

מבחן FullStack

FRONT

קיים שירות שמנהל קריאות API בשם UserService שמכיל פונקציה getUsers(). מהשרת (אין צורך לממש שרת ניתן לכתוב מוק דטה) ממש בהעלאת הקומפוננטה להצגת יוזרים קריאה של פונקציה זו לא מתוך הקומפוננטה ישירות ולאחר קבלת הדטה שמור בסטור.

1. צריך לנהל רשימת משתמשים ביישום שלך בעזרת NGRX. השתמש במבנה Entity (ספריה של NGRX) כדי להגדיר State שמנהל משתמשים:

- a. צור את ה-Adapter המתאים עבור משתמשים.
- b. הוסף את ה-state בצורה שתתמוך ב-Entity Management

2. נהל הוספה, מחיקה, עדכון, שמירה של משתמשים בסטור.
a. מנע הוספה של יוזרים כפולים עם אותו ID במידה ומדובר ביוזר קיים עדכן אותו במקום להוסיף חדש)

3. מבנה ה-state: (ניתן להחליף צורה לעבודה עם SIGNALS)

```
interface AppState {  
  users: {  
    entities: { [id: number]: User };  
    selectedUserId: number | null;  
  };  
  orders: {  
    entities: { [id: number]: Order };  
  };  
}  
  
interface User {  
  id: number;  
  name: string;  
}  
  
interface Order {  
  id: number;  
  userId: number;  
  total: number;  
}
```

- a. כתוב סלקטור שמחזיר את המשתמש הנבחר (selectedUser) על סמך ה-selectedUserId
- b. כתוב סלקטור שמחזיר את כל ההזמנות של המשתמש הנבחר.

4. צור קומפוננטה בשם UserOrdersComponent שמציגה את שם המשתמש הנבחר ואת סכום כל ההזמנות שלו. ודא שהקומפוננטה מאזינה לסלקטור שיצרת ומציגה את הנתונים.

מבחן FullStack

[BACK](#)

בנה ארכיטקטורה למערכת ניהול טיסות שמקבלת התראות על מחירים ושולחת התראות פוש למכשירים ניידים של המשתמשים.

דרישות המערכת:

1. קבלת התראות מחירים
המערכת צריכה לקבל מידע על מחירי טיסות ממקורות חיצוניים (כגון API של חברות תעופה או אתרי השוואת מחירים).
שליחת התראות פוש
2. המערכת צריכה לשלוח התראות פוש לטלפונים ניידים של המשתמשים כאשר מתגלים מחירים שמעניינים אותם.
3. ניהול התראות: (CRUD)
המערכת צריכה לנהל את המידע של המשתמשים וההתראות שלהם.
4. תמיכה בעומסים גבוהים
המערכת צריכה להיות מתוכננת לתמוך בעומסים גבוהים של בקשות והתראות.

אין צורך לפתח את כל הפרויקט, נדרש להציג:

1. תרשים ארכיטקטורה
2. מבני הנתונים שמשתמשים בכל רכיב

יש לממש ברמת הקוד רק את סעיף 3, רכיב ניהול ההתראות.

בסיום נא להעלות את הפרויקט ל- GIT ולשלוח קישור להורדה.