**React 04/03/2023 – Redux**

Auth

Authentication & Authorization

* Authentication – אימות ; בדיקת פרטי המשתמש בכדי לוודא שיש לו אישור כניסה
* Authorization - ניהול הרשאות – לאחר אימות המשתמש, אישור הפעולות שהוא רוצה לבצע.

JWT

* Json Web Token

זו שיטת Authentication and Authorization מאוד פופולרית כאשר עובדים עם SPL (single page application ).

JWT stands for JSON Web Token. JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. The client will need to authenticate with the server using the credentials only once. During this time the server validates the credentials and returns the client a JSON Web Token(JWT). For all future requests the client can authenticate itself to the server using this JSON Web Token(JWT) and so does not need to send the credentials like username and password.

השיטה עובדת כך:

1. כאשר המשתמש מבצע הרשמה ) register ) , הוא שולח לשרת אובייקט (בדרך כלל איזשהו User מודל – המכיל את פרטי המשתמש ( . השרת, הוא בודק האם הפרטים תקינים ( האם הכל חוקי, האם הפרטים נכונים, האם האימייל לא תפוס וכו).

אם הכל חוקי, הוא שומר את הפרטים ב DB (מסד נתונים) ומפיק מחרוזת (token) המכילה בתוכה (בצורה מקודדת) את האובייקט עצמו (ללא סיסמה). כמו כן, הוא מחזיר תאריך יצירה ותאריך תפוגה.

ה FRONT, מקבל את מחרוזת התווים הזו שנקראת token ( JWT) . השרת לא שומר את המחרוזת אצלו, ה FRONT מחוייב לשמור את ה Token.

1. כאשר המשתמש מבצע Login, הוא שולח לשרת גם אובייקט המכיל את פרטי הכניסה. השרת בודק האם הפרטים תקינים (מבצע בדיקת ולידציה מול ה db) ומנפיק עוד הפעם מחרוזת בצורה מקודדת של אותו User מודל (ללא הסיסמה) . גם כאן, הוא מחזיר את המחרוזת הזו (token) ל FRONT וגם כאן הוא לא שומר את המחרוזת הזו אצלו.
2. כאשר המשתמש רוצה לפנות לשרת( ל service ) כדי לקבל מידע שמצריך אותו להיות רשום, הוא חייב לשלוח את ה token הזה יחד עם הבקשה.
3. השליחה של ה token, מתבצעת בפורמט מאוד ספציפי, דרך ה Headers של הבקשה. ( request headers ).
4. השליחה דרך ה headers, נעשית בפורמט הבא:

Authorization: Bearer ${token}

1. השרת במקרה כזה, מקבל את ה header, מחלץ את ה token ובודק אם הוא חוקי (כלומר, אם הוא הנפיק אותו, אם הפרטים של המשתמש נמצאים ב db , וכו וכו, האם תאריך התפוגה לא עבר). אם הכל בסדר, הוא מאפשר לו לבצע את הפעולה, אם הוא לא, הוא מקפיץ שגיאה
2. כאשר המשתמש מבצע יציאה (logout) , אז הוא לא צריך לשלוח שום דבר לשרת . ה front צריך למחוק את ה token

להתקין : npm i jwt-decode