

סקירה כללית 🎯

במערכת האוטומציה שבנינו באמצעות **Playwright** ו-TypeScript, יישמנו **תשתית גנרית לפעולות על עמוד תוצאות חיפוש** באתר BuyMe. המטרה הייתה לבצע חיפוש, לאתר תוצאות בצורה דינמית, לבחור תוצאה (באופן רנדומלי או לפי אינדקס), ולאחר מכן לפעול בהתאם לסוג התוצאה — בין אם מדובר בכרטיס עם סכום קבוע או שדה קלט.

✓ תכנות מונחה עצמים (OOP) – פעמיים

השתמשנו במבנה OOP **שכפול רמות** על מנת להפריד בין הלוגיקה הגנרית של עמוד תוצאות, לבין האסטרטגיה לבחירת פריט ספציפי:

1. מחלקת `ab` – `SearchResultsPage`

- מחלקה אבסטרקטית, המכילה את כל הלוגיקה המשותפת:
 - המתנה לטעינת הדף
 - איתור רשימת פריטים (`li.bm-gift-card-link`)
 - פעולה על פריט נבחר בהתאם לסוגו (קבוע או עם שדה קלט)
- מכילה מתודת `abstract selectItem(count)` — אסטרטגיה שמחייבת מחלקת בן לממש בחירה בפריט יחיד.

2. מחלקות בן – `RandomItemResultsPage` ו-`IndexedItemResultsPage`

- מיישמות את החוזה מהאב:
 - `RandomItemResultsPage` בוחר פריט אקראי מתוך התוצאות.
 - `IndexedItemResultsPage` בוחר פריט לפי אינדקס שנשלח מראש.
 - כך מימשנו את עיקרון **Open/Closed** של SOLID — המחלקה האב פתוחה להרחבה אך סגורה לשינוי.
-

🎁 שימוש ב-Decorator (הרכבה דינמית)

בנוסף ל-OOP, עשינו שימוש בגישה שמזכירה **דקורטור פונקציונלי** – לא דרך תחביר `@Decorator`, אלא על ידי הרכבה דינמית של מחלקות.

```
const flow = ExtendWithResultsActions(BuymeSimpleSearch,
RandomItemResultsPage)(page, 'שוקולד');
```

כאן, אנו יוצרים **pipeline התנהגותי**, בו מחלקת `BuymeSimpleSearch` מבצעת את פעולת החיפוש, ומחלקת `RandomItemResultsPage` דואגת לבחור וללחוץ על פריט מתאים לאחר שהחיפוש הושלם.

`ExtendWithResultsActions` הוא **factory function** שמקבל שתי מחלקות – מחלקת חיפוש ומחלקת תוצאות – ומחזיר מחלקה מורחבת שמרכיבה את שתי ההתנהגויות. כך שמרנו על **אבחנה בין אחריות** (*Separation of Concerns*), וכל פעולה אחראית רק לחלק שלה בזרימה.

טכניקות מתקדמות 🔍

- שימוש ב-**Playwright Locators** עם תנאי `count() > 0` לזיהוי דינמי של שדה קלט (`input[placeholder="הכנס סכום"]`), דבר המאפשר טיפול בתצוגות משתנות.
- זיהוי לפי **סלקטורים יציבים**, לאחר ניתוח DOM מעמיק באתר, כדי להימנע משברים עתידיים.
- הרצת **טסטים חכמים** שמגיבים לשינויים בדינמיקה של האתר (לדוג' תוצאות עם סכום או שובר כספי).

יתרונות לגישה 🧠

- **מודולריות גבוהה** – ניתן להחליף בקלות את מנגנון החיפוש או הבחירה מבלי לגעת בקוד אחר.
- **הרחבה קלה** – ניתן להוסיף אסטרטגיות בחירה נוספות כמו `HighestPriceItemResultsPage` בלי לשנות את הקוד הקיים.
- **שימוש חוזר** – מתודת `addToCart()` אינה תלויה באסטרטגיה ספציפית, אלא רק בפריט שנבחר.
- **התאמה לריבוי תרחישים** – פתרון שמכסה גם תוצאות פשוטות וגם כאלה עם קלט סכום – באופן שקוף למשתמש.