

# הפרויקט שלכם יקום או ייפול על תכסיסי בדיקות

אאור שרף, יועץ לCTOים

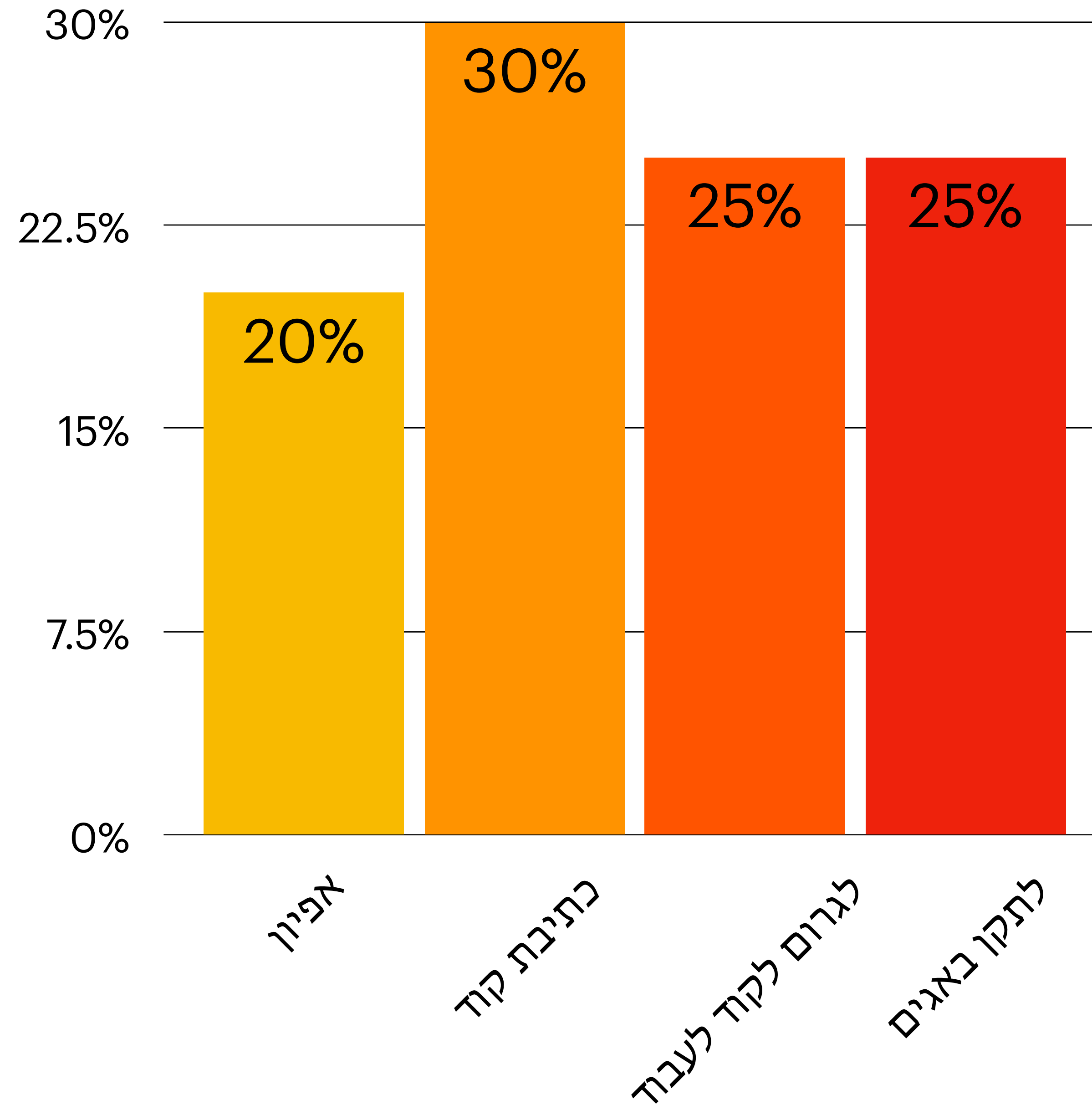
אוגוסט פינגווין 2023

**אוהבים לתכנת?**

**אוהבים לכתוב בדיקות?**

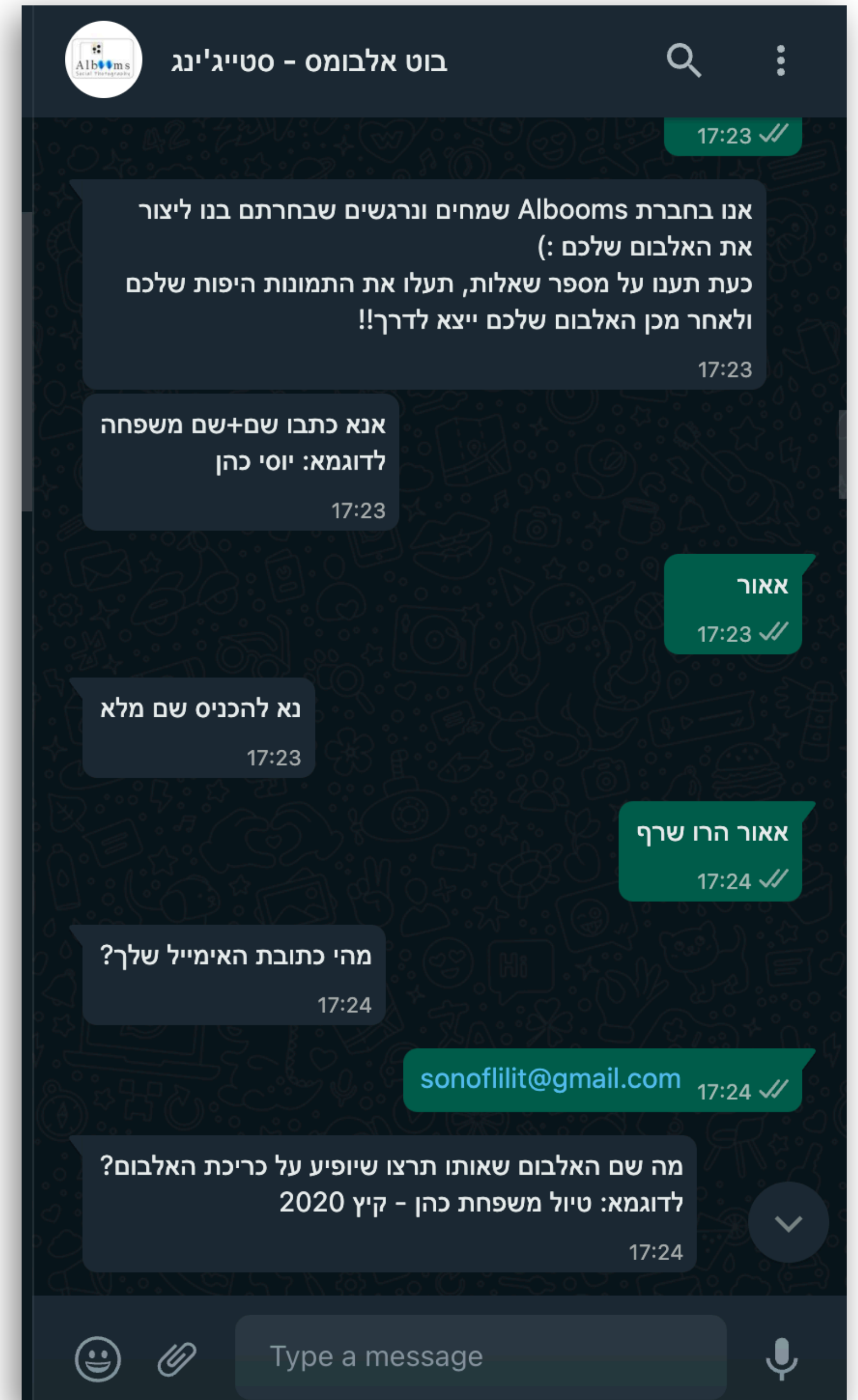
**אוהבים לדבג?**

# כמה זמן פיתוח מושקע ב...

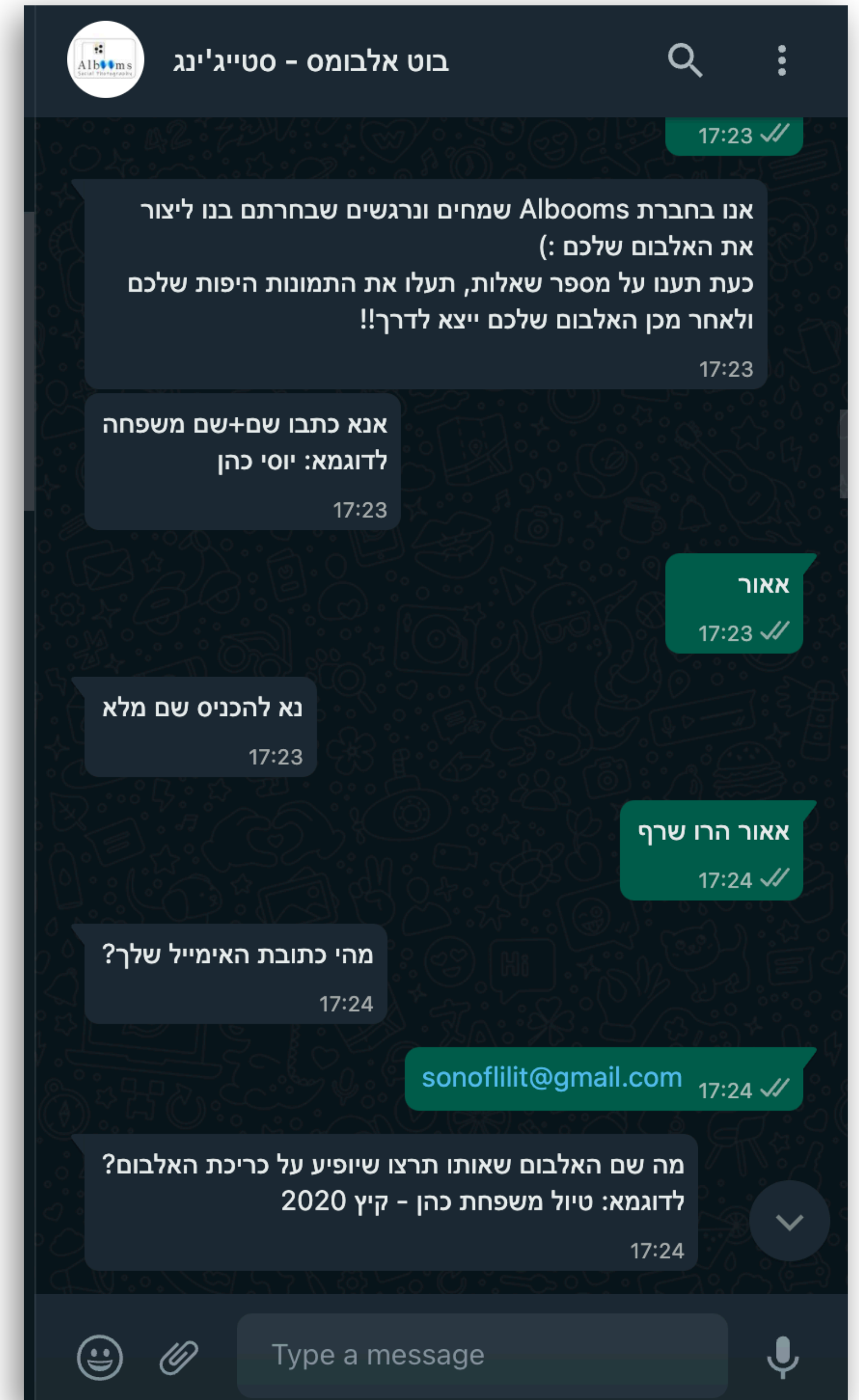
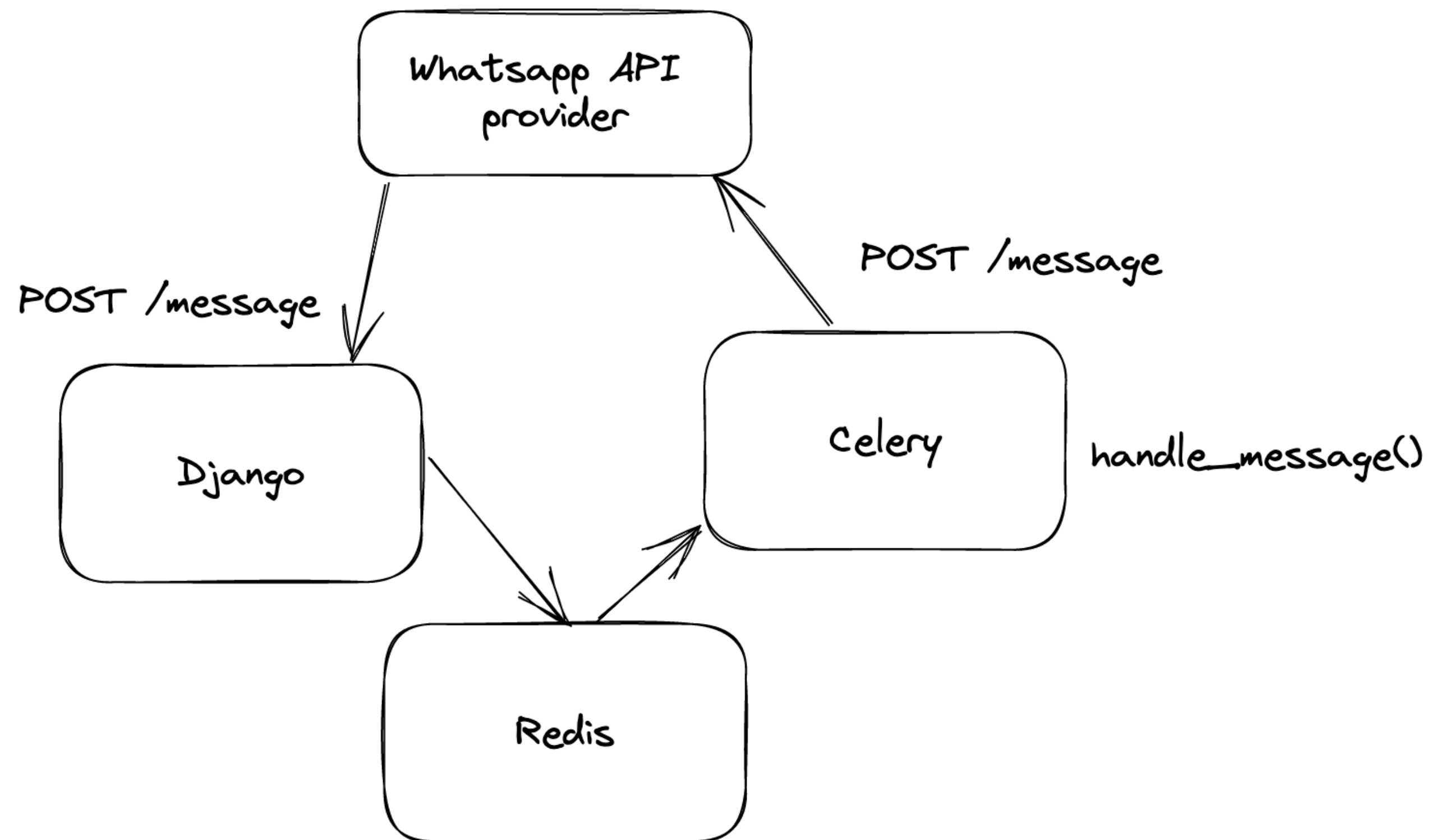


אפשר יותר טוב

# בוט ווטסאפ שמוכר דברים

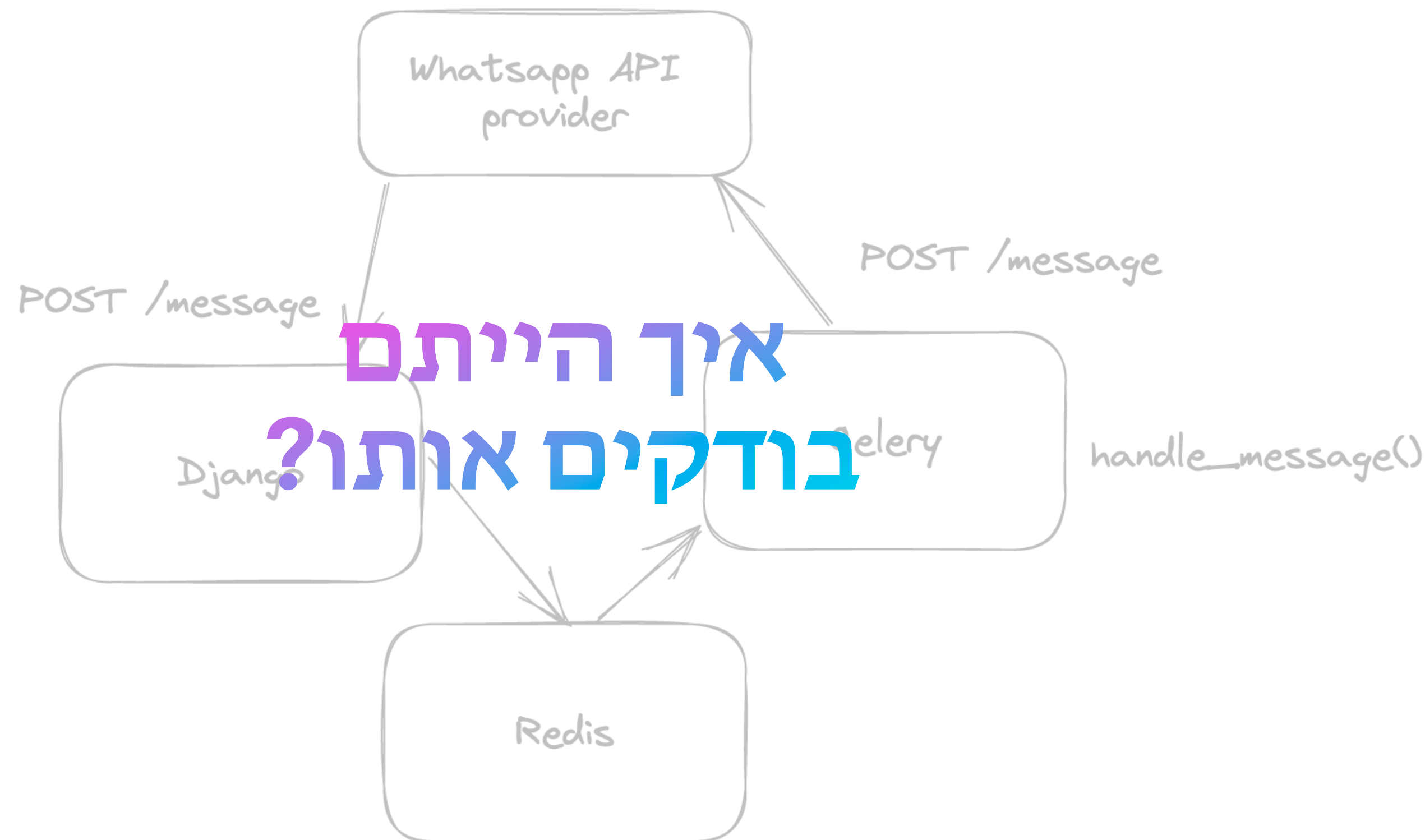


# בוט ווטסאפ שמוכר דברים

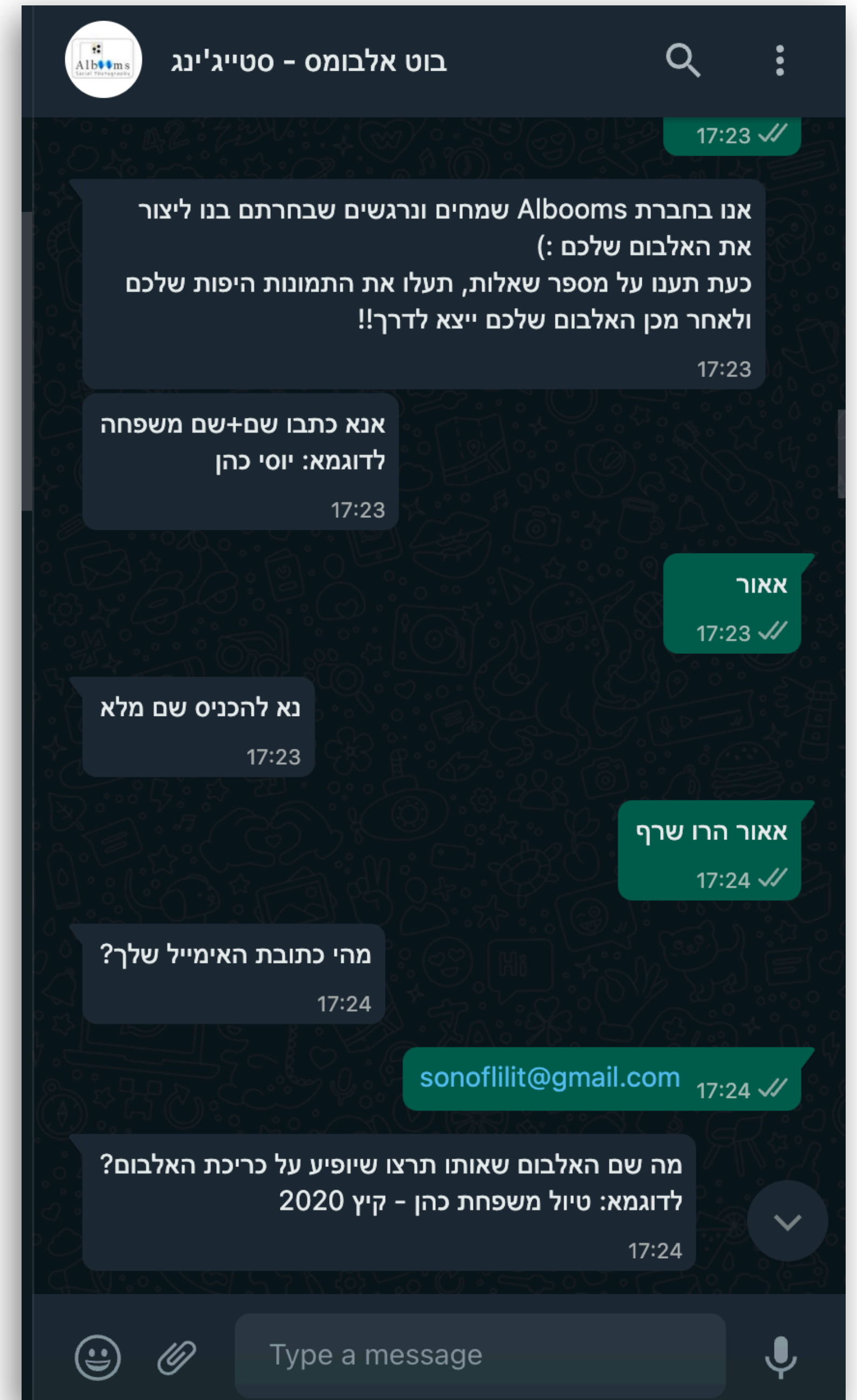




# בוט ווטסאפ שמוכר דברים



איך הייתם  
בודקים אותו?



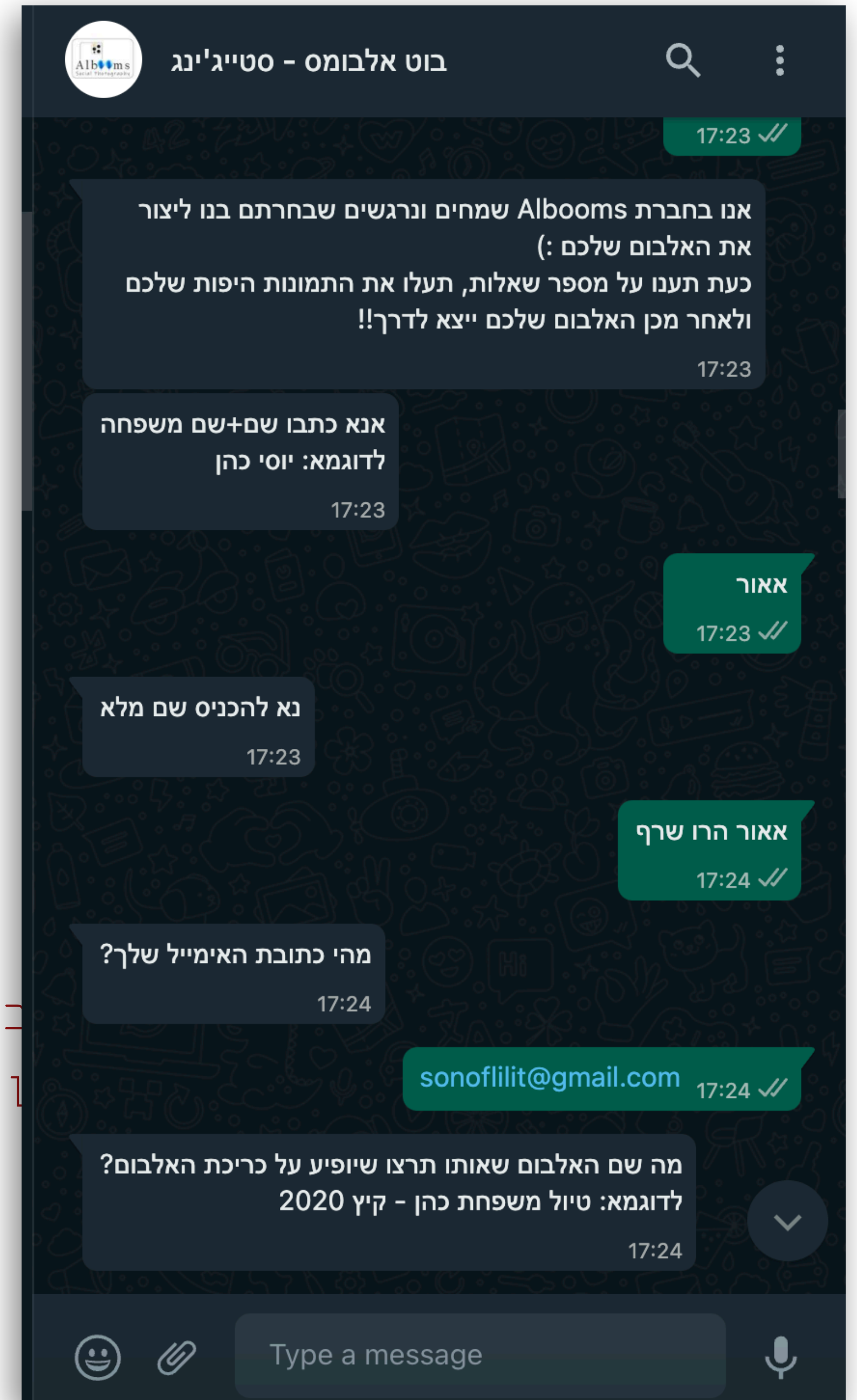
```

def send(text):
    return handle_message({
        "from": NUMBER, "id": "somebase64id", "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

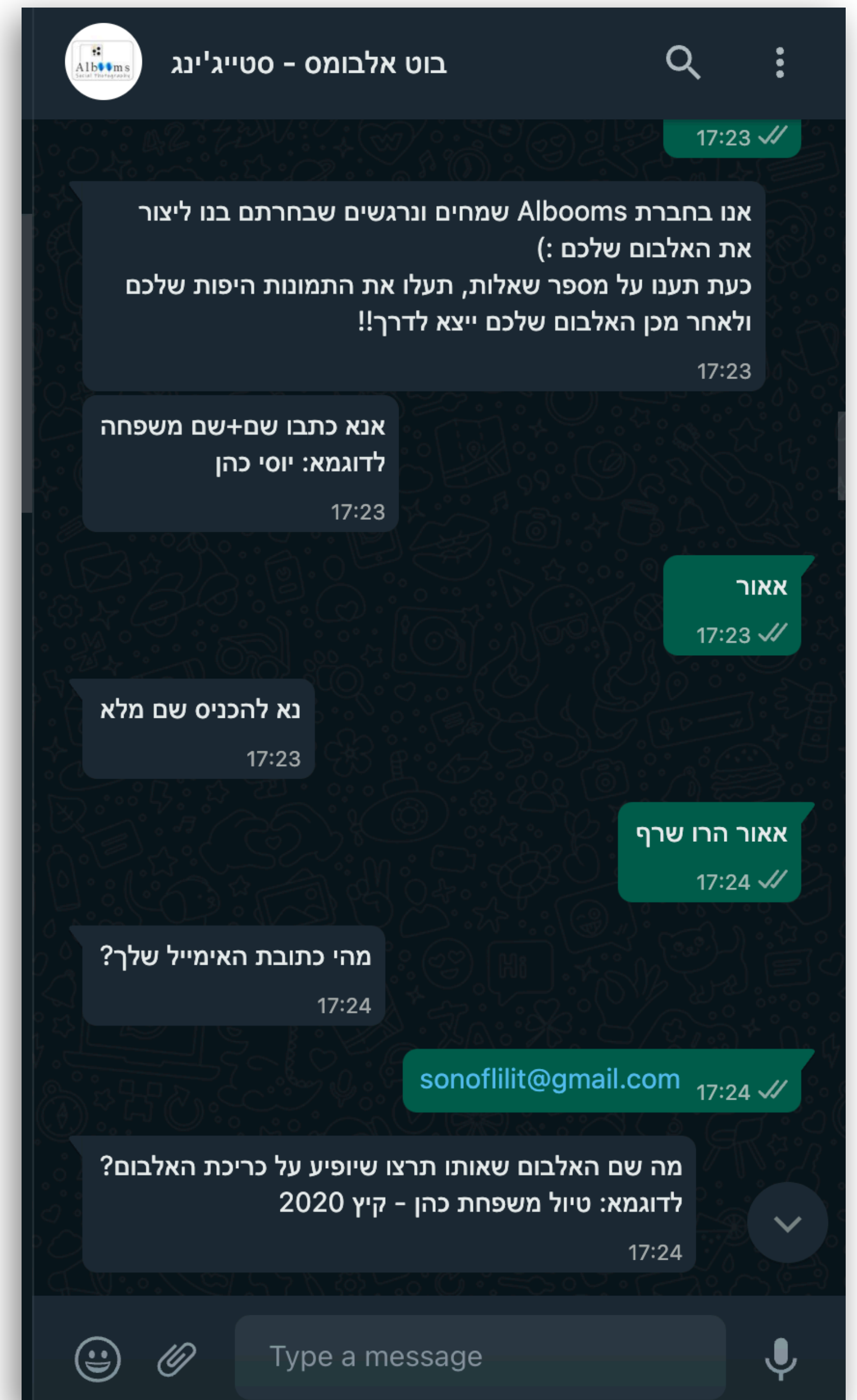
@patch('whatsapp_business_api.messages.send_message')
def test_form(send_message_mock):
    user = WaUser.objects.create(number=NUMBER, ...)
    event = Event.objects.create(client=user)

    send("גם אני רוצה אלבום")
    receive(send_message_mock, "בחרתם בנו Albooms אנו בחברת")
    receive(send_message_mock, "\nאנו אנו כתבו שם+שם משפחה")
    send("אאור")
    receive(send_message_mock, "נא להכניס שם מלא")
    send("אאור הרו שרף")
    receive(send_message_mock, "?מהי כתובת המייל שלך")

```



# אפשר יותר טוב



- להקמת אירוע

- B: | -

חזרתם בנו להקים את האירוע הבא שלכם \*Albooms\* אנו בחברת  
עת תענו על מספר שאלות ולאחר מכן האירוע שלכם ייצא לדרך

- B: | -

אנא כתבו שם+שם משפחה  
לדוגמא: יוסי כהן

- אאור

- B: נא להכניס שם מלא

- אאור הרו שרף

- B: ?מה כתובת המייל שלך?

- [sonoflilit@gmail.com](mailto:sonoflilit@gmail.com)

- B: ?מה גודל האלבום הרצוי

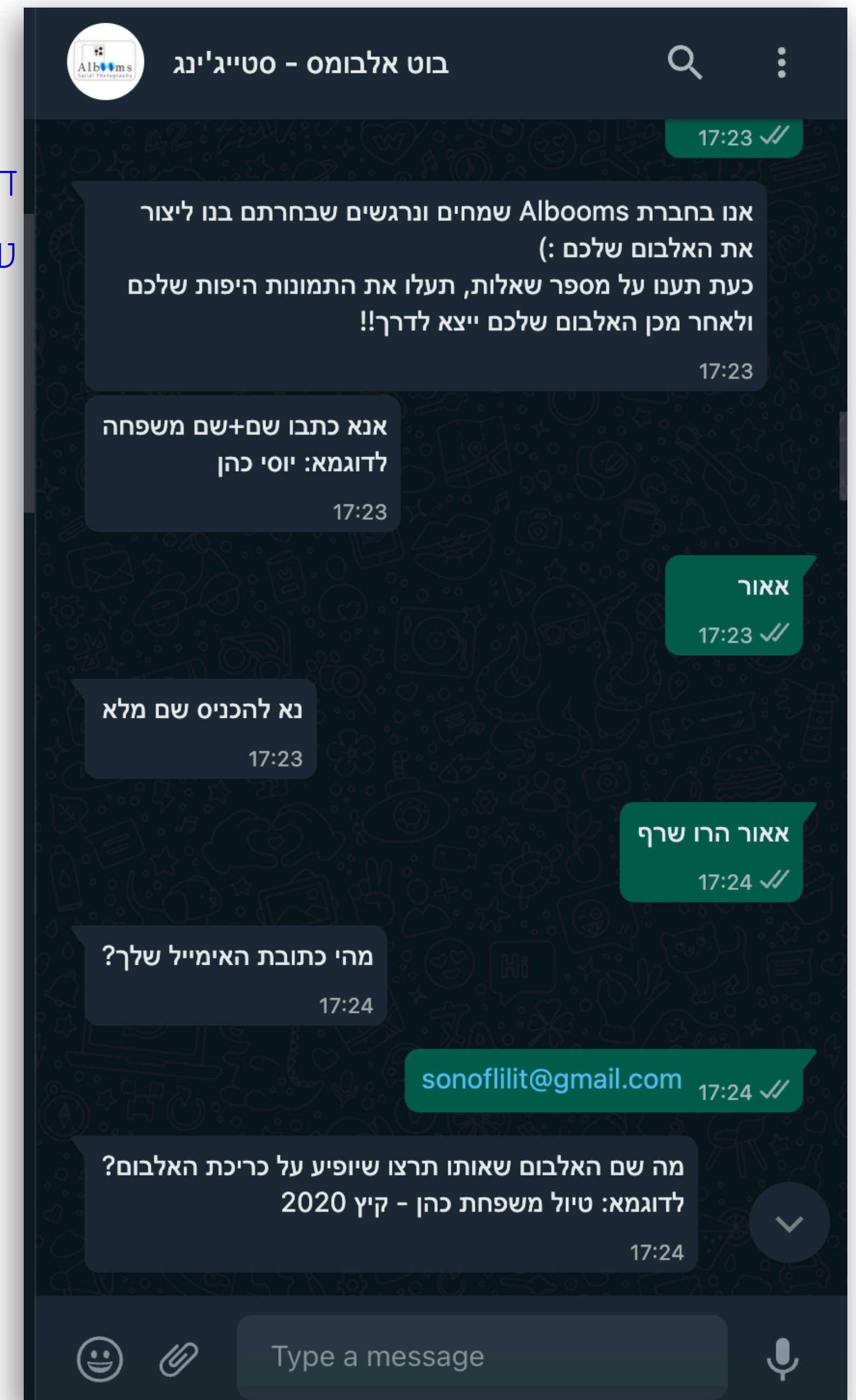
**buttons:**

- אלבום קטן

- אלבום בינוני

- אלבום גדול

- אלבום בינוני



- להקמת אירוע

- B: |-

חזרתם בנו להקים את האירוע הבא שלכם \*Albooms\* אנו בחברת  
עת תענו על מספר שאלות ולאחר מכן האירוע שלכם ייצא לדרך

- B: |-

אנא כתבו שם+שם משפחה

לדוגמא: יוסי כהן

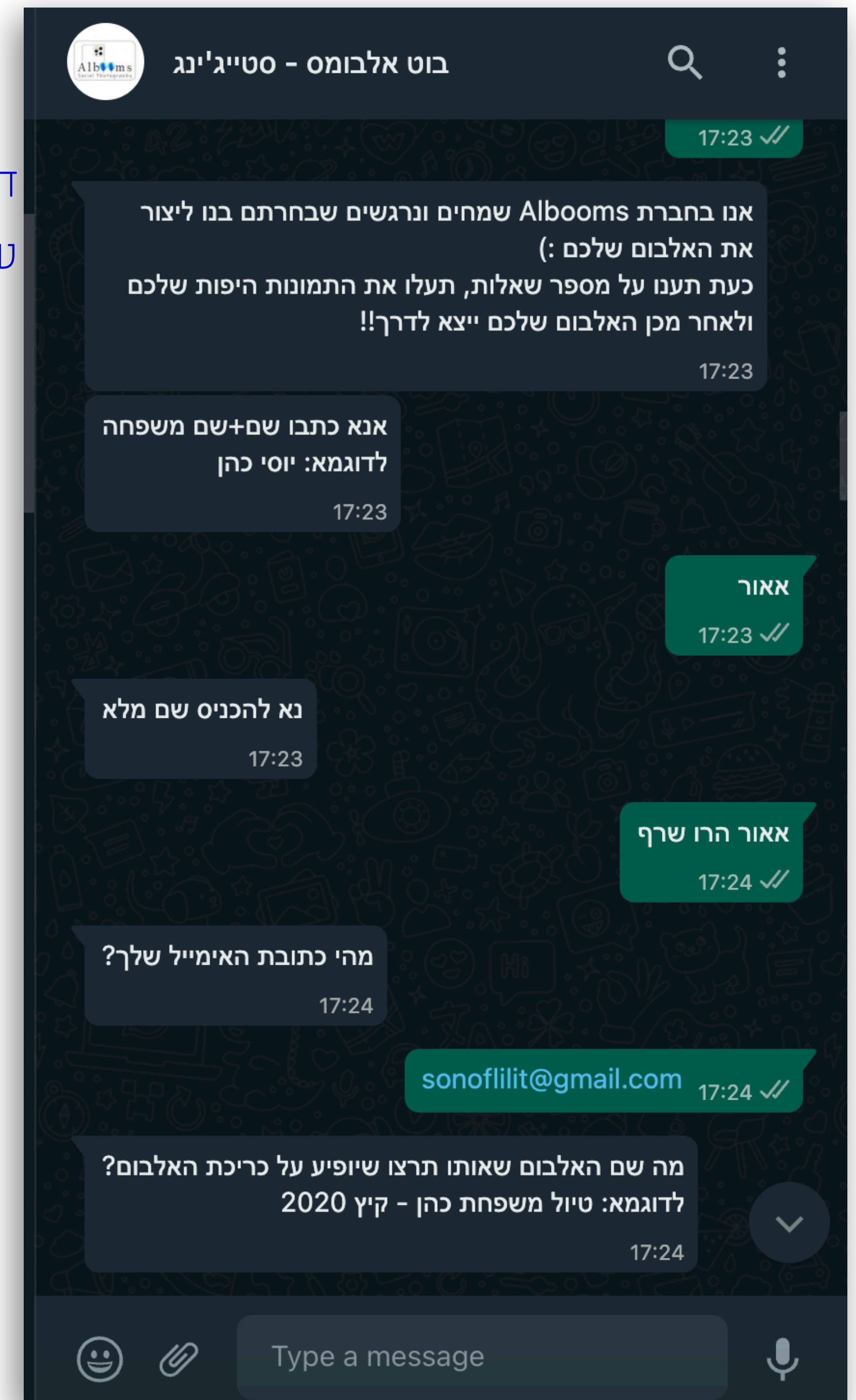
- אאור

- B: נא להכניס שם מלא

- אאור הרו שרף

- B: מה כתובת האימייל שלך?

## נוצר מתוך שיחה



- להקמת אירוע

- B: |-

חזרתם בנו להקים את האירוע הבא שלכם \*Albooms\* אנו בחברת  
עת תענו על מספר שאלות ולאחר מכן האירוע שלכם ייצא לדרך

- B: |-

אנא כתבו שם+שם משפחה

לדוגמא: יוסי כהן

- אאור

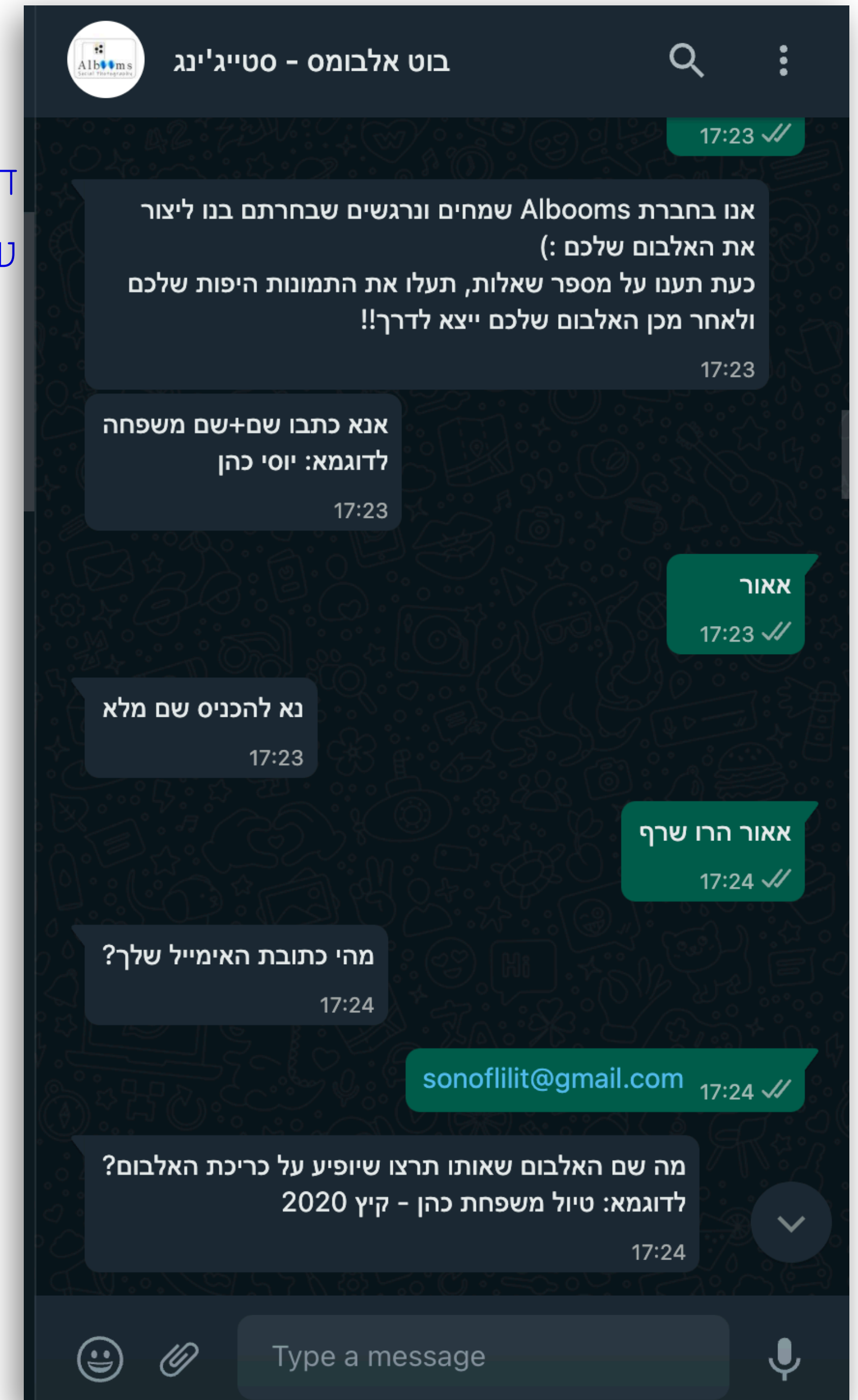
- B: נא להכניס שם מלא

- אאור הרו שרף

- B: מה כתובת האימייל שלך?

## נוצר מתוך שיחה

## קל לערוך



- להקמת אירוע

- B: |-

חזרתם בנו להקים את האירוע הבא שלכם \*Albooms\* אנו בחברת  
עת תענו על מספר שאלות ולאחר מכן האירוע שלכם ייצא לדרך

- B: |-

אנא כתבו שם+שם משפחה

לדוגמא: יוסי כהן

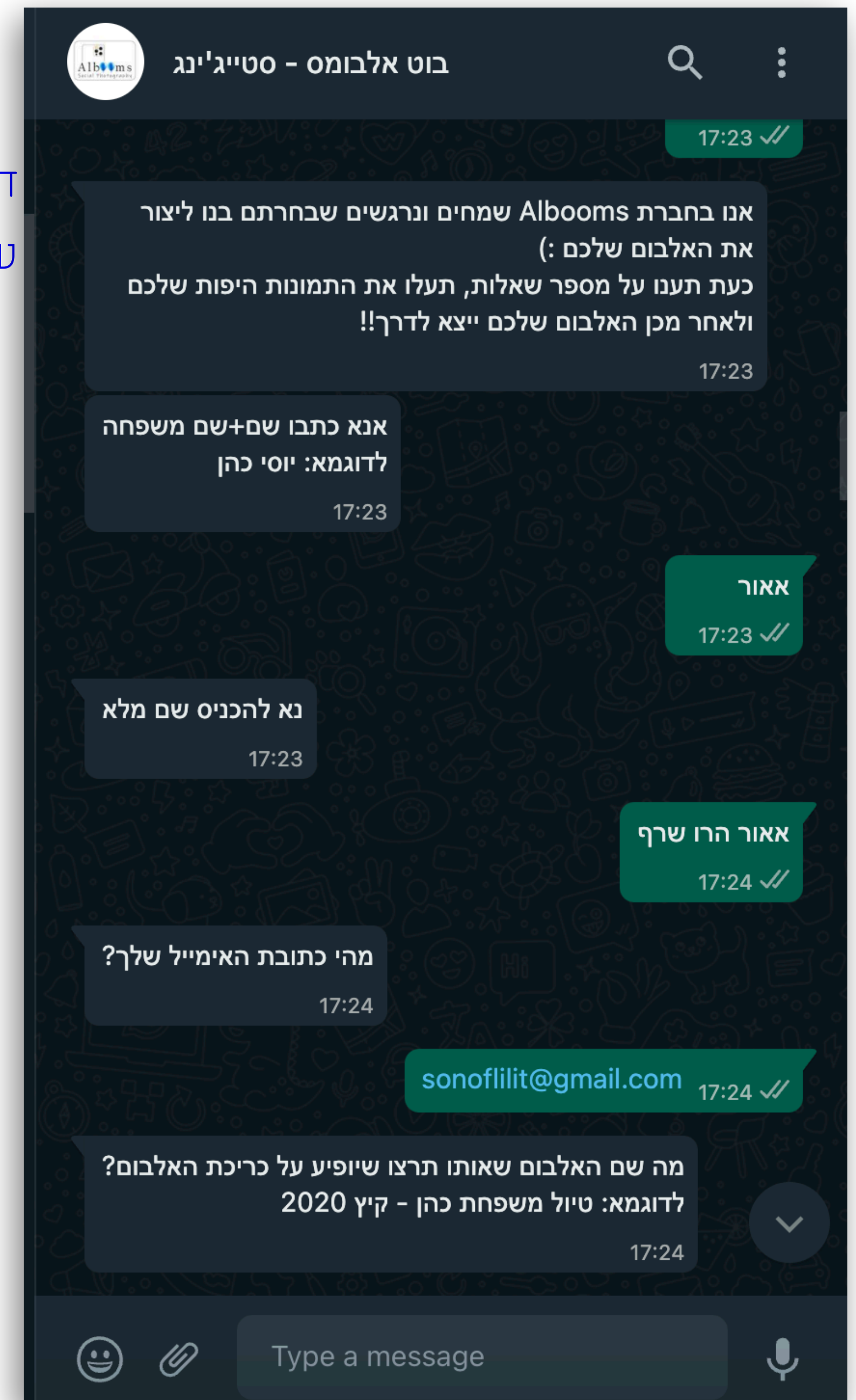
- אאור

- B: נא להכניס שם מלא

- אאור הרו שרף

- B: מה כתובת האימייל שלך?

# נוצר מתוך שיחה קל לערוך מתעדכן בכל הרצה



- להקמת אירוע

- B: |-

חזרתם בנו להקים את האירוע הבא שלכם \*Albooms\* אנו בחברת  
עת תענו על מספר שאלות ולאחר מכן האירוע שלכם ייצא לדרך

- B: |-

אנא כתבו שם+שם משפחה

לדוגמא: יוסי כהן

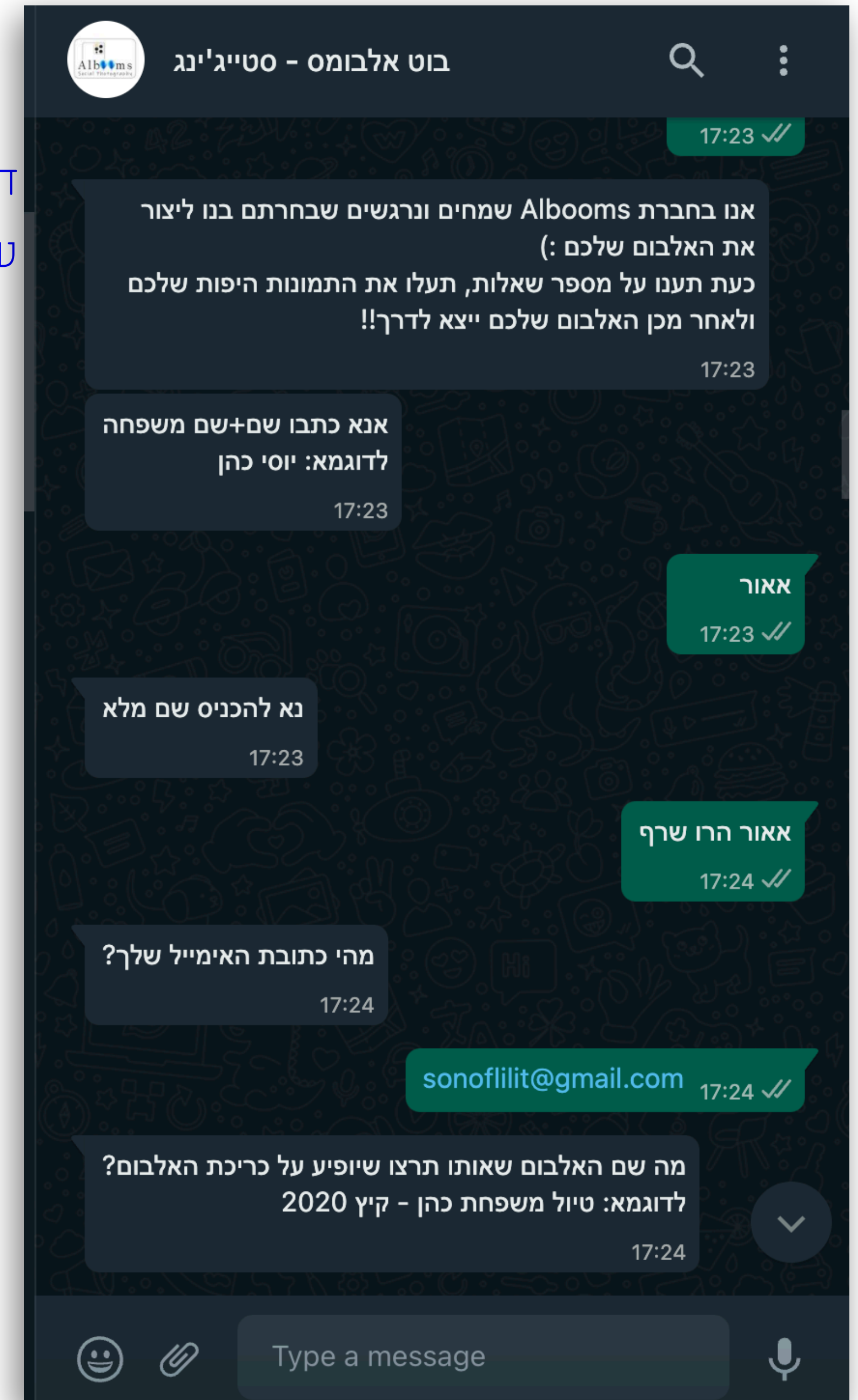
- אאור

- B: נא להכניס שם מלא

- אאור הרו שרף

- B: מה כתובת המייל שלך?

נוצר מתוך שיחה  
קל לערוך  
מתעדכן בכל הרצה  
diff של git עושה הכל





- להקמת אירוע

- B: |-

חזרתם בנו להקים את האירוע הבא שלכם \*Albooms\* אנו בחברת  
עת תענו על מספר שאלות ולאחר מכן האירוע שלכם ייצא לדרך

- B: |-

אנא כתבו שם+שם משפחה

לדוגמא: יוסי כהן

- אאור

- B: נא להכניס שם מלא

- אאור הרו שרף

- B: מה כתובת המייל שלך?

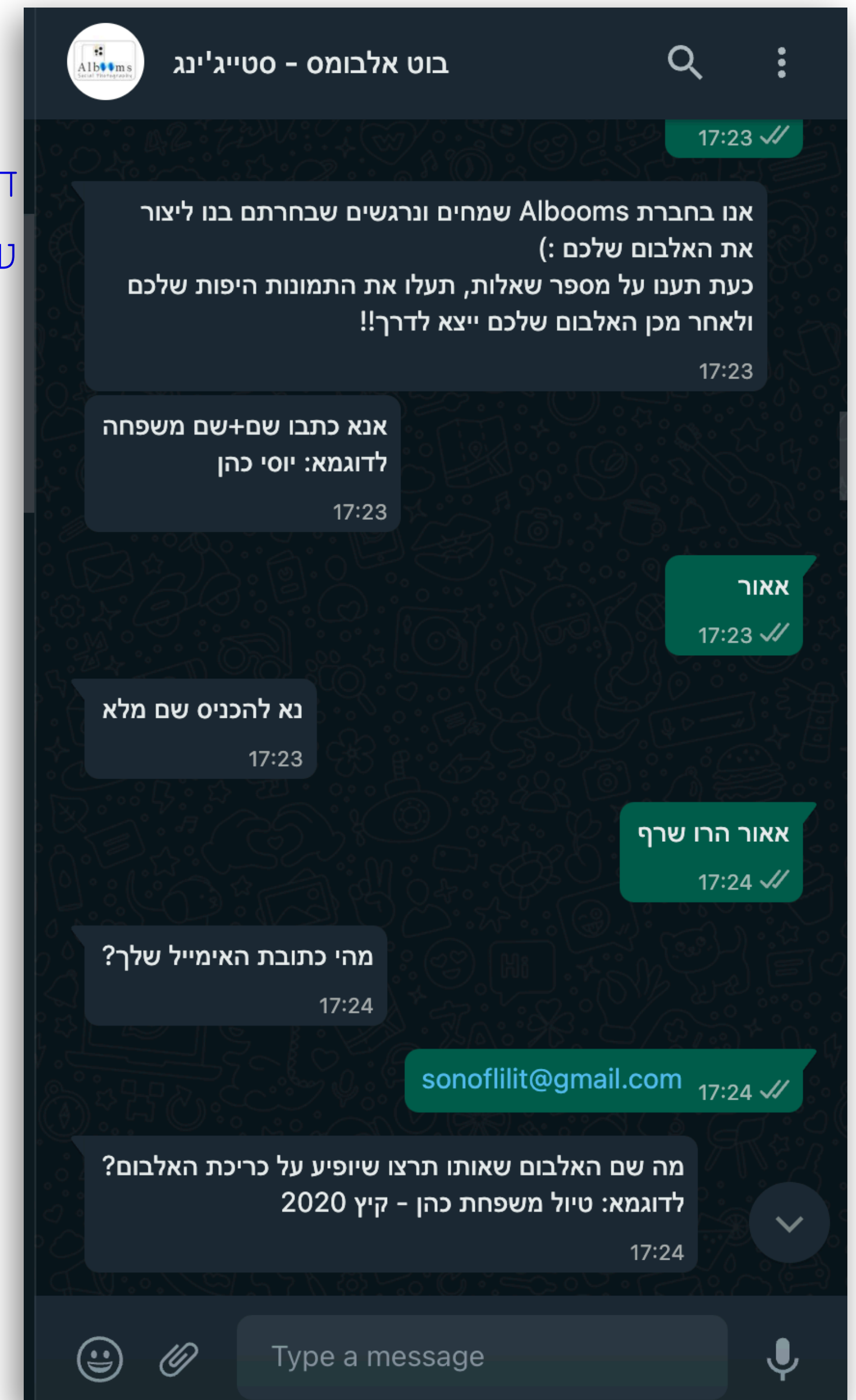
נוצר מתוך שיחה

קל לערוך

מתעדכן בכל הרצה

diff של git עושה הכל

מפעיל את כל המסלול האמיתי



- להקמת אירוע

- B: |-

חזרתם בנו להקים את האירוע הבא שלכם \*Albooms\* אנו בחברת  
עת תענו על מספר שאלות ולאחר מכן האירוע שלכם ייצא לדרך

- B: |-

אנא כתבו שם+שם משפחה

לדוגמא: יוסי כהן

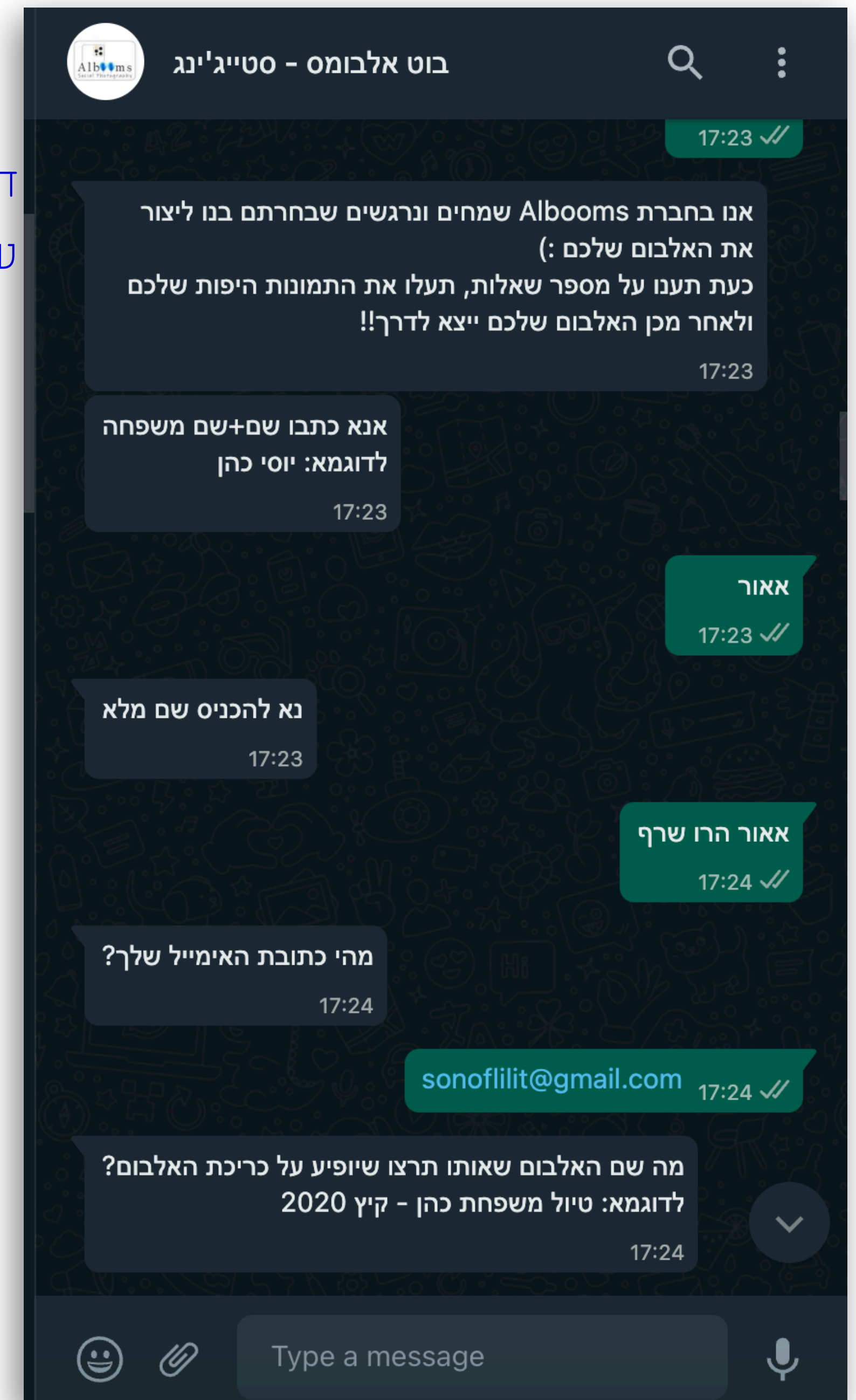
- אאור

- B: נא להכניס שם מלא

- אאור הרו שרף

- B: מה כתובת האימיל שלך?

אפשר להראות ללקוח



- להקמת אירוע

- B: |-

חזרתם בנו להקים את האירוע הבא שלכם \*Albooms\* אנו בחברת עת תענו על מספר שאלות ולאחר מכן האירוע שלכם ייצא לדרך

- B: |-

אנא כתבו שם+שם משפחה לדוגמא: יוסי כהן

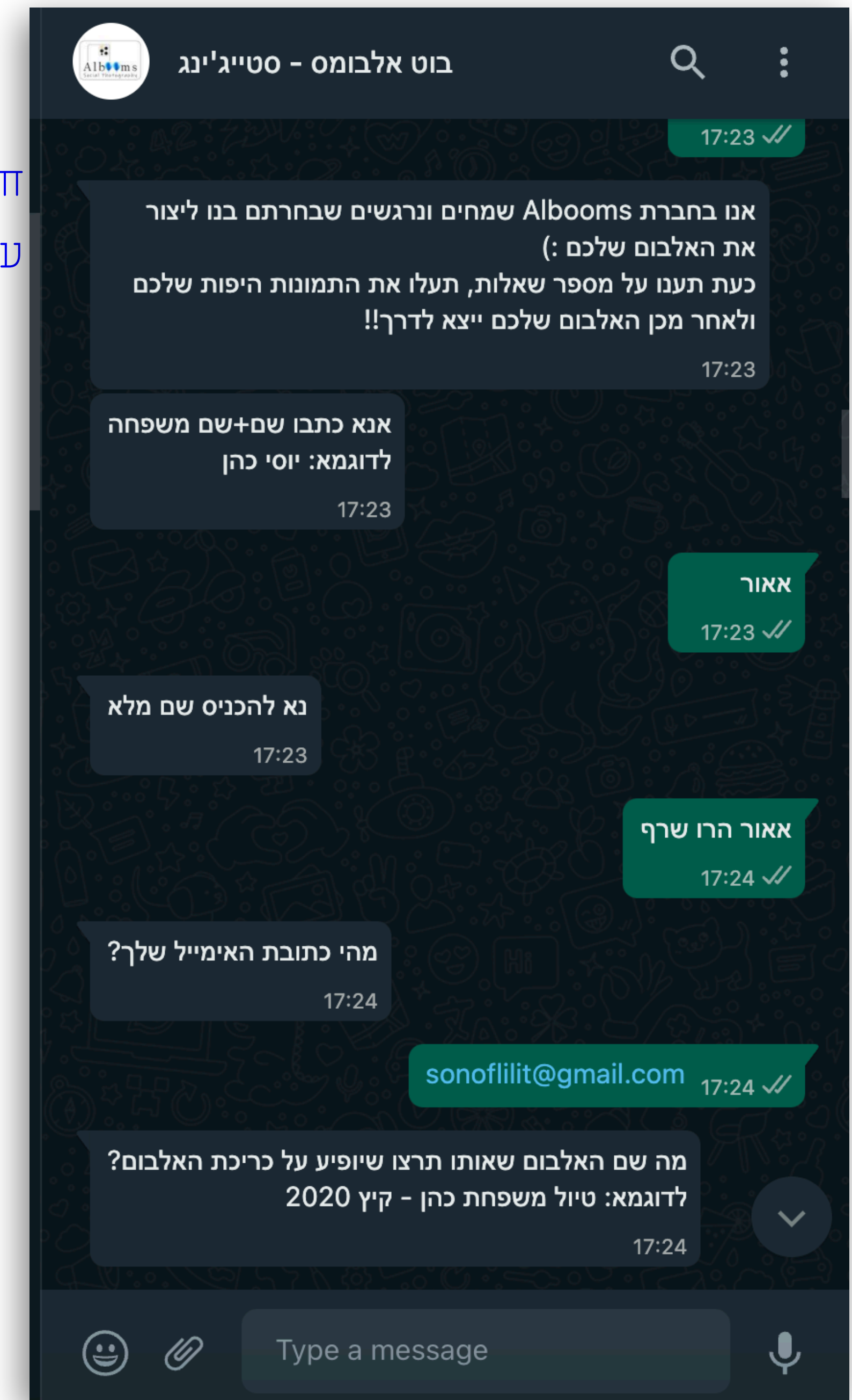
- אאור

- B: נא להכניס שם מלא

- אאור הרו שרף

- B: מה כתובת האימיל שלך?

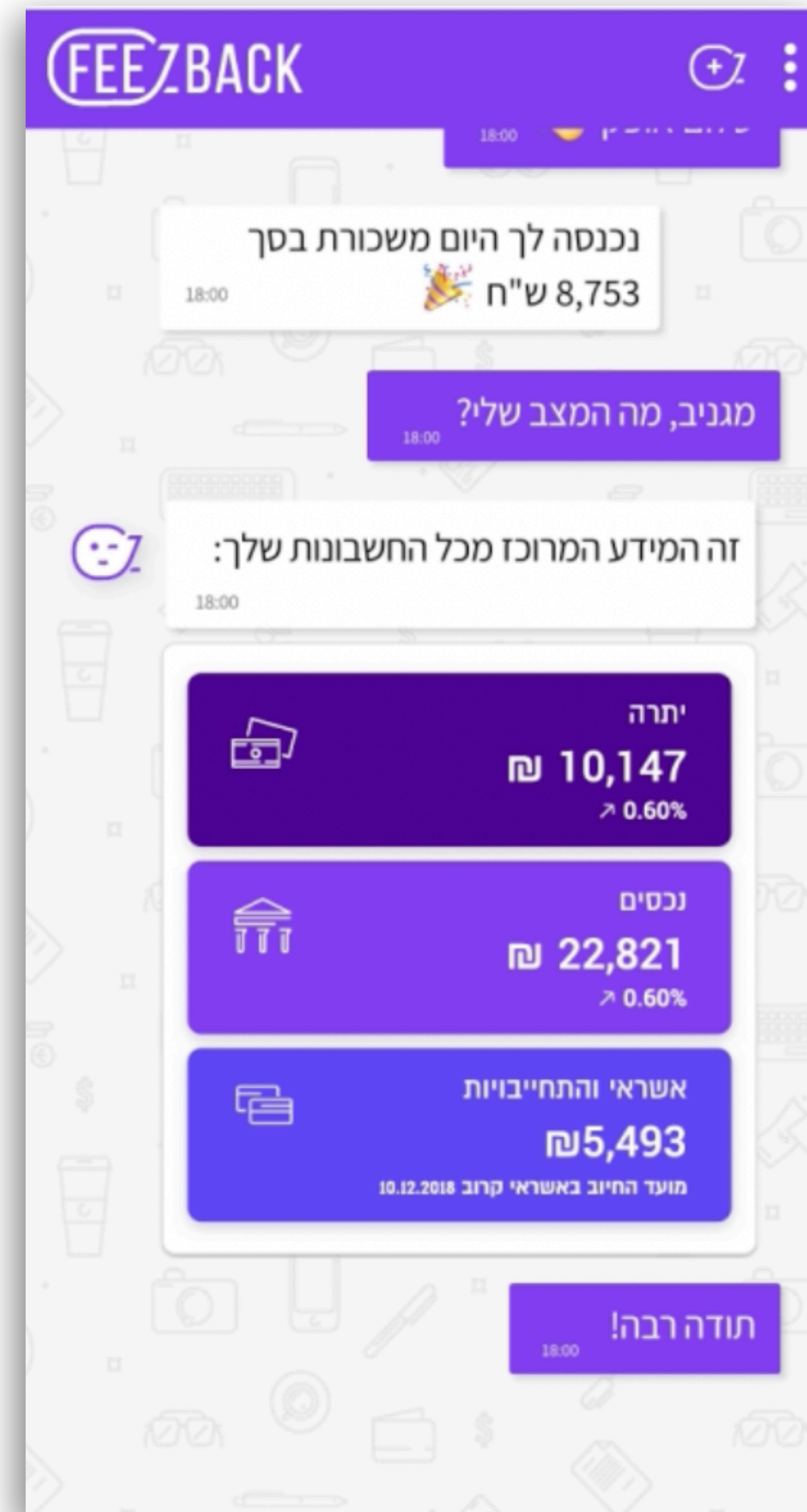
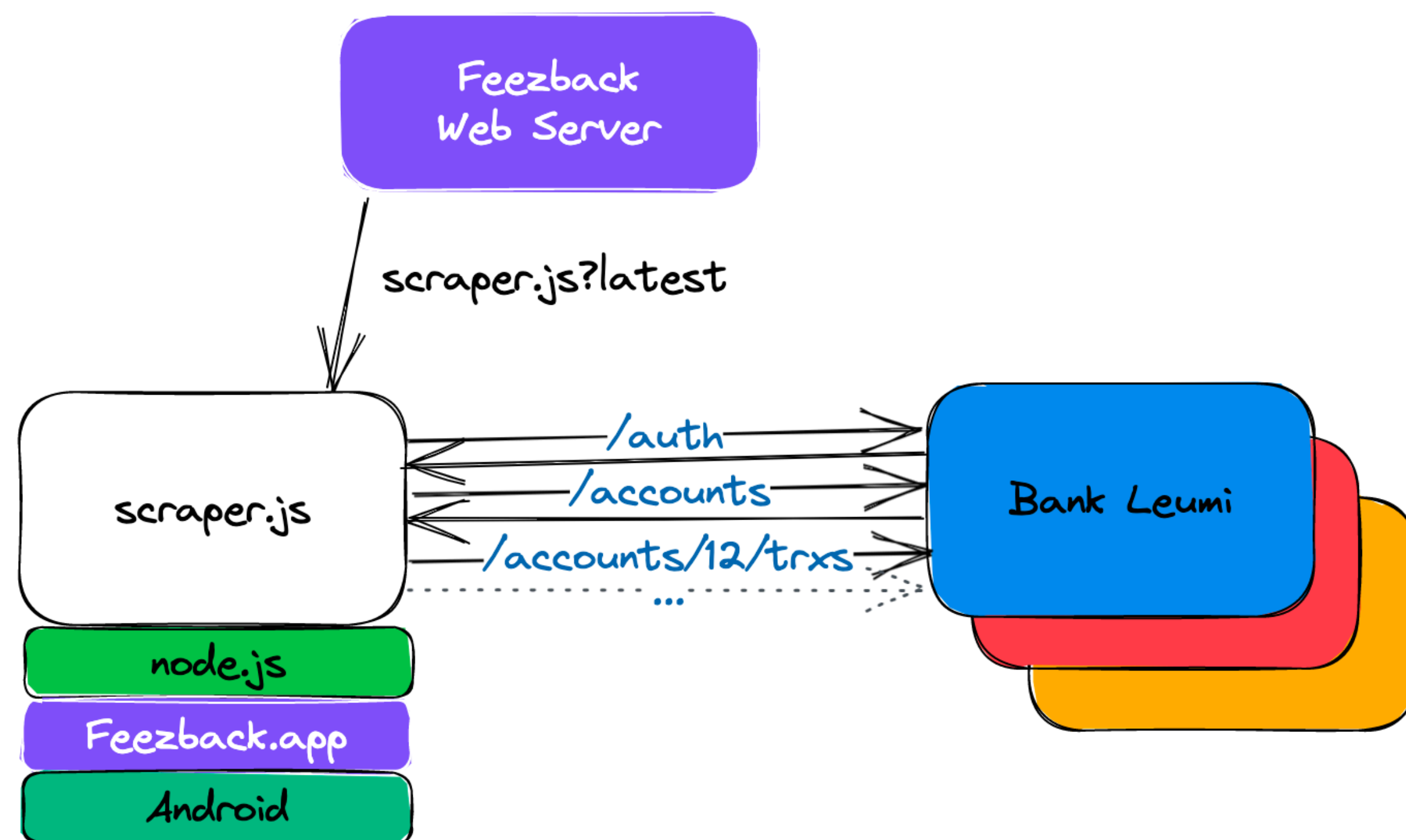
מאד YAML  
כה טכנולוגיה  
אפשר להראות ללקוח  
הרבה וואו  
התרשמות



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

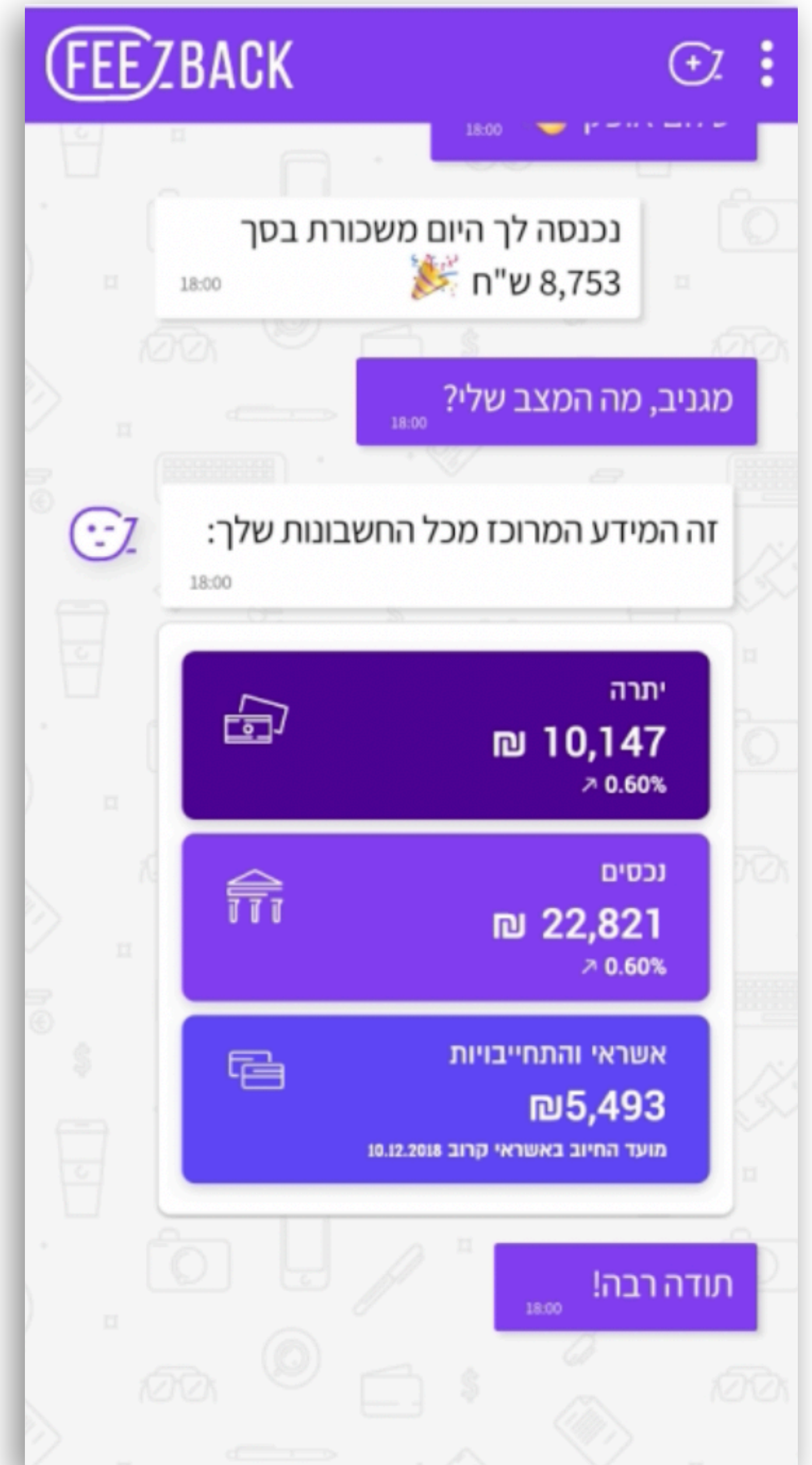
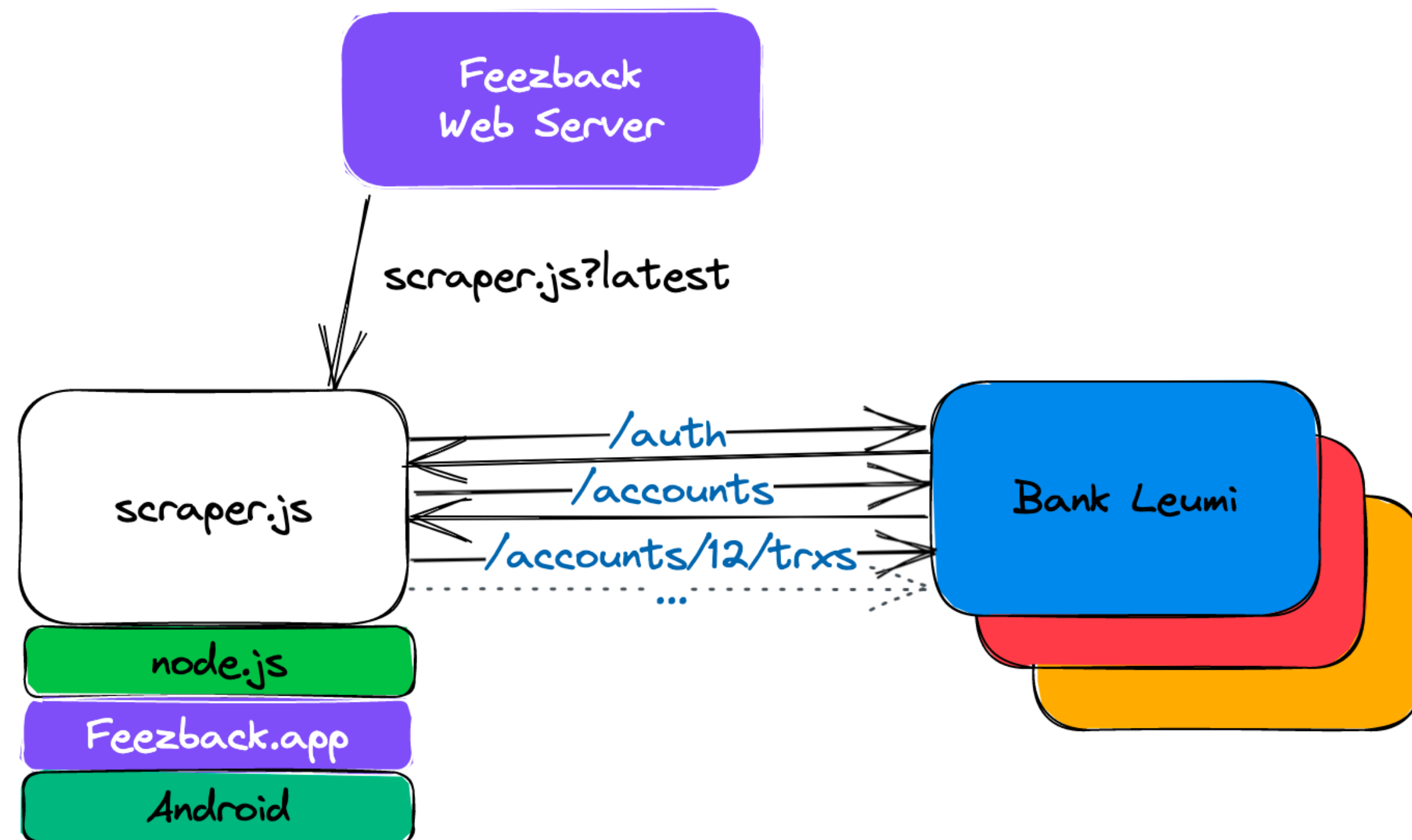
**pytest-gold, pytest-datarecorder,  
pytest-snapshot, snappiershot,  
jest, snap-shot-it...  
midas (mine, alpha)**

# סקרייפינג לבנקים



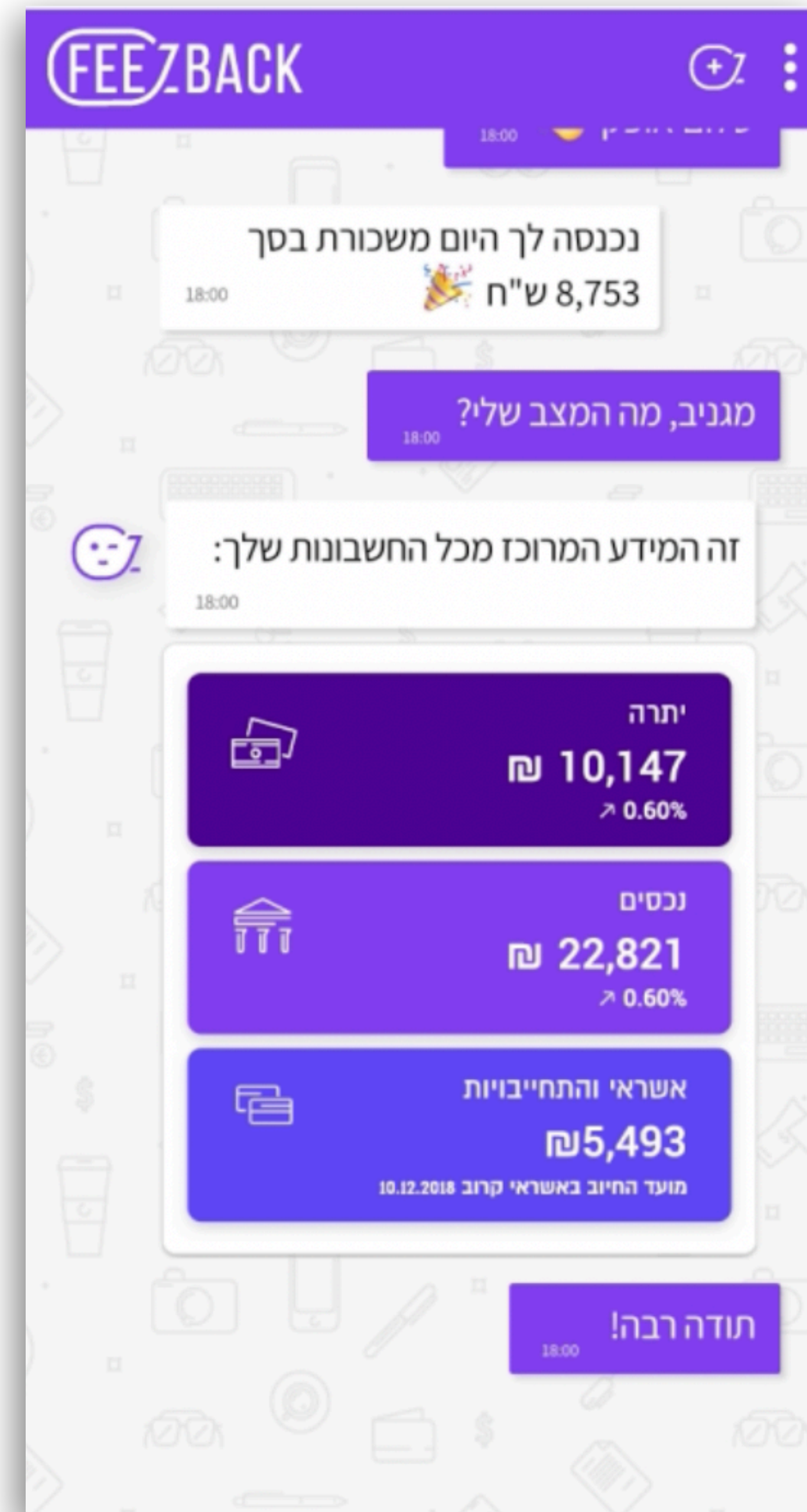
# סקרייפינג לבנקים

לפעמים הקוד משתנה  
לפעמים אתר הבנק משתנה



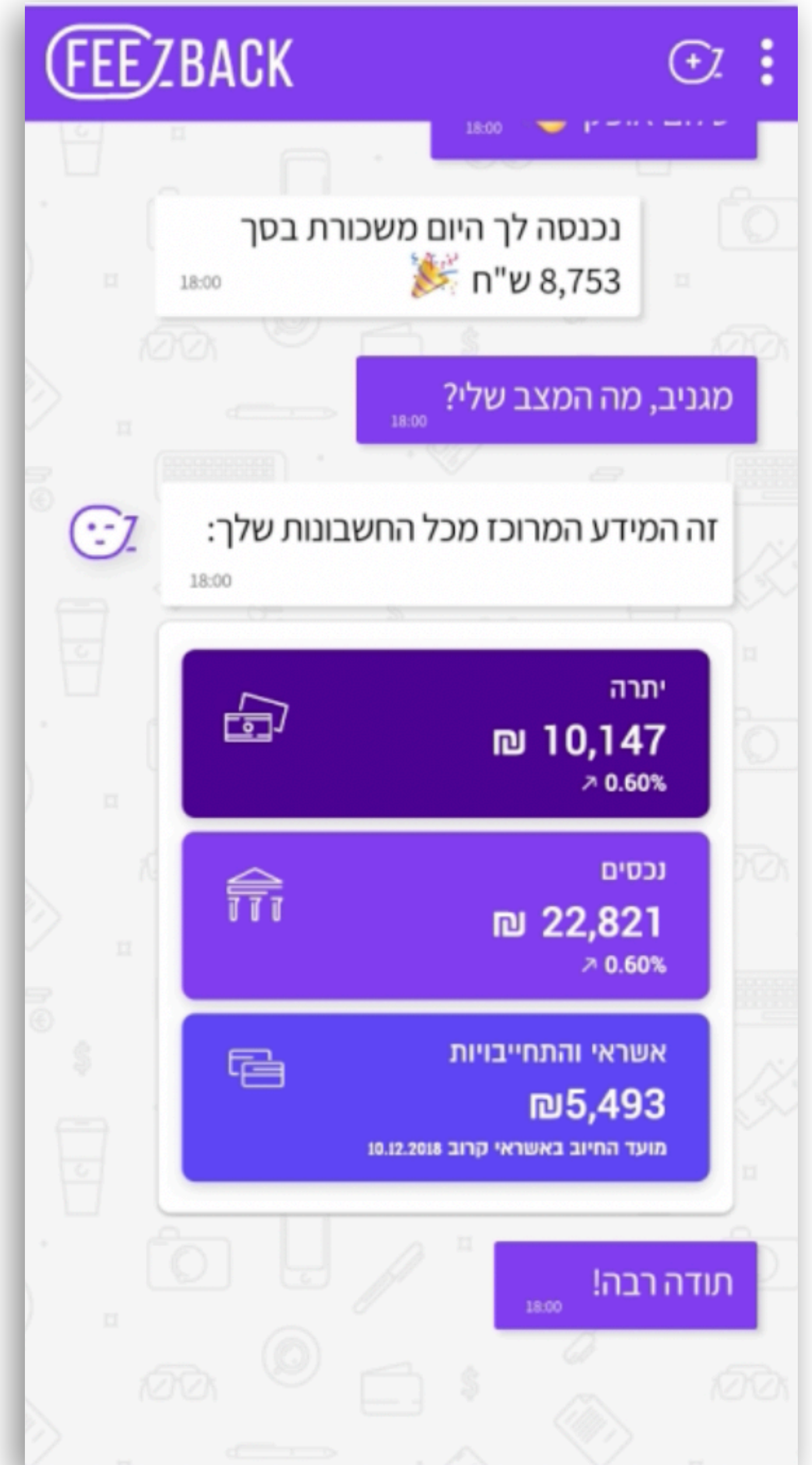
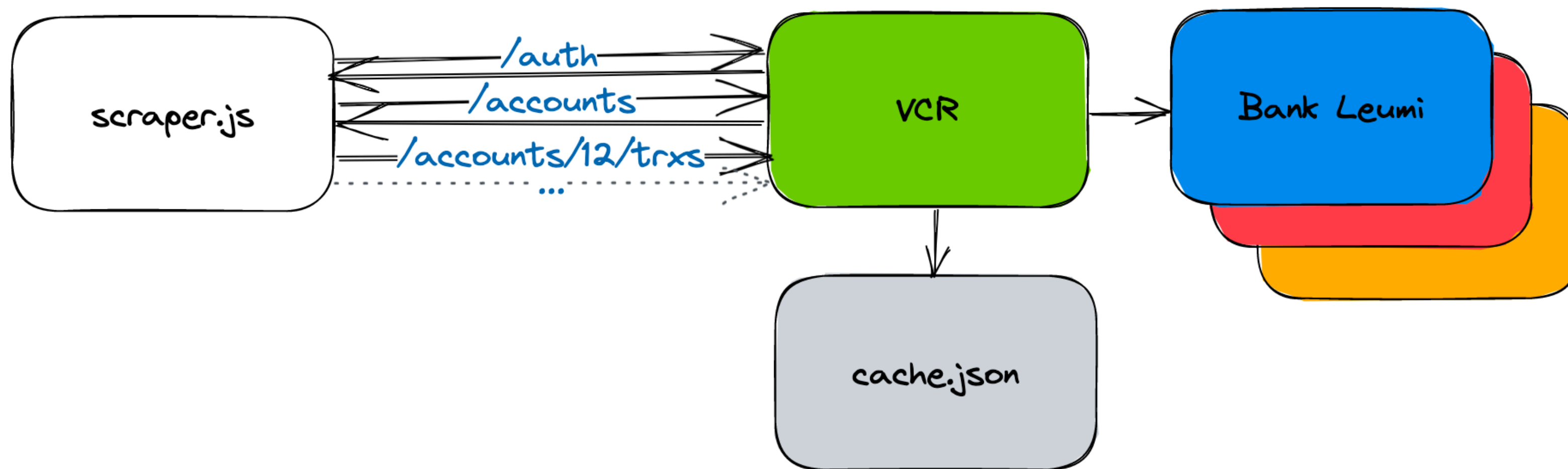
# סקרייפינג לבנקים

לפעמים הקוד משתנה  
לפעמים אתר הבנק משתנה



# VCR Test

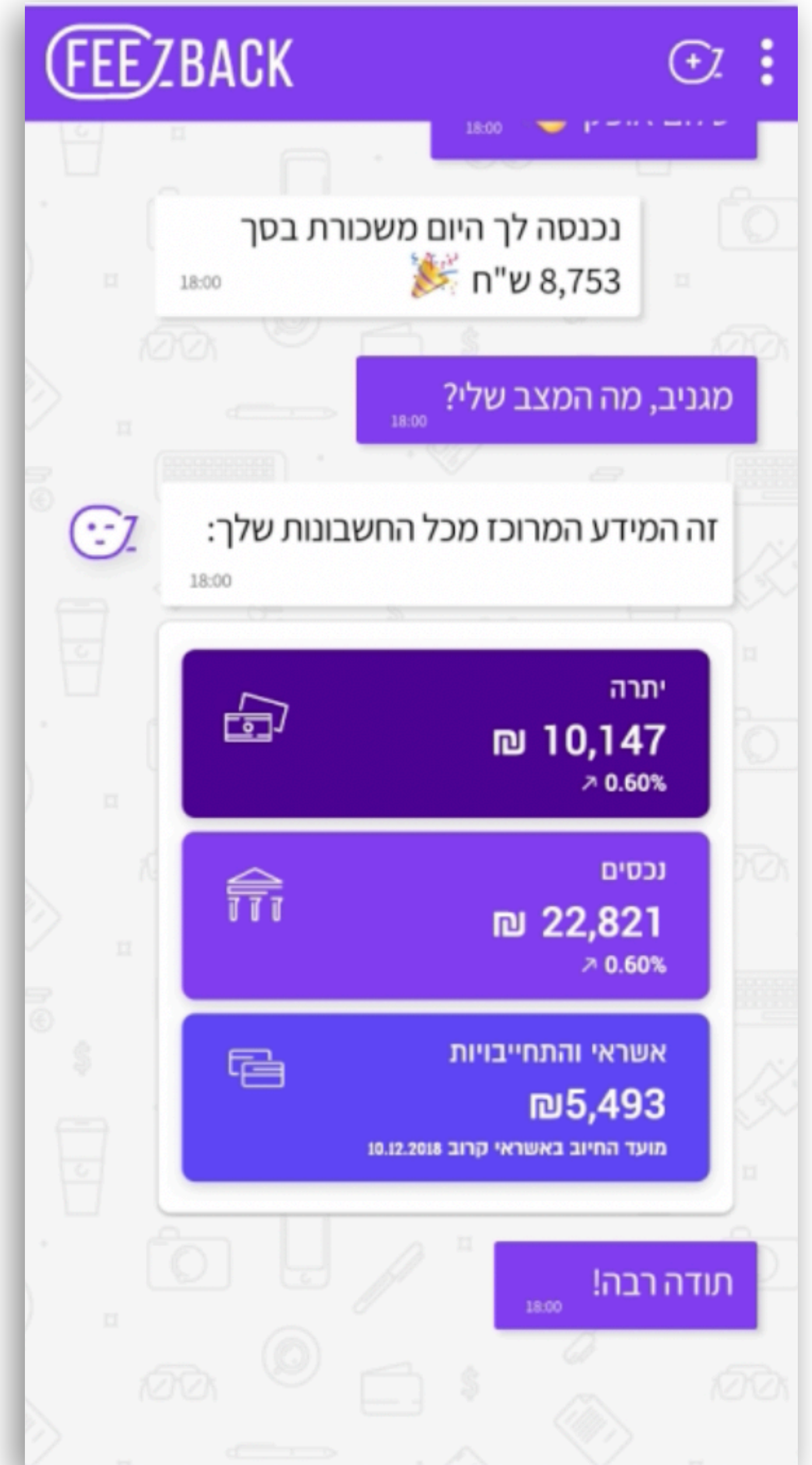
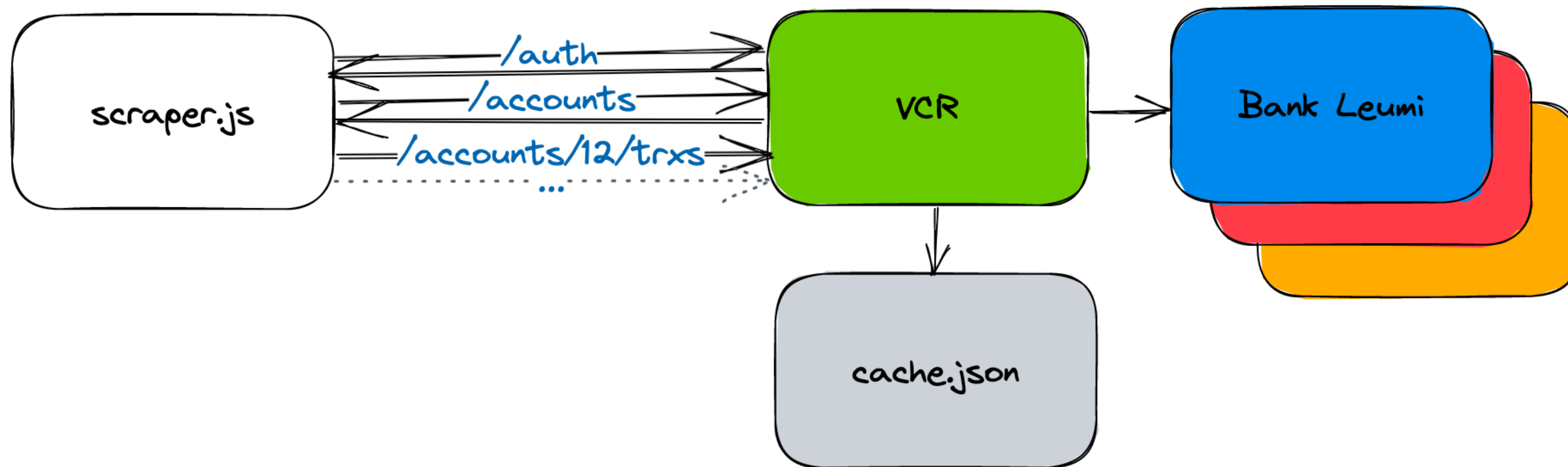
עושים proxy  
ל-http client





# VCR Test

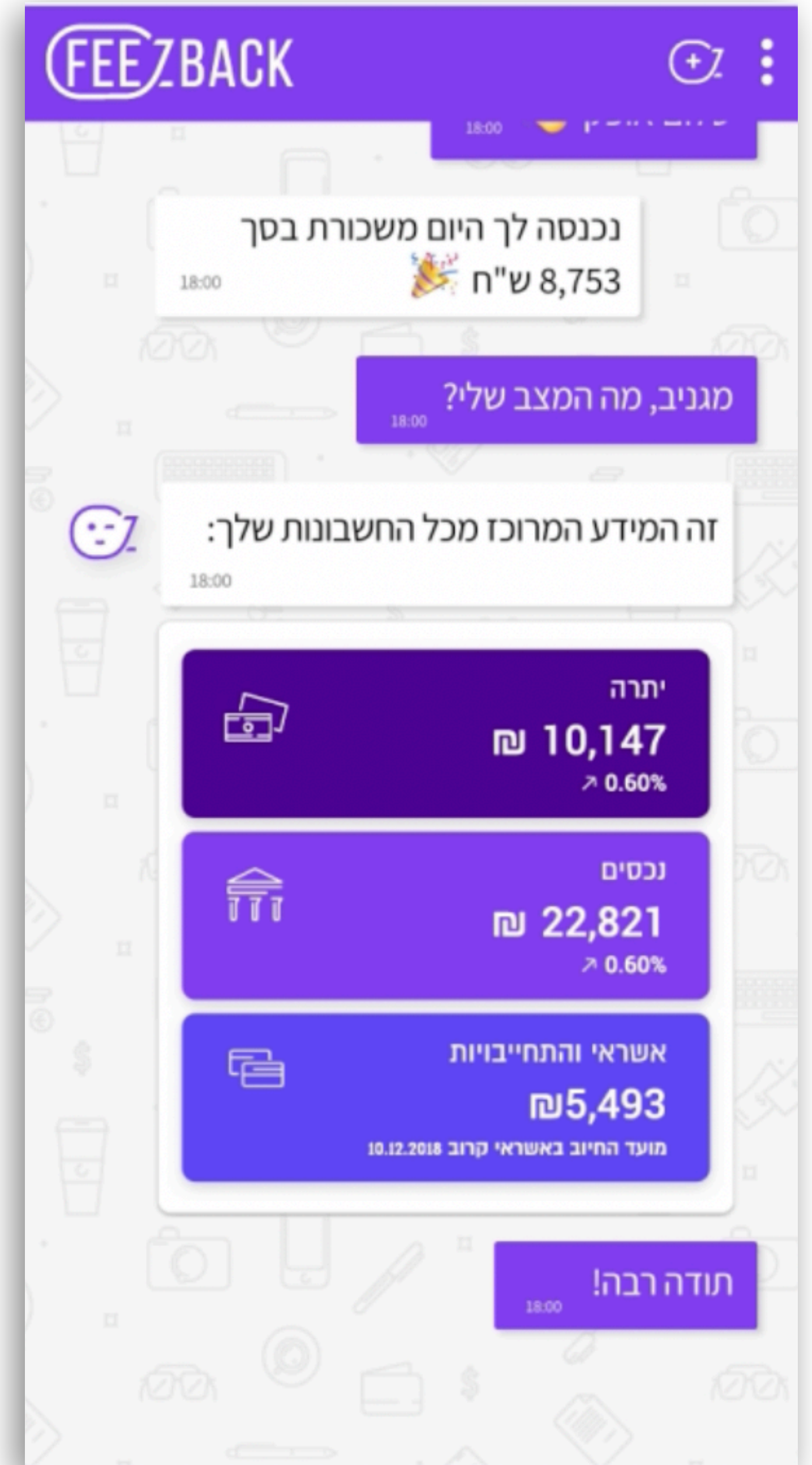
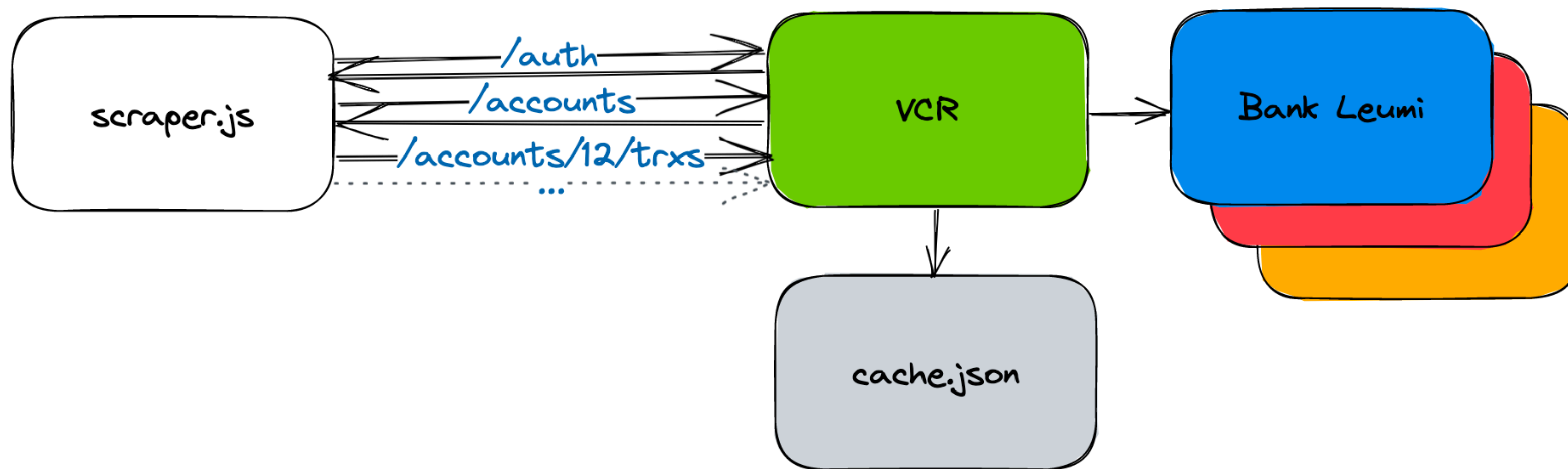
זה יוניט טסט?



# VCR Test

זה יוניט טסט?

זה integration test?

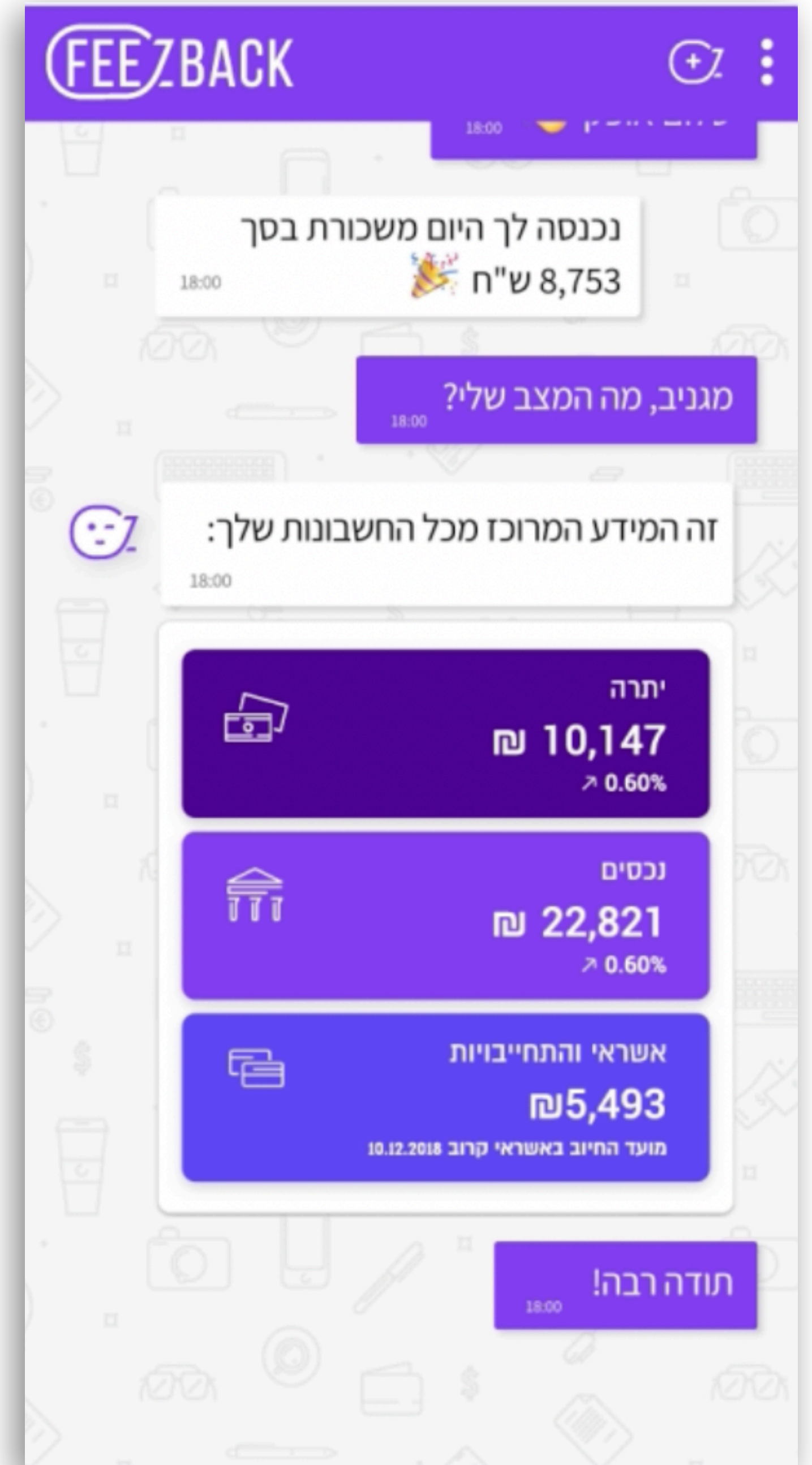
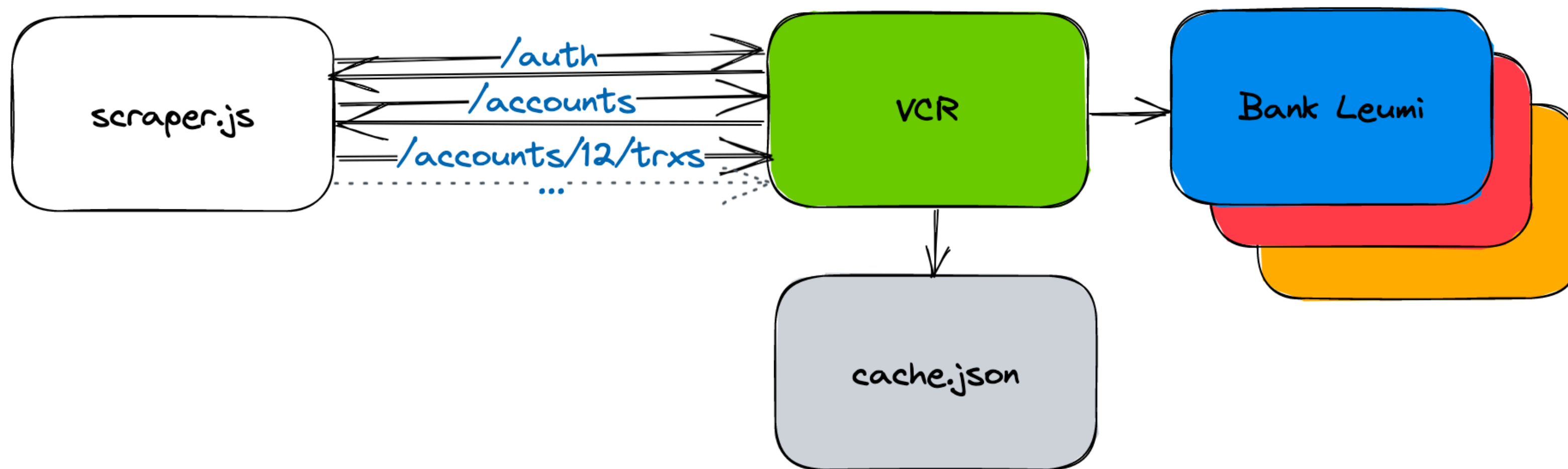


# VCR Test

זה יוניט טסט?

זה integration test?

לא, זה VCR test!



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

# RegEx, 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:
        ...
```

# RegEx, תחביר מודרני ל-RegEx

```
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:
```

```
    ...
```

מיתרגם ל-regex רגיל, רץ באותו מנוע

מימוש בפייתון

מימוש ב-rust ששמיש מ-js

איך הייתם

בודקים אותו?

# RegEx, תחביר מודרני ל-RegEx

```
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:
```

```
    ...
```

מיתרגם ל-regex רגיל, רץ באותו מנוע

מימוש בפייתון

מימוש ב-rust ששמיש מ-js

בואו נאפשר שימוש במימוש ב-rust מפייתון!

# RegEx, תחביר מודרני ל-RegEx

```
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:
```

```
    ...
```

מיתרגם ל-regex רגיל, רץ באותו מנוע

מימוש בפייתון

מימוש ב-rust ששמיש מ-js

בואו נאפשר שימוש במימוש ב-rust מפייתון!

אפשר לכתוב הרבה ביטויים ולעשות gold testing!



# RegEx, תחביר מודרני ל-RegEx

```
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:
```

```
    ...
```

מיתרגם ל-regex רגיל, רץ באותו מנוע

מימוש בפייתון

מימוש ב-rust ששמיש מ-js

בואו נאפשר שימוש במימוש ב-rust מפייתון!

אפשר לכתוב הרבה ביטויים ולעשות gold testing!

בואו נתרגם טסטים של פייתון!

# אגב, kleenexp שמיש בפייתון וב-vscode

A screenshot of a GitHub repository page for the 'kleenexp' project, showing a list of open issues. The issues are displayed in a dark-themed interface. Each issue entry includes a checkbox, a title, a description, and various labels. The issues are:

- 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**  
**pyo3**  
**CPython**

אז מה למדנו היום?

לפעמים אפשר לקבל המון  
בדיקות טובות בחינם

אז מה למדנו היום?

זה עשוי לעשות את ההבדל בין  
פרויקט מוצלח לכשלון מהדהד

לפעמים אפשר לקבל המון  
בדיקות טובות בחינם

אז מה למדנו היום?

זה עשוי לעשות את ההבדל בין  
פרויקט מוצלח לכשלון מהדהד

לפעמים אפשר לקבל המון  
בדיקות טובות בחינם

אז מה למדנו היום?

לפעמים כדאי להשקיע טיפה  
ב-DSL שטבעי לבטא בו  
בדיקות

זה עשוי לעשות את ההבדל בין  
פרויקט מוצלח לכשלון מהדהד

לפעמים אפשר לקבל המון  
בדיקות טובות בחינם

אז מה למדנו היום?

לפעמים כדאי להשקיע טיפה  
ב-DSL שטבעי לבטא בו  
בדיקות

git הוא כלי נהדר למימוש  
של golden tests תפורים  
לפרויקט



זה עשוי לעשות את ההבדל בין  
פרויקט מוצלח לכשלון מהדהד

לפעמים אפשר לקבל המון  
בדיקות טובות בחינם

## אז מה למדנו היום?

VCR tests הופכות הרצה  
של המוצר לבדיקת  
אינטגרציה

לפעמים כדאי להשקיע טיפה  
ב-DSL שטבעי לבטא בו  
בדיקות

git הוא כלי נהדר למימוש  
של golden tests תפורים  
לפרויקט

זה עשוי לעשות את ההבדל בין  
פרויקט מוצלח לכשלון מהדהד

לפעמים אפשר לקבל המון  
בדיקות טובות בחינם

## אז מה למדנו היום?

VCR tests הופכות הרצה  
של המוצר לבדיקת  
אינטגרציה

VCR tests הופכות בדיקת  
אינטגרציה לבדיקות יחידה  
מהירות

לפעמים כדאי להשקיע טיפה  
ב-DSL שטבעי לבטא בו  
בדיקות

git הוא כלי נהדר למימוש  
של golden tests תפורים  
לפרויקט

זה עשוי לעשות את ההבדל בין פרויקט מוצלח לכשלון מהדהד

לפעמים אפשר לקבל המון בדיקות טובות בחינם

## אז מה למדנו היום?

VCR tests הופכות הרצה של המוצר לבדיקת אינטגרציה

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

לפעמים כדאי להשקיע טיפה ב-DSL שטבעי לבטא בו בדיקות

VCR tests הופכות בדיקת אינטגרציה לבדיקות יחידה מהירות

איך הייתם בודקים אותו?

git הוא כלי נהדר למימוש של golden tests תפורים לפרויקט

תמיד אפשר יותר טוב

# אני אאור

אני מייעץ לסטארטאפים על טכנולוגיה ואסטרטגיה.  
אני עושה מנטורינג ו-design reviews.  
אני פותר בעיות טכנולוגיות קשות.

## שאלות?

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

[bit.ly/aurs-dojو](https://bit.ly/aurs-dojو)