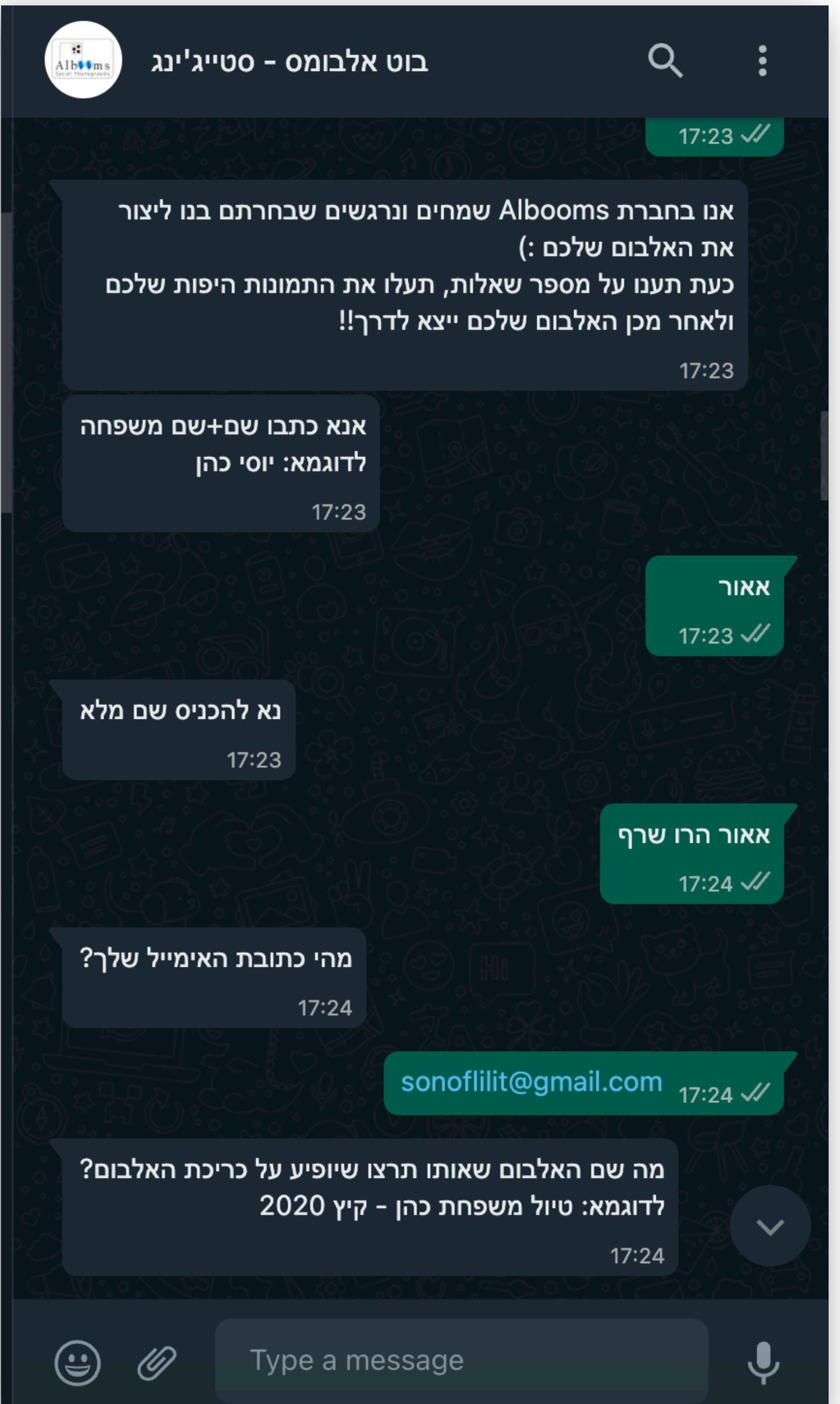
- אלבום גדול

- אלבום בינוני



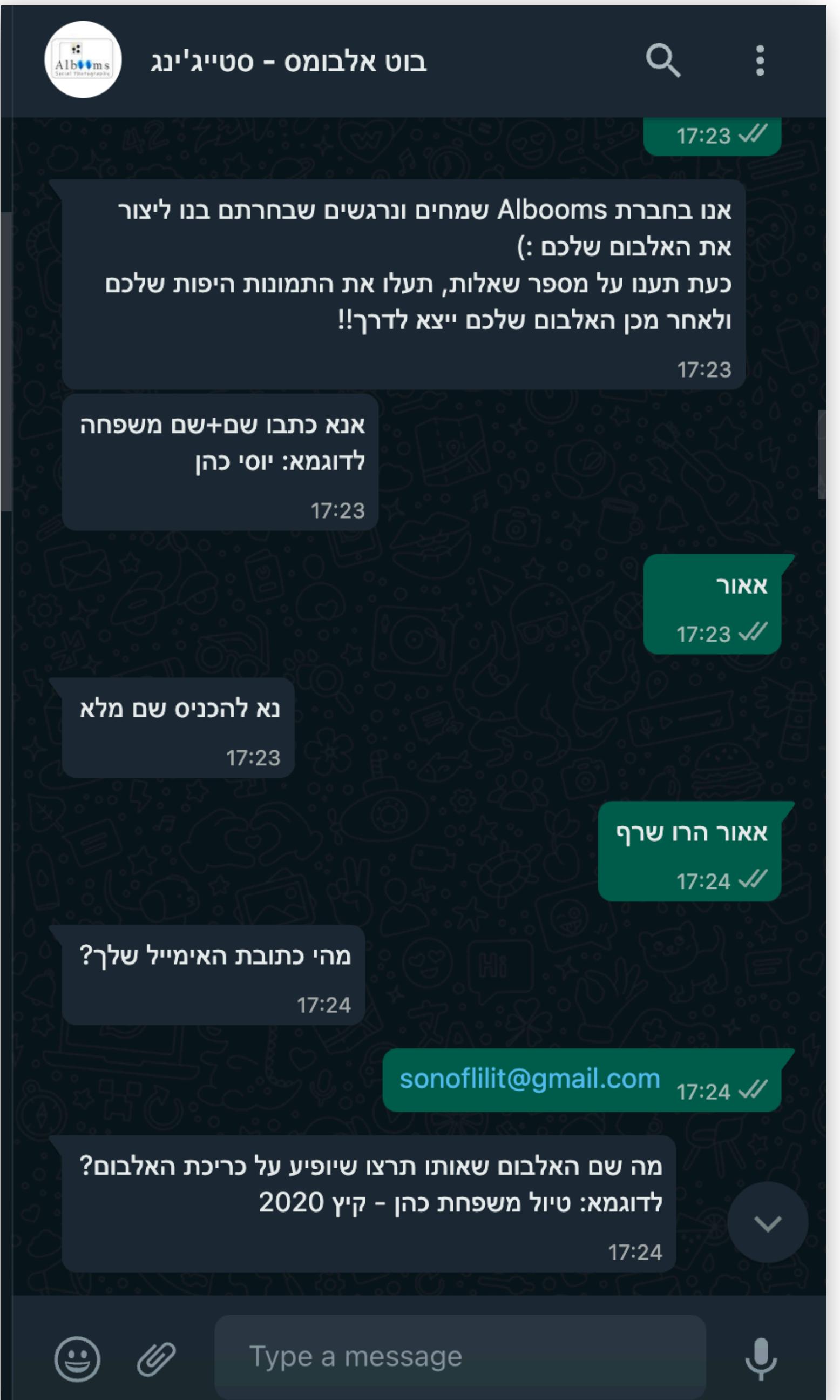
- להקמת אירו<sup>ע</sup>
- B: -
- חרתם בנו להקים את האירו הבא שלכם \*Albooms\* אנו בחברת Albooms שמחים ונרגשים שבחרתם בנו ליצור את האלבום שלכם:
- B: -
- אנא כתבו שם+שם משפחה
- לדוגמא: יוסי כהן
- א/or
- B: נא להכניס שם מלא:
- א/or הרו שרפ
- B: מה כתובת המייל שלך?

## נוצר מitor שיחה



- להקמת אירו<sup>ע</sup>
- B: -
- חרתם בנו להקים את האירו הבא שלכם \*Albooms\* אנו בחברת Albooms שמחים ונרגשים שבחרתם בנו ליצור את האלבום שלכם:
- B: -
- אנא כתבו שם+שם משפחה
- לדוגמא: יוסי כהן
- א/or
- B: נא להכניס שם מלא:
- א/or הרו שף
- B: מה כתובת המייל שלך?

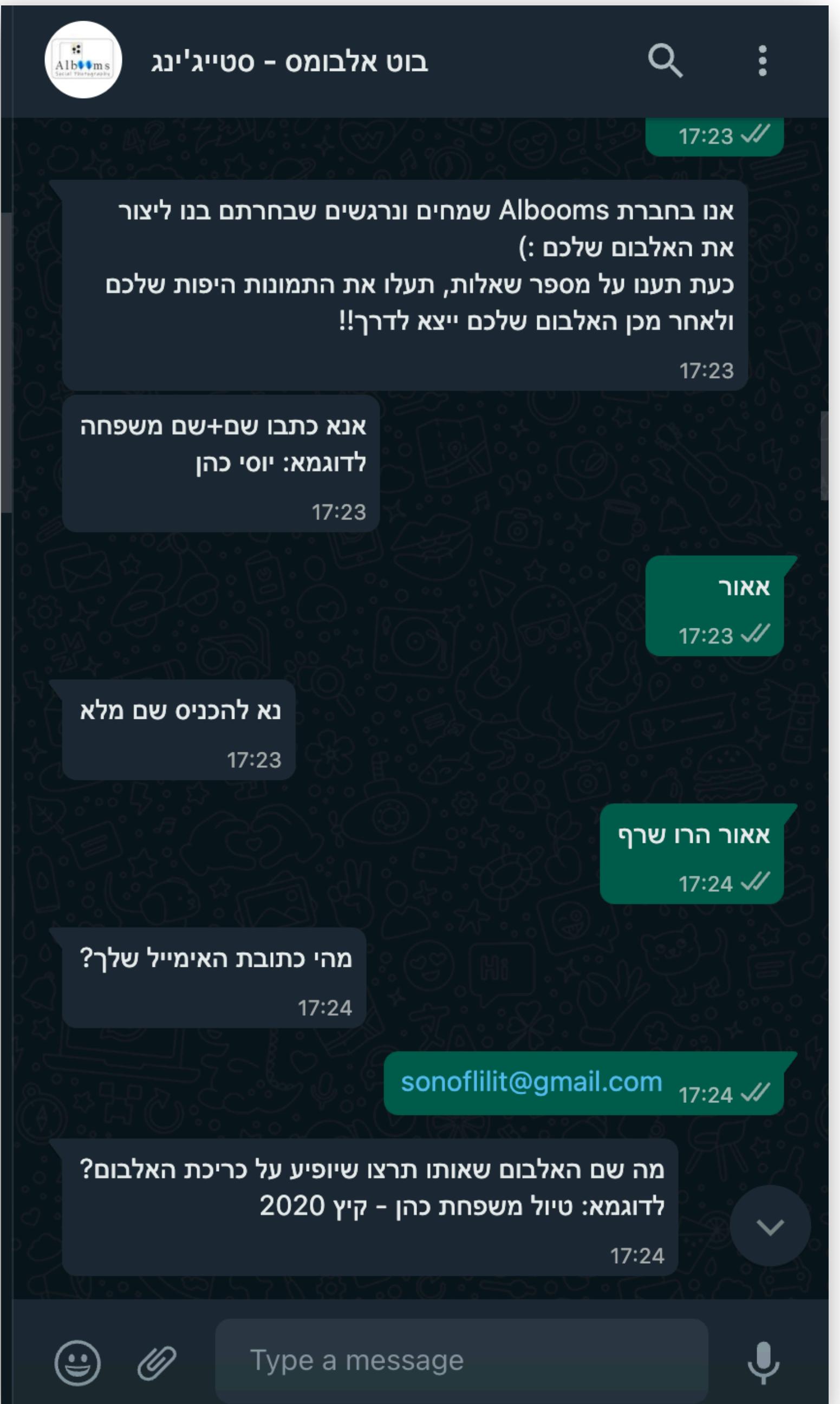
# נוצר מטור שיחה קל לעורך



- להקמת אירוּע
- B: -
- חרתם בנו להקים את האירוּע הבא שלכם \*Albooms\* אנו בחברת Albooms שמחים ונרגשים שבחרתם בנו ליצור את האלבום שלכם:
- B: -
- אנא כתבו שם+שם משפחה
- לדוגמא: יוסי כהן
- א/or
- B: נא להכניס שם מלא:
- א/or הרו שרפ
- B: מה כתובת המייל שלך?

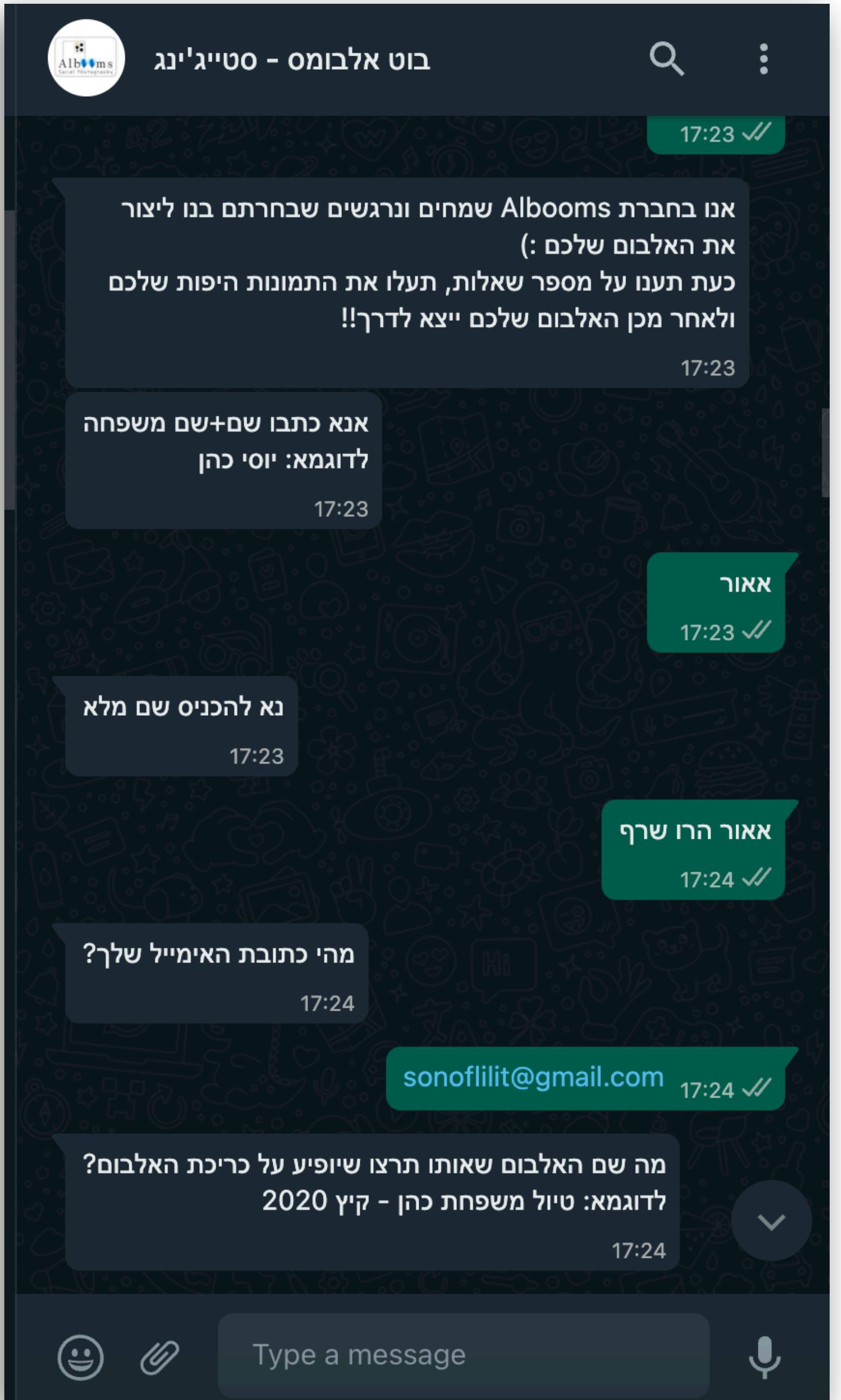
# נוצר מטור שיחה קל לעור

## מתעדכו בכל הרצה



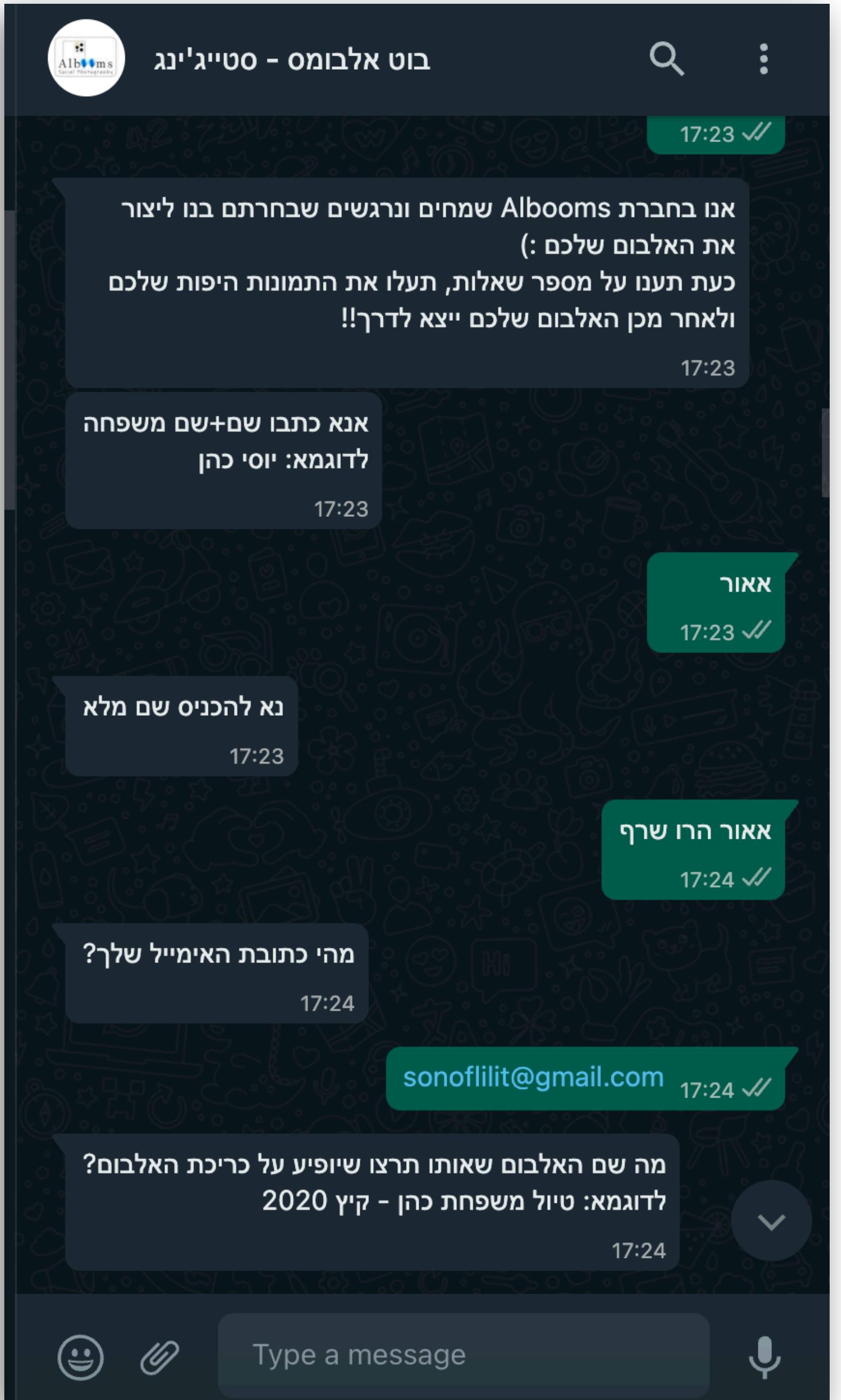
- להקמת אירוּע
- B: -
- חרתם בנו להקים את האירוּע הבא שלכם \*Albooms\* אנו בחברת Albooms שמחים ונרגשים שבחרתם בנו ליצור את האלבום שלכם:
- B: -
- אנא כתבו שם+שם משפחה
- לדוגמא: יוסי כהן
- א/or
- B: נא להכניס שם מלא:
- א/or הרו שרף
- B: מה כתובת המייל שלך?

**נוצר מטור שיחה  
קל לעורך  
מתעדכו בכל הרצה  
כל של git עושה הכל**



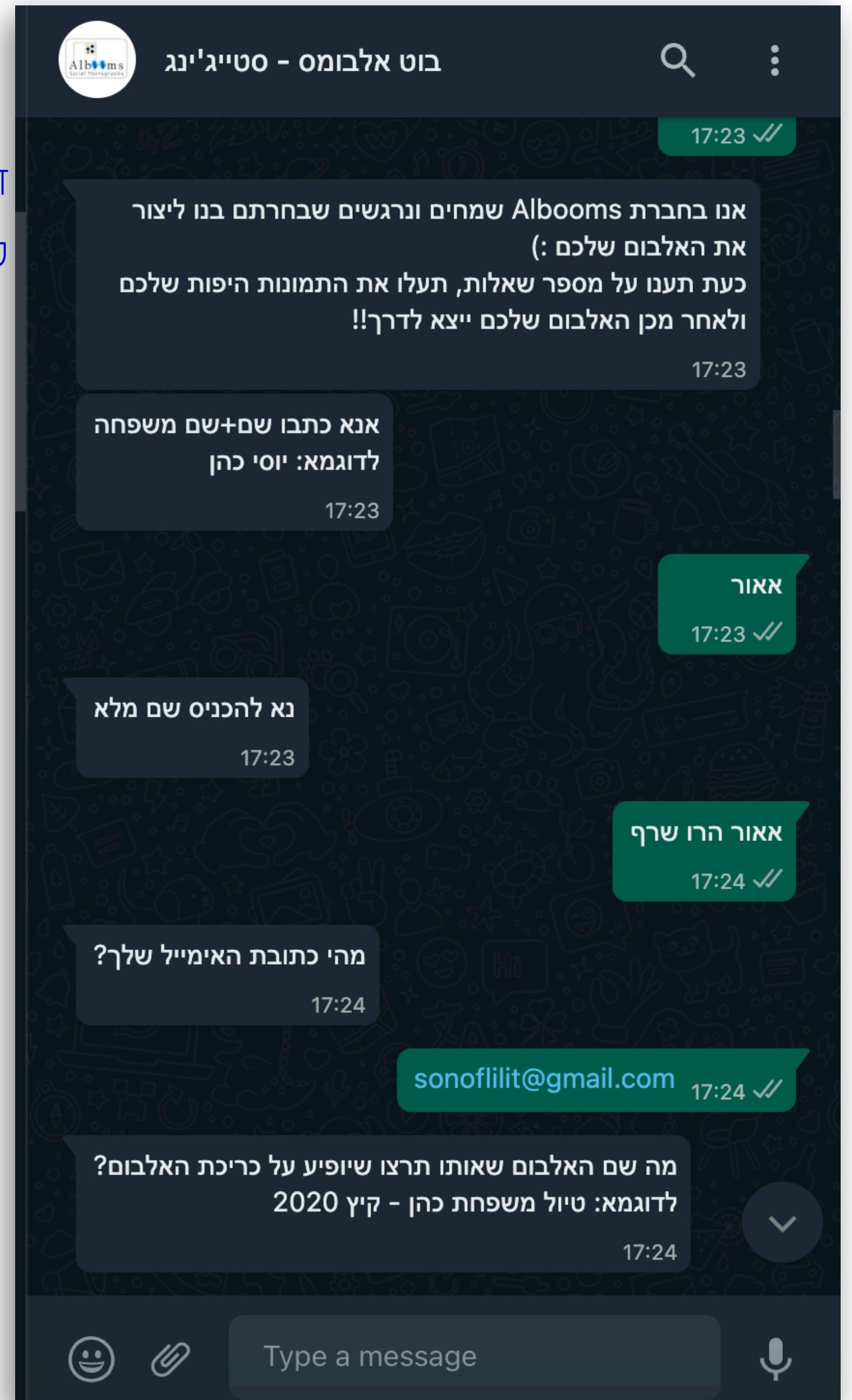
- להקמת אירו<sup>ע</sup>
- B: -
- חרתם בנו להקים את האירו הבא שלכם \*Albooms\* אנו בחברת Albooms שמחים ונרגשים שבחרתם בנו ליצור את האלבום שלכם:
- B: -
- אנא כתבו שם+שם משפחה
- לדוגמא: יוסי כהן
- א/or
- B: נא להכניס שם מלא :
- א/or הרו שף
- B: מה כתובת המייל שלך?

**נוצר מטור שיחה  
קל לעורך  
מטען בכל הרצה  
diff של git עושה הכל  
מפעיל את כל המסלול האמיתי**



- להקמת אירו<sup>ע</sup>
- B: -
- חרתם בנו להקים את האירו הבא שלכם \*Albooms\* אנו בחברת Albooms שמחים ונרגשים שבחרתם בנו ליצור את האלבום שלכם:
- B: -
- אנא כתבו שם+שם משפחה
- לדוגמא: יוסי כהן
- א/or
- B: נא להכניס שם מלא:
- א/or הרו שרפ
- B: מה כתובת המייל שלך?

# אפשר להראות לך



- להקמת אירוּע
- B: -
- חרתם בנו להקים את האירוּע הבא שלכם \*Albooms\* אנו בחברת עת תענו על מספר שאלות ולאחר מכן האירוּע שלכם י יצא לדרך
- B: -
- אנא כתבו שם+שם משפחה
- לדוגמא: יוסי כהן
- א/or
- B: נא להכניס שם מלא :
- א/or הרו שרף
- B: מה כתובת המייל שלך ?

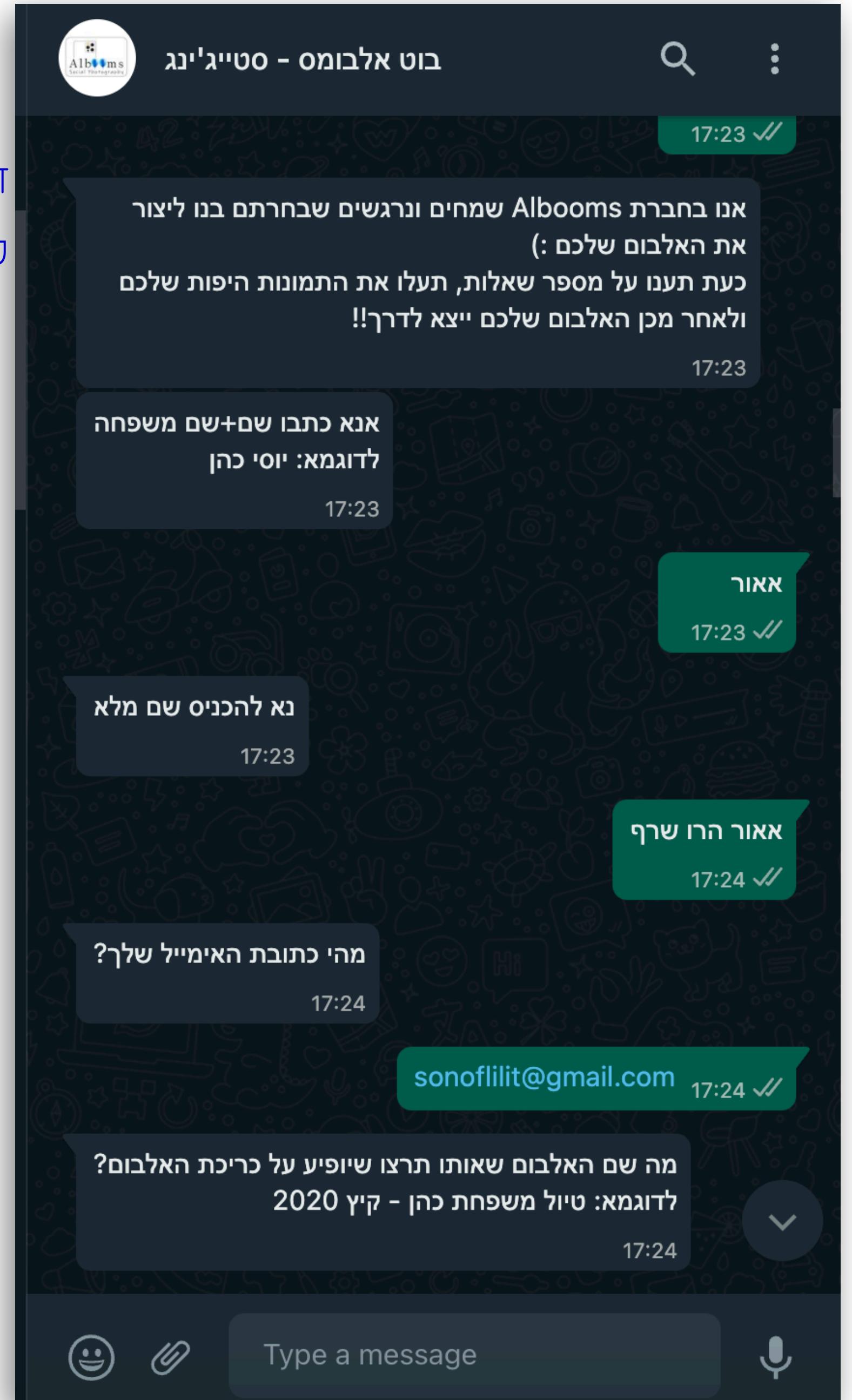
הרביה וואו

**אפשר להראות ללקוח**

התרשומות

מאד YAML

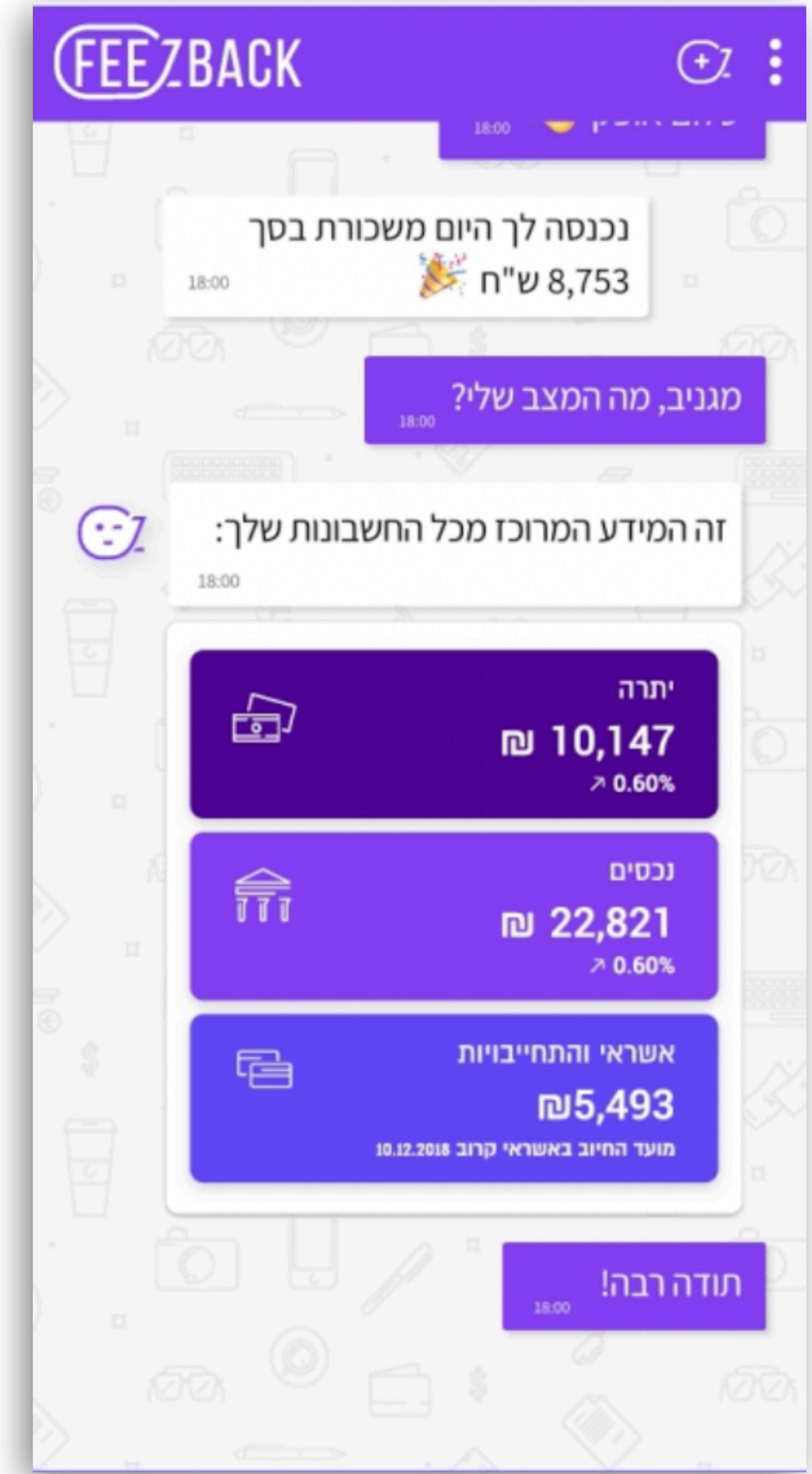
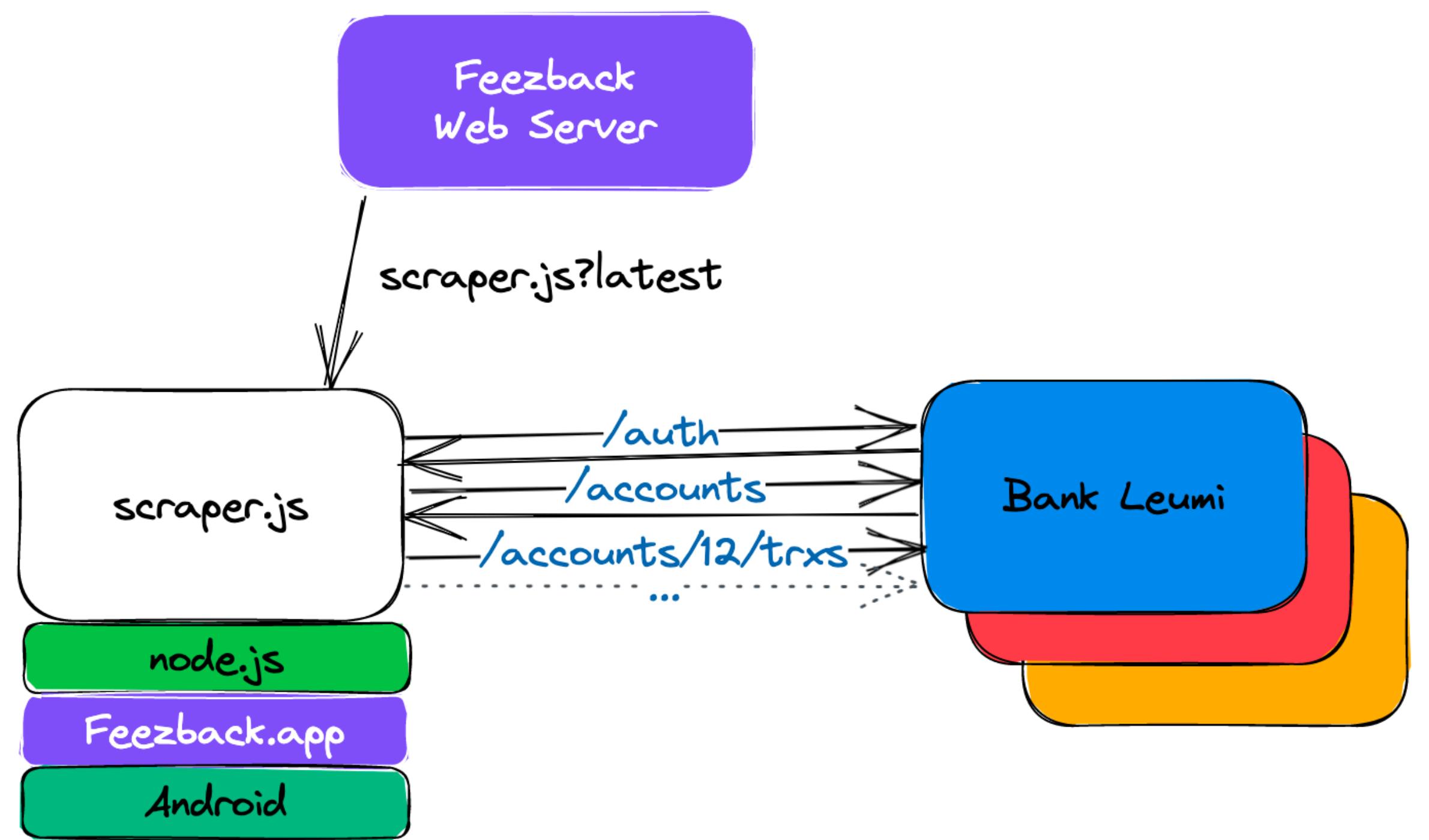
כה טכנולוגית



**django-whatsapp-business-is  
git**

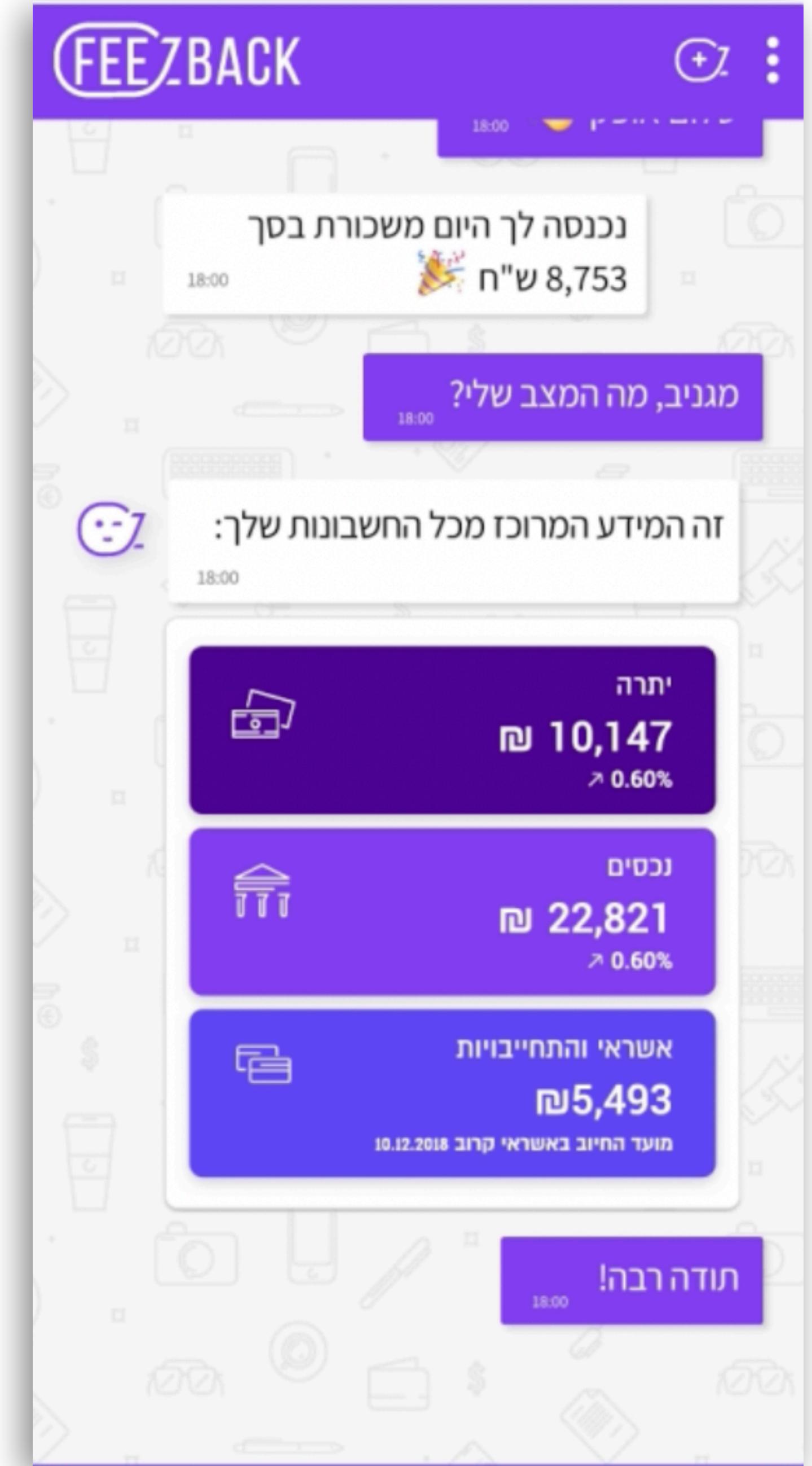
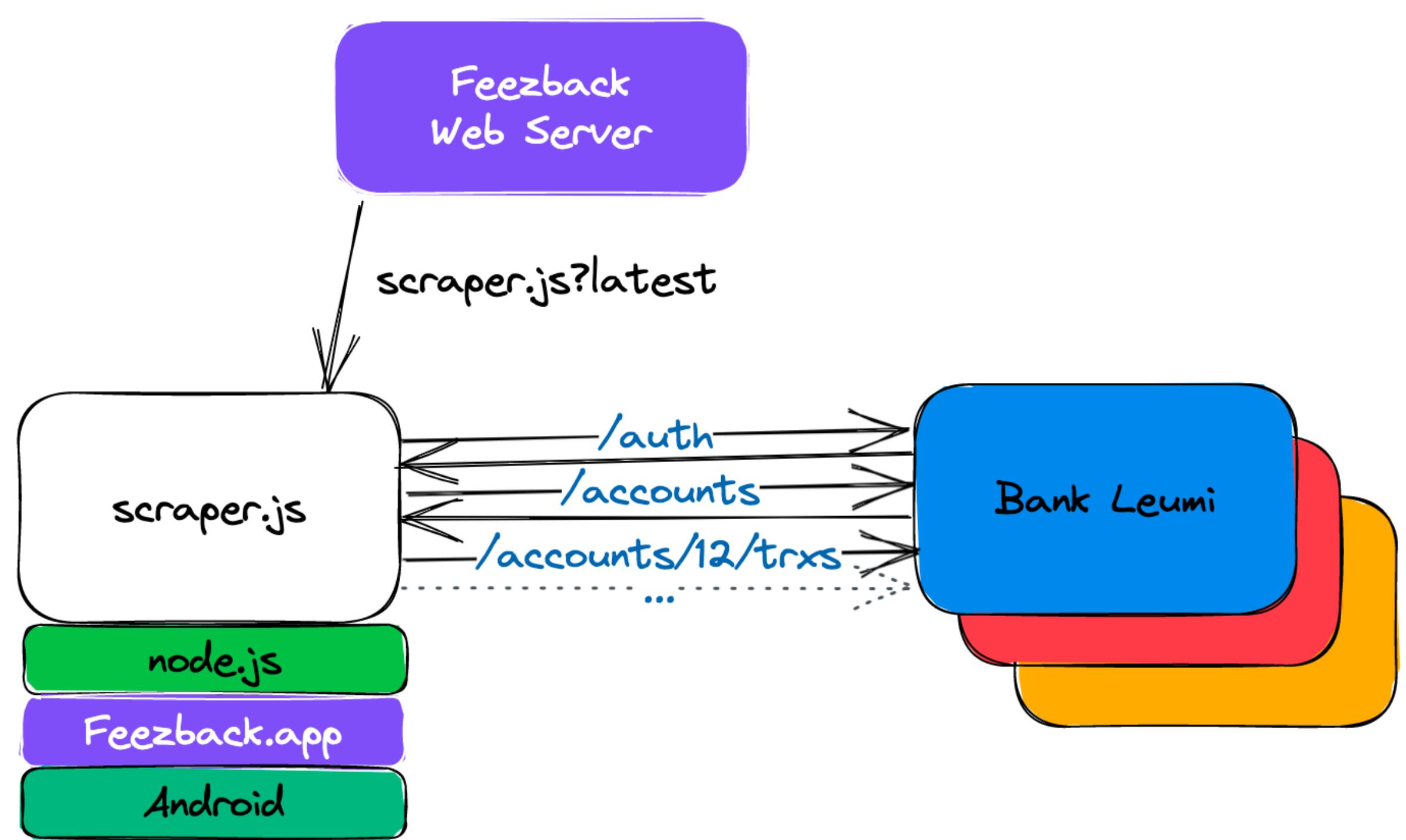
**pytest-gold, pytest-datarecorder,  
pytest-snapshot, snappiershot,  
jest, snap-shot-it...  
midas (mine, alpha)**

# סקריופינג לבנקים



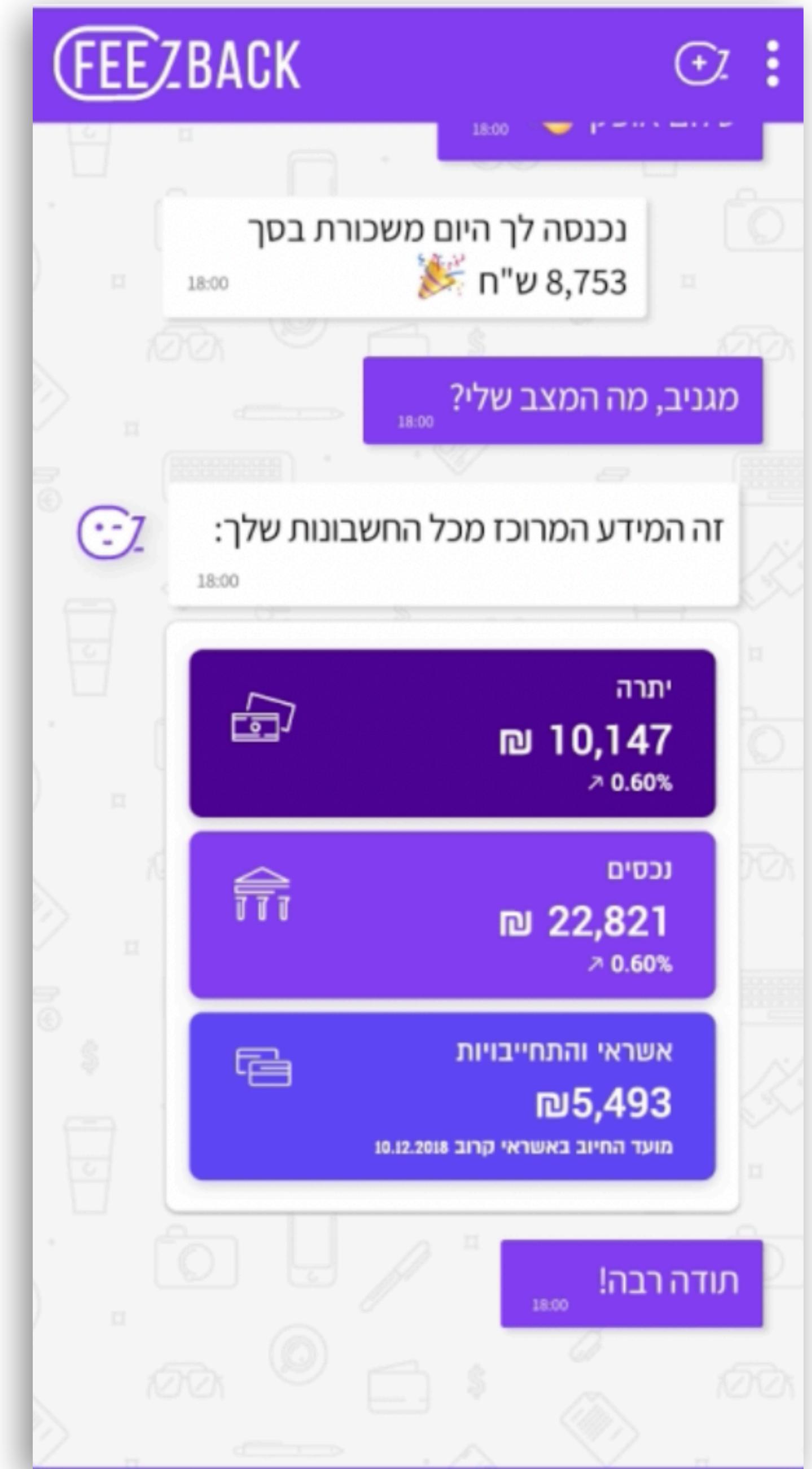
# סקריופינג לבנקים

לפעמים הקוד משתנה  
לפעמים אתר הבנק משתנה



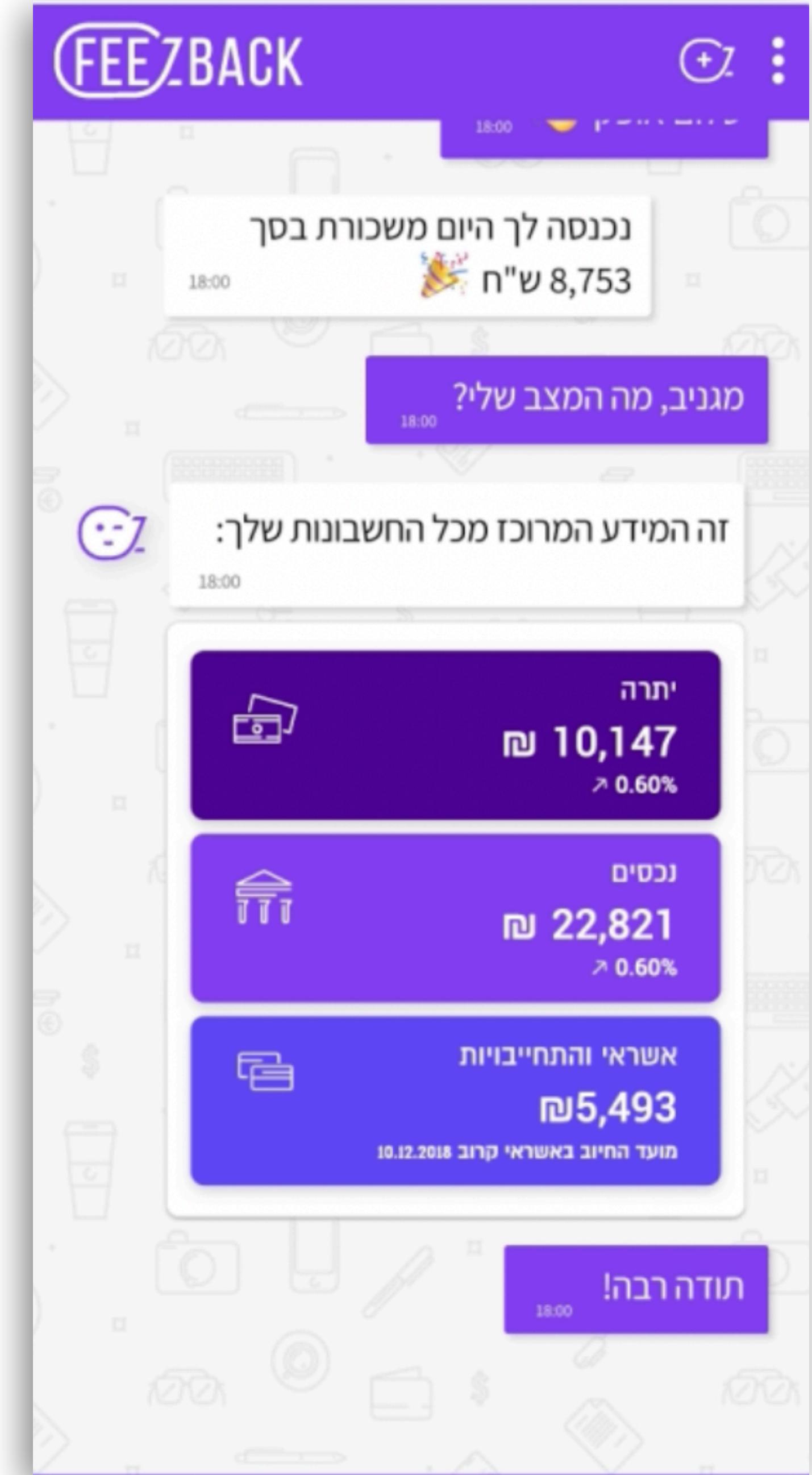
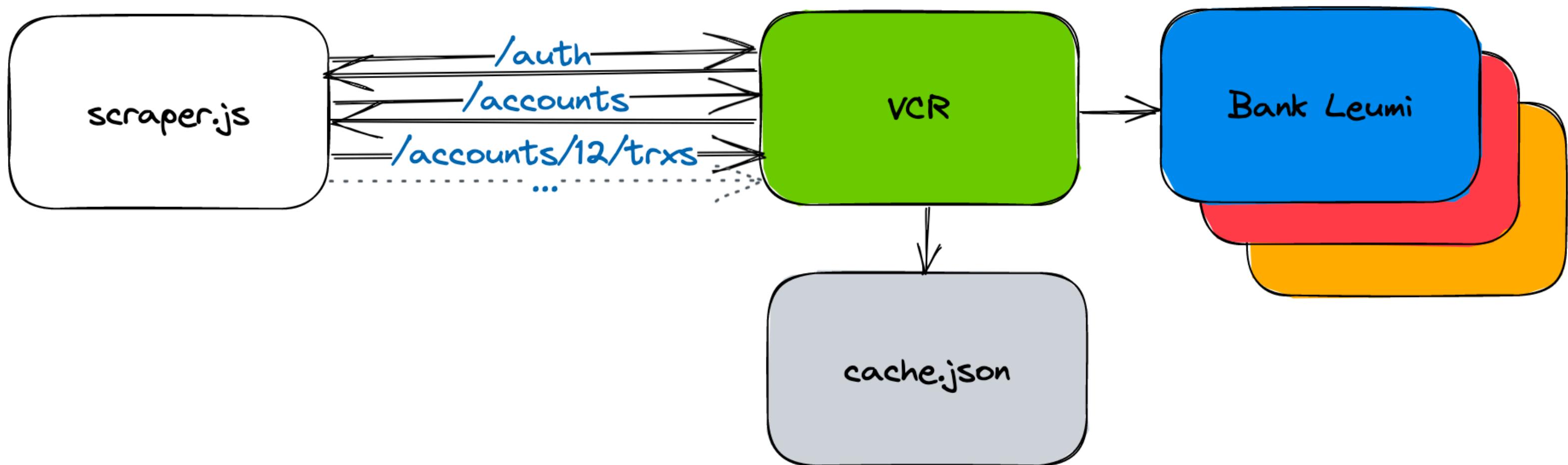
# סקריופינג לבנקים

לפעמים הקוד משתנה  
לפעמים אתר הבנק משתנה



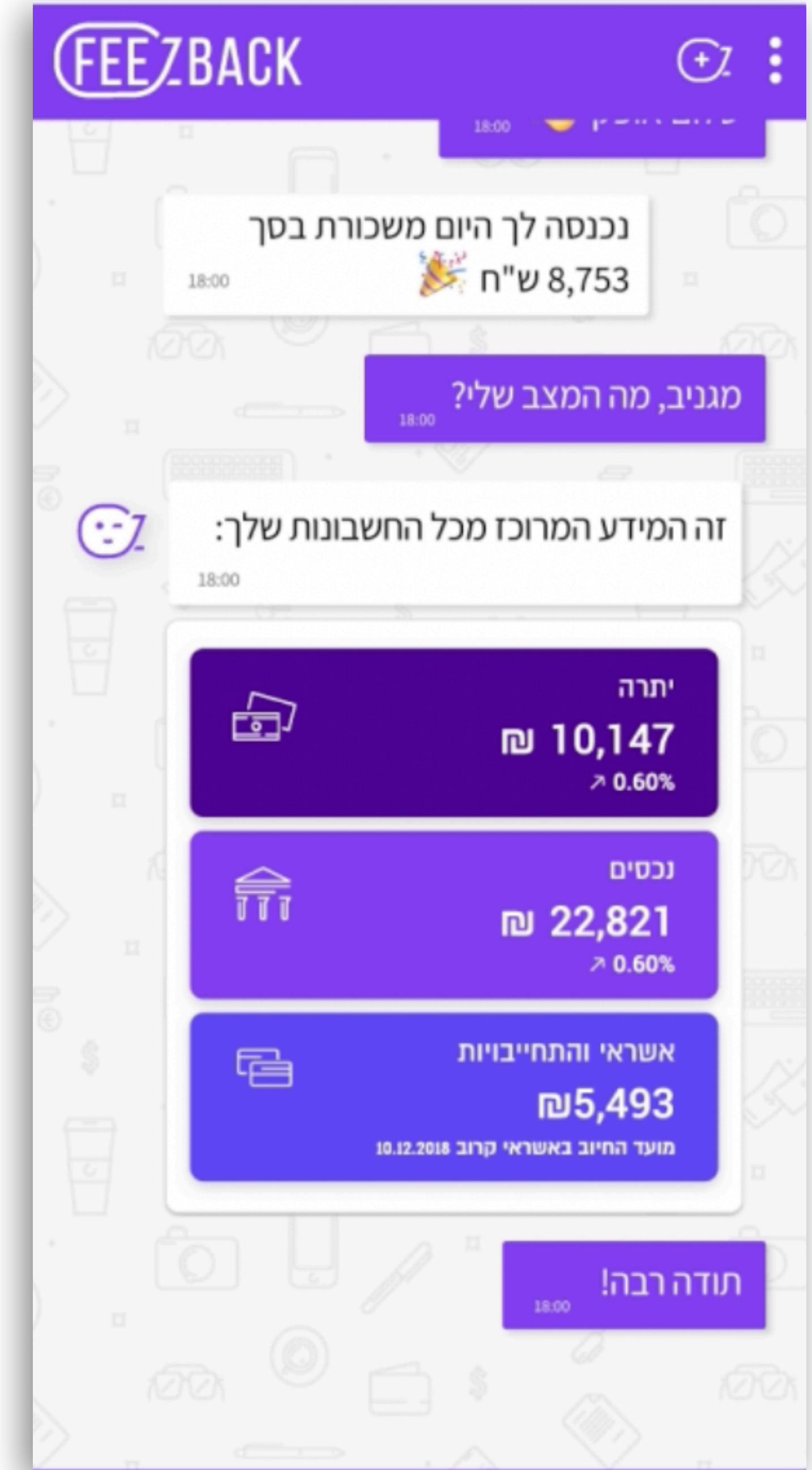
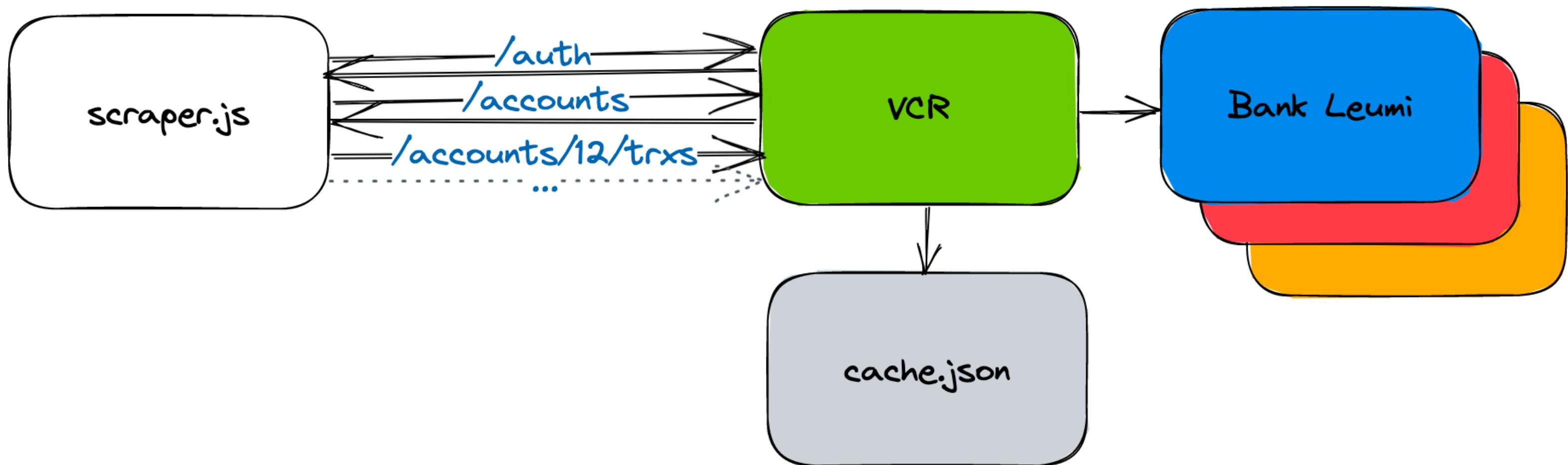
# VCR Test

עושים proxy  
 ל http client



# VCR Test

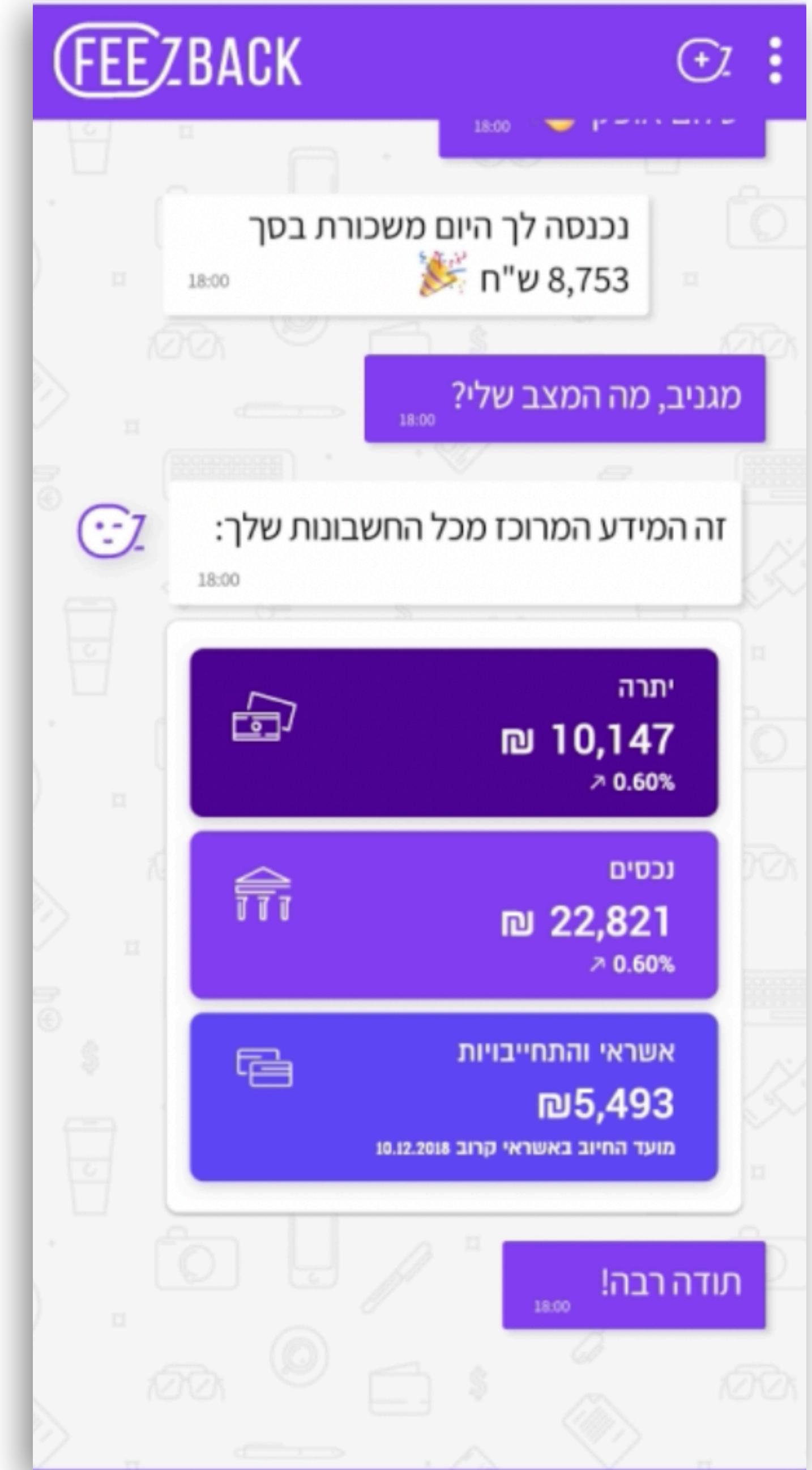
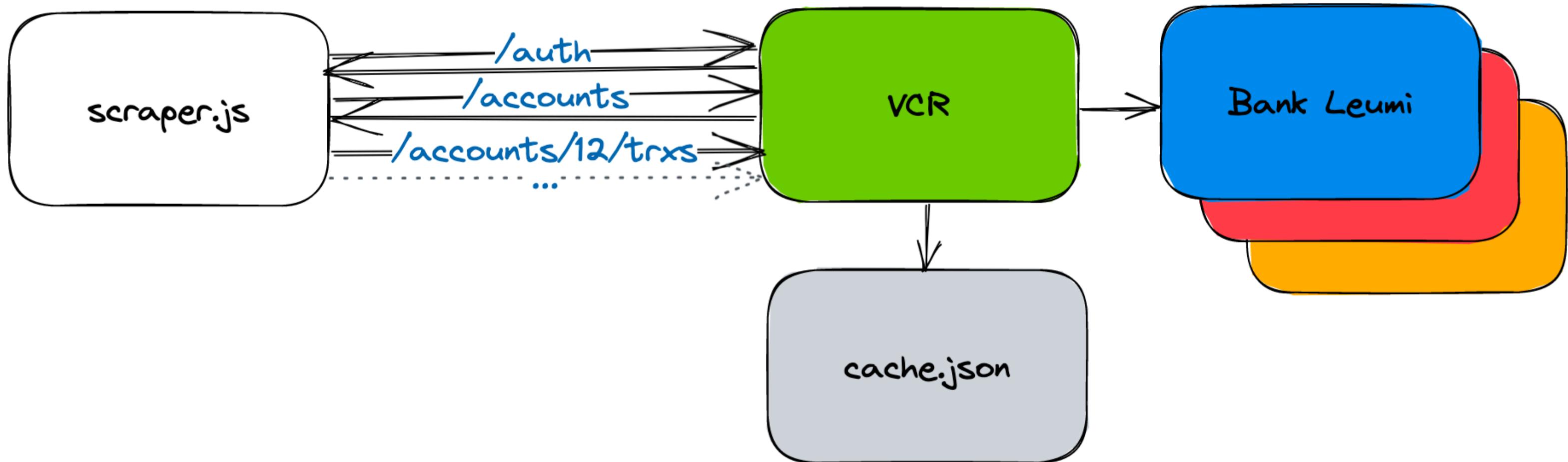
זה יוניס טסטע?



# VCR Test

זה יוניט טסט?

?integration test זה

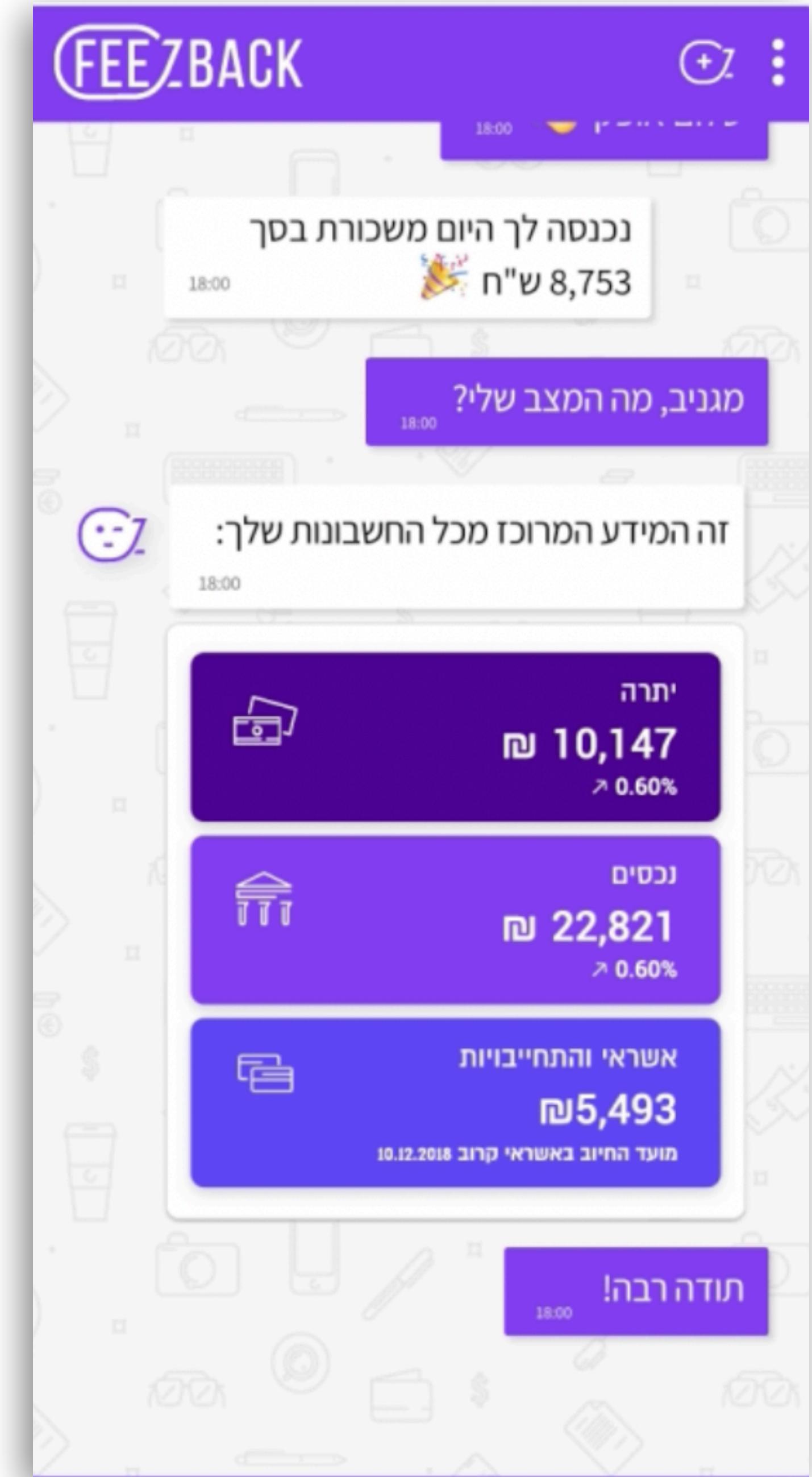
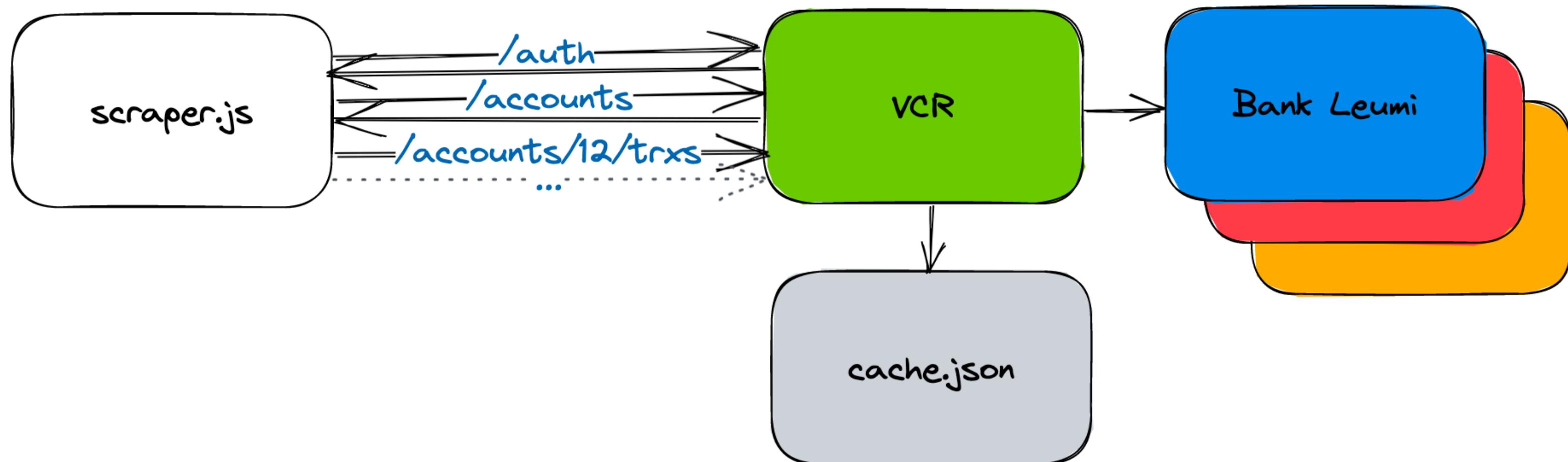


# VCR Test

זה יוניס טסט?

?integration test זה

לא, זה VCR test



**vcr**  
**pytest-vcr**  
**node-vcr**

# RegEx- מודרני לתחבר,kleenexp

```
import ke

username = input('Choose a username:')
# if not re.match(r'[A-Za-z][A-Za-z\d]*$', password):
if not ke.match('#letter [0+ [#letter | #digit]] #end_string', username):
    print("Invalid username")
else:
    password = input('Enter a new password:')
    # if re.match(r'\A(?=.*[a-z]*[a-z])(?=.*[A-Z]*[A-Z])(?=.*\D*(\d))(?!.*(?:123|pass|Pass))\w{6,}\Z',
password):
        if not ke.match('''[
            #has_lower=[lookahead [0+ not #lowercase] #lowercase]
            #has_upper=[lookahead [0+ not #uppercase] #uppercase]
            #has_digit=[lookahead [0+ not #digit] [capture #digit]]
            #no_common=[not lookahead [0+ #any] ["123" | "pass" | "Pass"]]
            #start_string #has_lower #has_upper #has_digit #no_common [6+ #token_character] #end_string
        ''', password):
            print("Password should have at least one uppercase letter, one lowercase, one digit.")
            print("And nothing obvious like '123'")
        else:
            ...
    
```

# תחביר מודרני ל-RegEx, kleenexp

```
import ke

username = input('Choose a username:')
# if not re.match(r'[A-Za-z][A-Za-z\d]*$', password):
if not ke.match('#letter [0+ [#letter | #digit]] #end_string', username):
    print("Invalid username")
else:
    password = input('Enter a new password.')
    # if re.match(r'\A(?=.*[a-z]*[a-z])(?=.*[A-Z]*[A-Z])(?=.*\D*(\d))(?!.*(?:123|pass|Pass))\w{6,}\Z', password):
        if not ke.match(''[
            #has_lower=[lookahead [0+ not #lowercase] #lowercase]
            #has_upper=[lookahead [0+ not #uppercase] #uppercase]
            #has_digit=[lookahead [0+ not #digit] [capture #digit]]
            #no_common=[not lookahead [0+ #any] ["123" | "pass" | "Pass"]]
            #start_string #has_lower #has_upper #has_digit #no_common [6+ #token_character] #end_string
        ]'', password):
            print("Password should have at least one uppercase letter, one lowercase, one digit.")
            print("And nothing obvious like '123'")
        else:
            ...
    else:
        print("Invalid password")
```

# RegEx, תחבר מודרני ל-kleenexp

```
import ke

username = input('Choose a username:')
# if not re.match(r'[A-Za-z][A-Za-z\d]*$', password):
if not ke.match('#letter [0+ [#letter | #digit]] #end_string', username):
    print("Invalid username")
else:
    password = input('Enter a new password.')
    # if re.match(r'\A(?=.*[a-z]*[a-z])(?=.*[A-Z]*[A-Z])(?=.*\d)(?!.*(?:123|pass|Pass))\w{6,}\Z',
password):
        if not ke.match("[#lowercase] #uppercase #digit #no_common [6+ #token_character] #end_string",
#has_lower=[lookahead [0+ not #lowercase] #lowercase]
#has_upper=[lookahead [0+ not #uppercase] #uppercase]
#has_digit=[lookahead [0+ not #digit] [capture #digit]]
#no_common=[not lookahead [0+ #any] ["123" | "pass" | "Pass"]]
#start_string #has_lower #has_upper #has_digit #no_common [6+ #token_character] #end_string
]", password):
            print("Password should have at least one uppercase letter, one lowercase, one digit.")
            print("And nothing obvious like '123'")
        else:
            ...
```

**מתרגם ל-regex רגיל, רץ באוטומנו**

**שימוש בפיתון**

**שימוש ב-rust שימוש מ-js**

**בואר נאפשר שימוש ב-rust מפייתונו!**

# RegEx, תחביר מודרני ל-kleenexp

```
import ke

username = input('Choose a username:')
# if not re.match(r'[A-Za-z][A-Za-z\d]*$', password):
if not ke.match('#letter [0+ [#letter | #digit]] #end_string', username):
    print("Invalid username")
else:
    password = input('Enter a new password.')
    # if re.match(r'\A(?=.*[a-z]*[a-z])(?=.*[A-Z]*[A-Z])(?=.*\d)(?!.*(?:123|pass|Pass))\w{6,}\Z',
password):
        if not ke.match("[#has_lower=[lookahead [0+ not #lowercase] #lowercase]
[#has_upper=[lookahead [0+ not #uppercase] #uppercase]
[#has_digit=[lookahead [0+ not #digit] [capture #digit]]
[#no_common=[not lookahead [0+ #any] ["123" | "pass" | "Pass"]]

#start_string #has_lower #has_upper #has_digit #no_common [6+ #token_character] #end_string
]", password):
            print("Password should have at least one uppercase letter, one lowercase, one digit.")
            print("And nothing obvious like '123'")
else:
    ...
```

**מתרגם ל-regex רגיל, רץ באותו מנוע**

**שימוש בפיתון**

**שימוש ב-rust שימוש מ-js**

**בואר נאפשר שימוש במימוש ב-rust מפיזיתו!**

**אפשר לכתוב הרבה ביטויים ולעשות !gold testing**

# RegEx, תחביר מודרני ל-kleenexp

```
import ke
```

```
username = input('Choose a username: ')
# if not re.match(r'[A-Za-z][A-Za-z\d]*$', password):
if not ke.match('#letter [0+ [#letter | #digit]] #end_string', username):
    print("Invalid username")
else:
    password = input('Enter a new password.')
    # if re.match(r'\A(?=.*[a-z]*[a-z])(?=.*[A-Z]*[A-Z])(?=.*\d)(?!.*(?:123|pass|Pass))\w{6,}\Z',
password):
```

**מתרגם ל-regex רגיל, רץ באוטומנו!**

**שימוש בפיתון**

**שימוש ב-rust שימוש מ-js**

**בואר נאפשר שימוש במימוש ב-rust מפייתו!**

**אפשר לכתוב הרבה ביטויים ולעשות !gold testing**

**בואר נתרגם טסטים של פיתון!**

```
#start_string #has_lower #has_upper #has_digit #no_common [6+ #token_character] #end_string
]'', password):
    print("Password should have at least one uppercase letter, one lowercase, one digit.")
    print("And nothing obvious like '123'")
else:
    ...
    ...
```

# אגב, vscode שימוש בפיזיטון וב-kleenexp

- Unicode matching  
#30 opened on Sep 19, 2022 by mikaelho
- Wrap Javascript Regex API and build npm package [help wanted](#) [typescript](#)  
#28 opened on Sep 2, 2022 by SonOfLilit
- Support Rust and RustFancy flavors [compiler](#) [completeness](#) [good-first-issue](#) [rust](#)  
#27 opened on Sep 2, 2022 by SonOfLilit
- Comb over Python regex docs and tests and make sure every feature is supported [compiler](#) [completeness](#)  
#26 opened on Sep 2, 2022 by SonOfLilit
- Inline flags [compiler](#) [completeness](#) [good-first-issue](#) [python](#) [rust](#)  
#25 opened on Sep 2, 2022 by SonOfLilit
- Backreferences [compiler](#) [completeness](#) [good-first-issue](#) [python](#) [rust](#)  
#24 opened on Sep 2, 2022 by SonOfLilit
- Syntax for non-greedy quantifiers [compiler](#) [completeness](#) [good-first-issue](#) [python](#) [rust](#)  
#23 opened on Sep 2, 2022 by SonOfLilit

contributions welcome

**kleenexp**  
**ryo3**  
**CPython**

**از מה למדנו היום?**

לפעמים אפשר לקבל המון  
בדיקות טובות בחינוך

אז מה למדנו היום?

**בבדיקות טובות בוחנים  
לפעמים אפשר לקבל המון**

**זה עשויל לעשות את ההבדל בין  
פרויקט מוצלח לכשלונו מהדריך**

**אז מה למדנו היום?**

לפעמים אפשר לקבל המון  
בדיקות טובות בחינם

זה עשוily לעשות את ההבדל בין  
פרויקט מוצלח לכשלונו מהדריך

## אז מה למדנו היום?

לפעמים כדאי לשקיע טיפה  
ב-DSL טבעי לבטא בו  
בדיקות

לפעמים אפשר לקבל המון  
בדיקות טובות בחינם

זה עשוily לעשות את ההבדל בין  
פרויקט מוצלח לכשלונו מהדרה

## אז מה למדנו היום?

לפעמים כדאי לשקיע טיפה  
ב-DSL טבעי לבטא בו  
בדיקות

git הוא כלי נחמד למימוש  
של tests golden תפורים  
לפרויקט

לפעים אפשר לקבל המון  
בדיקות טובות בחינם

זה עשו לעשות את ההבדל בין  
פרויקט מוצלח לכשלונו מהדרה

## אז מה למדנו היום?

VCR tests הופכות הרצת  
של המוצר לבדיקת  
אינטגרציה

לפעים כדי להשكيיע טיפה  
ב-DSL טבעי לבטא בו  
בדיקות

git הוא כלי נהדר למימוש  
של golden tests תפורים  
לפרויקט

לפעים אפשר לקבל המון  
בדיקות טובות בחינם

זה עשו לעשות את ההבדל בין  
פרויקט מוצלח לכשלונו מהדרה

## אז מה למדנו היום?

VCR tests הופכות הרצת  
של המוצר לבדיקת  
אינטגרציה

VCR tests הופכות בדיקת  
אינטגרציה לבדיקות יחידה  
מhireות

לפעים כדי להשكيיע טיפה  
ב-DSL טבעי לבטא בו  
בדיקות

git הוא כלי נהדר למימוש  
של tests golden תפורים  
לפרויקט

**לפעים אפשר לקבל המון  
בדיקות טובות בחינם**

**זה עשו לעשות את ההבדל בין  
פרויקט מוצלח לכשלו מהדרה**

## אז מה למדנו היום?

**VCR tests** הופכות הרצה  
של המוצר לבדיקת  
אינטגרציה

**VCR tests** הופכות בדיקת  
אינטגרציה לבדיקות יחידה  
מהירות

**בפרויקט הבא אולי נמציא 3  
תכיסים אחרים, אז תמיד  
לשאול את עצמנו:**

**איך הייתה  
בדיקות אותו?**

**לפעמים כדאי לשקיע טיפה  
ב-DSL טבעי לבטא בו  
בדיקות**

**git** הוא כלי נהדר למימוש  
של **golden tests** תפורים  
לפרויקט

**תמיד אפשר יותר טוב**

# אני אאור

אני מיעץ לסטארטאפים על טכנולוגיה וסטרטגיה.  
אני עושה מנטורינג ו-design reviews.  
אני פותר בעיות טכנולוגיות קשות.

## שאלות?

בימי רביעי בערב אני מעביר  
אימוני dev senior  
בחינם, בזום,  
לכיף שלי.

[bit.ly/auras-dojo](https://bit.ly/auras-dojo)