25.02.2024

שיעור 9

Context: **useContext**  
האופציה לשים סטייט באופן גלובלי. למנוע פרופס וuplifting

Redux – שיטת עבודה "design pattern", mobX. בהתאם לזה יש לריאקט את context.

להוציא סטייט החוצה – הוא יהיה מחובר לכל קומפוננטה.

בעצם כל מי ש"מאזין" לסטייט הזה ידע להשתנות.

* לא לשים כל דבר בקונטקסט. אם יש תקשורת בין 2-3 קומפוננטות אז נשתמש בuplifting, כי זה משויך רק אישי לקומפוננטות. קונטקס הוא משהו גלובלי גדול יותר, שנשתמש רק ברגע שיש שימוש עם כמה קומפוננטות שרחוקות אחת מהשנייה מבחינת תקשורת.

מייצרים קומפוננטה שהיא תהיה רק לקונטקסט, רק לסטייט הגלובלי.   
כותבים createContext() – צריך לייבא אותה לריאקט. קונטרול ספייס

A white background with black text

Description automatically generated

צריך לשלוח לו props

A black text on a white background

Description automatically generated

אומרים לו provider – מה תביא לי את היכולות:   
כותבים value וגם props.children – זה מילים של השפה.

ואז זה אומר שכל מי שירצה יוכל לגשת לזה.

לא מתקינים ספריות ברגע שאני עם השרת פתוח לכן תמיד עושים קונטרול C

import uuidfrom 'uuid';

מתקינים את זה: npm I uuid

נותן לי ID ייחודי – זה מביא משהו ארוך כזה מכוער עם אותיות ומספרים ו"-"

להשלים

כדי להשתמש:

שמים בתוך APP עטיפה פנימית בתוך הfragment <> </> בתוך זה שמים <HobbyContextProvider> מעל ומתחת לקוד.  
  
ואז כדי להשתמש בפונקציות עושים useContext ובסוגריים שמים את השם של ה"קונסט" בתווך הקומפוננטה.

אקספורט – אם יש לי כמה אקספורטים אני יכולה לכתוב רק על אחד דיפולט. כשזה בלי סוגריים אז זה ייקח את הדיפולט. ואם אני רוצה משהו שהוא לא דיפולט אז אני אכתוב {} סוגריים מסולסלים ובתוכו את השם.

1. כותבים useContext ואת השם של הקונטקסט ורושמים שאנחנו רוצים את הפונקציה addHobby.