

## מעבדה 3 – Reactive Evolution

---

### מטרה

במעבדה זו נבין את האבולוציה של תכנות Reactive וכיצד הגענו ל- Reactive Programming באמצעות Spring Framework

### תכולה

- Spring Reactive Web
- Lombok
- Completable Future

### נקודת התחלה

את קוד המקור תוכלו למצוא בלינק: <https://github.com/KobiShashs/lab-3-Reactive-Evolution>

### מה במעבדה?

מחלקת User המתארת לקוח (id, name)

מחלקת Current המתארת חשבון עו"ש (balance, currency)

מחלקת Stock המתארת נייר ערך (name, amount, value, currency)

מחלקת Status המתארת תשובה סופית חוזרת (id, name, balance)

### דרישה

עליכם לחשב עבור משה כמה הוא שווה?

אציין כי למשה בעו"ש 2500 דולר, בנוסף משה מחזיק ב-2 ניירות ערך:

- נייר ערך GOOGL – כמות 100, מחיר למניה – 97.46 דולר
- נייר ערך MSFT – כמות 100, מחיר למניה – 247.49 דולר

### חישוב

$$2500 + (100 \times 97.46) + (100 \times 247.49) = 2500 + 9746 + 24,749 = 36995$$

## ממשו

ממשו את שלושת ה-endpoints הבאים:

- GET - <http://localhost:8080/api/status/option1>
- GET - <http://localhost:8080/api/status/option2>
- GET - <http://localhost:8080/api/status/option3>

עבורן תחזור התשובה הבאה:

```
{"id":123,"name":"Moshe","balance":36995.0}
```

## בקשה ראשונה

ממשו פתרון באופן סינכרוני

- א. משיכת פרטי הלקוח
- ב. משיכת פרטי העו"ש
- ג. משיכת פרטי ניירות ערך

חישוב פרטי העו"ש והחזרת אובייקט מתאים

## בקשה שניה

ממשו פתרון מקבילי ככל הניתן באמצעות CompletableFuture API

חישוב פרטי העו"ש והחזרת אובייקט מתאים

## בקשה שלישית

ממשו פתרון non-blocking באמצעות Mono.zip – קראו על כך באינטרנט!

חישוב פרטי העו"ש והחזרת אובייקט מתאים

# בהצלחה!