

מסמך דרישות - מיני מערכת רישום משתמשים:

FastAPI → Kafka → MongoDB

איפיון שלב 2 – הזרמת JSON מקובץ (Batches)

2.1 מטרה

להוסיף ל-API יכולת לבצע **bulk publish** על בסיס קובץ JSON גדול הנמצא בתוך קונטיינר ה-API, ולפרסם את הנתונים ל-Kafka ב-batches מבוקרים.

השלב מתמקד ב:

- קריאה מקובץ JSON
- שליחה מבוקרת ב-batches
- ניהול לולאת פרסום תקינה
- עבודה עם JSON מקונן

2.2 קלט (Dataset)

יתווסף קובץ JSON בתוך ה-image של שירות ה-API, לדוגמה:

`api/data/users_with_posts.json`

מבנה הקובץ:

- מערך של Users.

- כל User כולל מערך פנימי בשם `posts`.
- ה-`posts` מקוננים ברמת ה-`user` בלבד (אין קינון עמוק נוסף).

הקובץ נקרא מתוך מערכת הקבצים של הקונטיינר בזמן ריצה.

Endpoint 2.3 חדש – Trigger ל-Seed

GET /seed

תפקיד:

להפעיל תהליך שמפרסם את הנתונים מתוך הקובץ ל-Kafka.

דרישות חובה

1. קריאת הקובץ מתבצעת פעם אחת בלבד.
 2. הנתונים מחולקים ל-`batches` של 10 משתמשים.
 3. כל 5 שניות נשלח `batch` נוסף.
 4. אין לשלוח את כל הנתונים בבת אחת.
-

מנגנון לולאה

ה-`endpoint` מאתחל לולאה שממשיכה לשלוח `batches` עד לסיום הנתונים.

נדרש:

- תנאי עצירה ברור כאשר אין עוד רשומות לשליחה.
- אין להשאיר לולאה אינסופית ללא תנאי יציאה.
- ניתן לממש באמצעות `while`, `flag` פנימי, או מנגנון אחר — כל עוד ההתנהגות תקינה.

Response לדוגמה

```
OK 200
}
, "status": "started"
"message": "seeding from file in batches of 10 every 5 seconds"
```

Kafka 2.4

- שימוש ב-topic הקיים: `users.registered`

Acceptance Criteria 2.5 – שלב 2

השלב ייחשב תקין אם:

- קיים קובץ JSON בתוך קונטיינר ה-API.
- קריאה ל-`GET /seed` מפעילה זרימה מדורגת ל-Kafka.
- נשלחים batches של 10 כל 5 שניות.
- ה-Consumer מצליח לקרוא את ההודעות.
- מומלץ לוודא שכל הרשומות נקלטו בהצלחה במסד הנתונים.

לא יינתן מספר רשומות צפוי מראש. עליכם לבצע ולידציה עצמאית.

שלב נוסף