REST-API

API מה זה?

API ראשי תיבות של Application Programming Interface .
API היא דרך יפה לומר: תקשורת בין שני רכיבי תוכנה (לאו דווקא מאותה השפה) שאינם מחוברים בניהם בקשר ישיר.
למעשה כל הספריות של פייתון שראינו עד עכשיו הן סוג של API- אנחנו מבקשים משהו בפורמט מוגדר (פונקציה או מחלקה), ומקבלים ערך מהספרייה בהתאם לבקשה. לא אכפת לנו ממש מה קורה מאחורי הקלעים, אכפת לנו רק מהתוצר הסופי. אנלוגיה טובה ל-API היא מסעדה- אנחנו כלקוחות מזמינים משהו מהמטבח. לא אכפת לנו התהליך שקורה מאחורי הקלעים, רק מה שנקבל בסוף. המלצר, זה שמקשר בנינו לטבח, הוא ה-API .
את ה-API ניתן למצוא כמעט בכל מקום בעולם התוכנה, אתרי אינטרנט למשל מבוססים תקשורת REST-API או RESTFULL-API (שם נוסף לאותו דבר), הלקוח מבקש מהשרת תוכן מסוים והשרת מחזיר לו את התוכן, לפעמים ע"י שינוי ה-Frontend (קבלה של קבצי html), ולפעמים ע"י שליחה של קבצי json, וזה לא משנה שקוד צד השרת וקוד צד הלקוח כתובים בשפה אחרת.

מה זה בכלל REST?

REST ראשי תיבות של Representational State Transfer, הוא סגנון בארכיטקטורת קוד לעיצוב אפליקציות רשת.
הוא מבוסס על פרוטוקול stateless לקוח ושרת, שלרוב הוא פרוטוקול http.
מה שמעולה ב-REST שהוא לא מוגבל לפורמט ספציפי למשל html , הוא גם יכול להשתמש בקבצי json או xml ועוד ,מה שמאפשר שימוש בכמעט כל שפת תכנות קיימת תחת הפרטוקול, שכן רוב השפות מאפשרות לבצע בקשות http באיזושהי דרך ולרובן יש ספרייה שיודעת לטפל בקבצי json או xml .
לפרוטוקול http יש כמה מתודות:

- GET- הנפוצה ביותר. מקבלת נתונים ממקור מוגדר. למעשה לקוח מבצע בקשת GET כמעט כל הזמן למשל מכניסה לכתובת url של אתר.
- POST- שליחת נתונים לעיבוד במקור מוגדר. כל פעם שאנחנו ממלים איזשהו טופס אינטרנטי, רישום לאתר למשל, אנחנו לרוב משתמשים במתודת POST. ניתן גם להשתמש במתודת GET אבל היא לא מאובטחת כמו POST.
- PUT- עדכון של נתונים ממקור מוגדר. דומה ל-POST רק שהוא לא 'מכפיל' ערכים אם עדכנו, כלומר אם שלחנו ערך שכבר קיים הוא לא עלול להכפיל אותו.
- DELETE- מחיקה של נתונים ממקור מוגדר.
- HEAD- כמעט ולא משתמשים בה. דומה ל-GET רק בלי לקבל איזשהו תוכן בחזרה.
- PATCH- עדכון של חלק מהמקור המוגדר.

דוגמא ל-RESTFULL API: האתר https://exchangeratesapi.io/ מספק API של שערי מטבע חוץ של היורו.
הוא מאפשר לנו לראות ערכים של כמה סוגי מטבעות בעולם ולהשוות אותם יחסית לערך של מטבע מסוים. הוא גם מאפשר לראות את ערכי המטבעות לפי תאריכים (כברירת מחדל הוא משווה את המטבעות לפי ערך היורו).
השימוש ב-API הוא מאוד פשוט מכניסים לו את הריישא של כתובת ה-URL שלו, ומוספים בסוף את המשתנים של השאילתות.
למשל כדי לראות את ערכם של כל המטבעות יחסית ליורו נשתמש בכתובת(ניכנס לכתובת):

```
https://api.exchangeratesapi.io/latest
```

הוא אמור להפעיל מתודת GET של http שמחזירה לנו json עם ערכי המטבעות. אבל כרגע הוא דורש "מפתח גישה" – access key. הדבר נועד לאפשר לכל משתמש מספר מוגבל של גישות לממשק, כדי שלא יעמיסו יותר מדי על השרת. כדי להשיג מפתח-גישה, יש להיכנס לדף הראשי https://exchangeratesapi.io/ וללחוץ על GET

API KET. למלא את הפרטים ולרשום את המפתח שמקבלים. אחר-כך אפשר להכניס את המפתח כפרמטר ל-URL באופן הבא:

http://api.exchangeratesapi.io/latest?access_key=.....

ואנחנו אמורים לראות משהו כזה (שימו לב שהכל ביחס ליורו):

```
{
  "rates": {
    "CAD": 1.499,
    "HKD": 9.2325,
    "ISK": 151.75,
    "PHP": 57.724,
     ...
},
  "base": "EUR",
  "date": "2021-03-09"
}
```

אפשר להוסיף פרמטרים נוספים. לדוגמה, אם לא רוצים את כל המטבעות אלא רק שקל ודולר, אפשר להעביר פרמטר symbols=ILS,USD.

אפשר גם לפתוח חשבון בתשלום, ואז נפתחות אפשרויות נוספות, לדוגמה, אפשר לשנות את הבסיס להשוואה מיורו למטבע אחר. לדוגמה, אם נרצה לבחון את הערכים לפי מטבע מסויים, למשל דולר, ונרצה לראות רק כמה השקל שווה יחסית אליו, נוכל להשתמש בפרמטרים base לציון לפי מי ההשוואה ו-'symbols' כדי להגדיר אילו מטבעות אנחנו רוצים לראות. לפני שנכניס את המשתנים נשים סימן שאלה, ובין כל פרמטר שאנחנו מוסיפים (למעט הראשון) נוסיף '&'.
בדוגמא שהצנו לעיל נרצה לראות השוואה רק של השקל לפי הדולר:

```
https://api.exchangeratesapi.io/latest?base=USD&symbols=ILS
```

וה-json שאנחנו אמורים לקבל בחזרה ,נכון לכתיבת שורות אלה (אם שילמנו על הממשק המתקדם), הוא:

```
{
  "rates": {
    "ILS": 3.3295779385
  },
  "base": "USD",
  "date": "2021-03-09"
}
```

RESTFULL-API ופייתון:

לפייתון יש מודול שיכול לעזור לנו כדי להשתמש ב-RESTFULL-API , והוא נקרא requests.
requests היא ספרייה חיצונית שמאפשרת לשלוח בקשות http ולקבל ערכים חזרה.
בשביל להשתמש בספרייה נצטרך להתקין אותה עם pip :

```
pip install requests
```

הבקשה שאנחנו מבצעים היא בקשת GET, לכן נשתמש בפונקציה GET של המודול עם הכתובת שאליה אנחנו מנסים לגשת. משום שה-URL מחולק לשני חלקים- הריישא שנשאר קבוע, והפרמטרים שמשתנים בהתאם לסוג הבקשה. נחלק גם אנחנו את הבקשה לפי חלקים- מחרוזת שמכילה את הריישא ומילון שמכיל את הפרמטרים שנשתמש בהם.
ניצור משתנה שיקבל את הערך חזרה שנוכל לראות אותו אח"כ:

```
prefix = 'https://api.exchangeratesapi.io/latest?'
params = {'base':'USD','symbols': 'ILS'}
response = requests.get(prefix, params= params)
```

המשתנה שקיבלנו חזרה הוא מאובייקט מטיפוס Response של המודול request והוא מכיל בתוכו כמה משתנים בניהם status_code שאומר אם הבקשה התקבלה (200) או לא (404), headers שמחזיר מילון עם כל ה-headers של בקשת ה-http, text- שמחזיר מחרוזת עם תוכן הבקשה(הקובץ json כמחרוזת), וכו'.
מה שחשוב לנו בעיקר זאת הפונקציה .json() הפונקציה משתמשת במודול json של ספרייה הסטנדרטית של פייתון והופכת את תוכן הבקשה למילון עם הערכים של קובץ ה-json שחזר:

```
res_dict = response.json()
print(res_dict['date'])
print(res_dict['base'])
print(res_dict['rates']['ILS'])
```

```
2021-03-09
USD
3.3295779385
```

ועכשיו שיש לנו את זה נוכל לבצע כמה פעולות יותר מורכבות, למשל לראות כמה שווים 100 דולרים בשקלים, או בכמה אחוזים ירד הדולר מאתמול לעומת היום:

```
il_shekel = float(res_dict['rates']['ILS'])
amount_us_dollars = input('how much dollars do you have? ')
print(f'{amount_us_dollars} ILS relative to USD  is {il_shekel*int(amount_us_dollars)}')
```

למה זה כל-כך שימושי לנו?

REST-API הוא כלי שימושי מאוד לשילוב של כמה רכיבי תוכנה יחד בלי תלות בשפה שבהם הם נכתבו או בפלטפורמה של הקוד, ובלבד שהרכיבים יכולים להתחבר לרשת. תחשבו שנוכל לשלב את קוד הפייתון שכתבנו בפלטפורמה חיצונית כמו הטלפון שלנו מבלי לחבר אותם עם מלא חבילות מסובכות ותוכנות. כל מה שנחוץ לנו שהקוד יוכל לשלוח חישובים בצורה של json, שיהיה מותקן על שרת אינטרנטי, ושהפלטפורמה שאנחנו רוצים לחבר לשרת תוכל להתחבר לרשת.

בשיעורים הבאים נראה כיצד להשתמש ב-flask, מה שיאפשר לנו ליצור REST-APIs משלנו.

עוד על הספרייה requests ניתן למצוא כאן.