

# React Redux & Store

מדריך ושימושים

# Redux Store

## אחסון מידע משותף של אפליקציה

במהלך פיתוח אפליקציה,  
לא פעם אנו נתקלים במצבים בהם ברצוננו לאחסן מידע לשימוש חוזר.

ברמת הרכיבים שלנו (React Components), כדי לאחסן מידע אנחנו משתמשים בstate, ושומרים את המידע שלנו ונתונים של אובייקטים במבנה נתונים מסוג עץ ניתן לחשוב על store כstate משותף לכל האפליקציה.

דוגמא ל state ברכיב ריאקט,  
השייך לו, ולו בלבד.

```
const SomeComponent = (props) => {  
  const [name, setName] = useState('Hello World')  
  return (  
    <div>{name}</div>  
  )  
}
```

נחמד שאפשר לשמור state לכל רכיב ריאקט,  
אבל מה עם אנחנו צריכים לאחסן מידע שמשותף לכל האפליקציה?  
נתונים של המשתמש? מועדפים?

כאן הספרייה redux נכנסת לתמונה  
בעזרת store נוכל לעשות זאת

# ייבוא ספריית Redux

```
# NPM  
npm install @reduxjs/toolkit
```

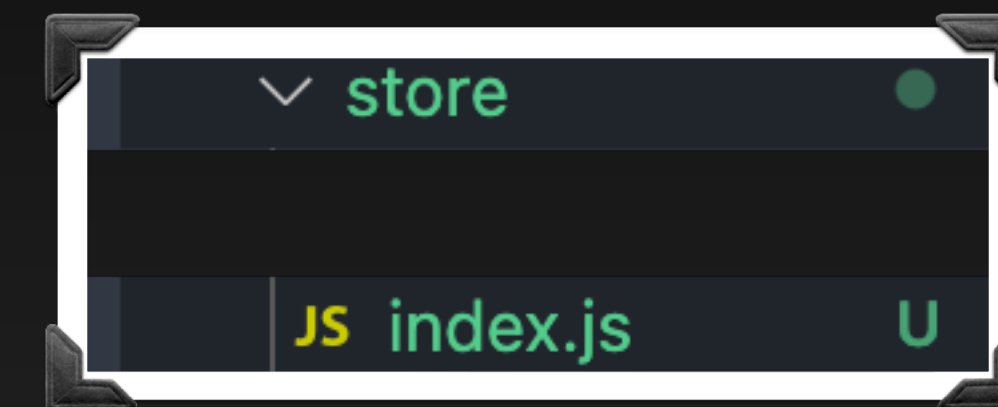
```
# NPM  
npm install redux
```

בטרמינל, נריץ את הפקודות הבאות  
לייבוא הספרייה וכלים נוספים של הספרייה

# נתחיל בלעקוב אחרי שלושה השלבים

1

נפתח תיקייה חדשה בפרויקט בשם  
store וניצור בה קובץ חדש index.js



2

נדביק את השורה הבאה בראש הקובץ,  
אנחנו מתכוונים להשתמש בשתי השיטות הללו  
createSlice, configureStore

```
import { createSlice, configureStore } from '@reduxjs/toolkit'
```

3

כדי ליצור state משותף חדש,  
נשתמש בשיטה createSlice אליה אנחנו מעבירים  
עץ ריק {}. בדוגמא שלנו  
נשמור נתונים של המשתמש, לכן נקרא ל 'נתח' שלנו userSlice

```
import { createSlice, configureStore } from '@reduxjs/toolkit'

const userSlice = createSlice(
  {
  }
)
```

# createSlice, initialState, reducers

## שיטות ושימושיים בstore

בreducers אנחנו נגדיר את השיטות שיישמשו אותנו לשנות את העץ השיטות (reducers) הם מסוג פונקציות חץ, לשיטות מועבר המצב נוכחי של העץ (state) וקלט חיצוני (action) state: ישמש אותנו לגשת לתכונות העץ, ולשנות את העץ

action: פרטמר בו נוכל להעביר את הערך החדש לתכונה בעץ, או לבצע עבודה אחרת בעץ.

כדי להוציא את הערך שהעברנו לשיטה מaction, נשתמש בתכונה payload שמחזיקה את אותו ערך שהועבר

המחשה:

setUserAge(25) // action.payload = 25

מילון נוח:

initialState: המצב ההתחלתי של ה'נתח' במחסן שלנו  
reducers: השיטות שממשות לשנות את ה'נתח'

בinitialState אנחנו נגדיר את הערך ההתחלתי של העץ שלנו, לא חייב לשים ערכים, אפשר להשאיר ריק בדוגמא שלנו יש אובייקט עם 3 מאפיינים userName, userAge, userGender המאותחלים עם ערכים תחיליים כלהלן

```
const userSlice = createSlice({
  initialState: {
    userName: 'Nadav',
    userAge: 22,
    userGender: 'male'
  },
  name: 'userState',
  reducers: {
    setUsername: (state, action) => { state.userName = action.payload },
    setUserAge: (state, action) => { state.userAge = action.payload },
    setUserGender: (state, action) => { state.userGender = action.payload }
  }
})
```

בname נגדיר שם ייחודי ל'נתח' שלנו במחסן לדוגמא: userState

# המחשה נוספת

אם העץ שלי נראה ככה:

```
const user = {  
  userName: 'Nadav',  
  userAge: 21  
}
```



העץ ייראה ככה ב'נתח':

```
initialState: {  
  userName: '',  
  'userAge': 0  
}
```

הערה:

ניתן להכניס מאפיינים  
התחלתיים,  
וניתן להשאיר אותם ריקים.

אם היה לי שיטה שאחראית לשנות

name לuser מסוים

```
function setName(name) {  
  user.name = name  
}
```



ככה היא תראה כפונקציית חץ

reducer

שאחראית על שינויים בstore:

```
reducers: {  
  setName: (state, action) => {  
    state.name = action.payload  
  }  
}
```

הערה:

action.payload = name  
העורך שנעביר לsetName ימצא ב payload

# הגדרת פוינטרים לשימוש ב'נתח'

```
const userSlice = createSlice({
  initialState: {
    userName: 'Nadav',
    userAge: 22,
    userGender: 'male'
  },
  name: 'userState',
  reducers: {
    setUsername: (state, action) => { state.userName = action.payload },
    setUserAge: (state, action) => { state.userAge = action.payload },
    setUserGender: (state, action) => { state.userGender = action.payload }
  }
})

export const actions = userSlice.actions

const store = configureStore({
  reducer: {
    userReducer: userSlice.reducer
  }
})
export default store
```

נגדיר 2 קבועים (const) באותו הקובץ:

## const actions

לקבוע זה ניגש מקבצים אחרים,  
לשם גישה למאפיינים של ה'נתח'

## const store

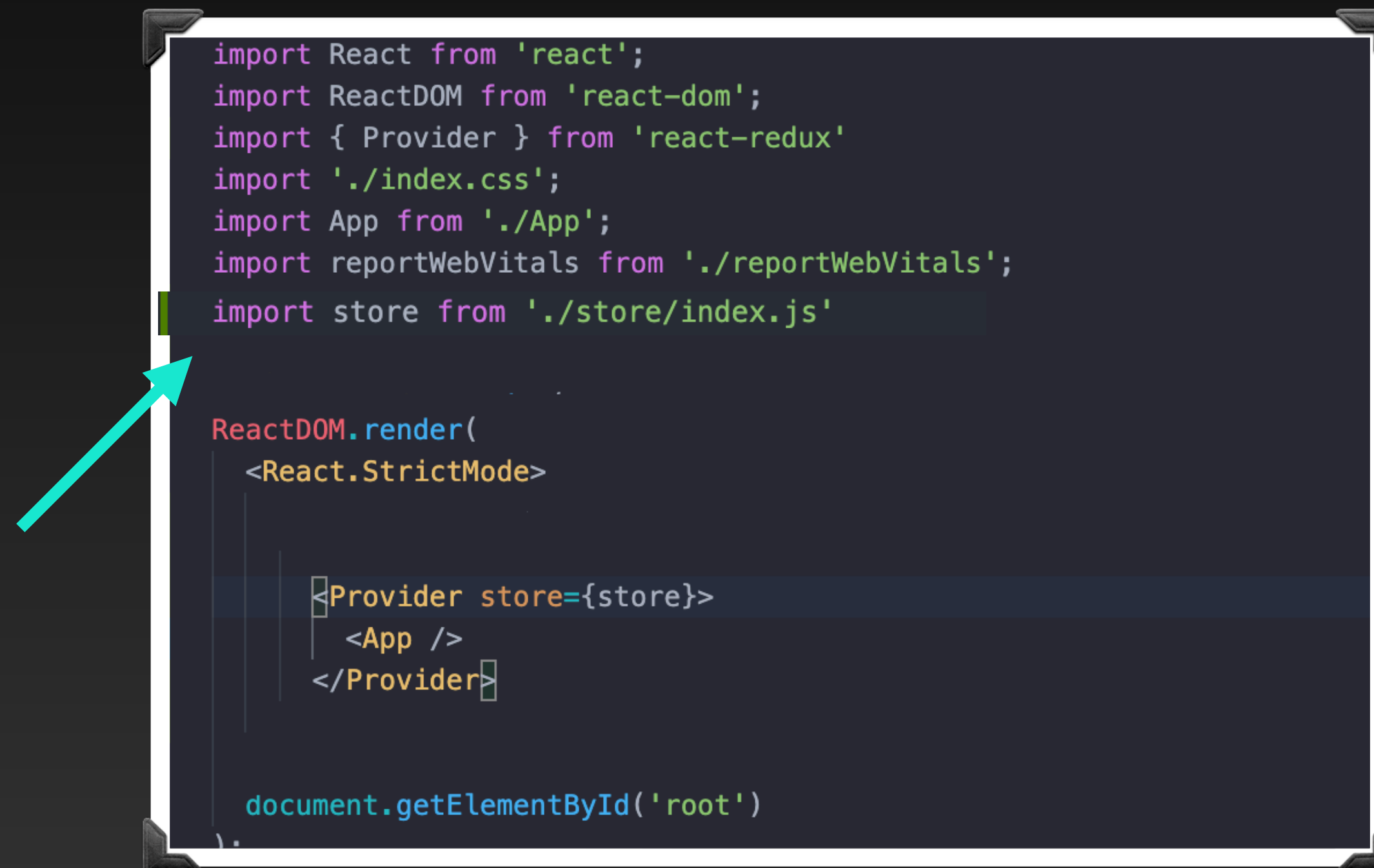
קבוע זה מגדיר לנו את הstore,  
אנחנו מעבירים לו אובייקט מהצורה בדוגמא  
כאשר באובייקט reducers, נגדיר כמאפיינים  
את הreducers שבהם נרצה להשתמש, במקרה שלנו  
הreducer של userSlice

לבסוף, נבצע ייצוא בעזרת המילה export  
לשם שימוש בהם בשאר הפרויקט



כעת כדי להגדיר את הstore שבנינו להיות זה שהאפליקציה משתמשת בו,  
ננווט לקובץ index.js שנמצא בתיקיית src  
ונעטוף את <App/> בתגית

```
<Provider store = {store}>  
  <App/>  
</Provider>
```



```
import React from 'react';  
import ReactDOM from 'react-dom';  
import { Provider } from 'react-redux';  
import './index.css';  
import App from './App';  
import reportWebVitals from './reportWebVitals';  
import store from './store/index.js';  
  
ReactDOM.render(  
  <React.StrictMode>  
    <Provider store={store}>  
      <App />  
    </Provider>  
  ,  
  document.getElementById('root')  
)
```

**לא לשכוח!**  
לייבא את store מהקובץ שלנו  
store/index.js



# שליפת מידע מהstore, שינוי מידע בstore

## הצגת מידע מהstore ברכיב ריאקט

```
import store from "../store/index.js"
function SomeComponent(props) {
  const userState = useSelector(state => state.userReducer)

  return (
    <div>
      <h1>{userState.userName}</h1>
    </div>
  )
}
export default SomeComponent
```

כדי לגשת למידע בstore  
נשתמש בhook, **useSelector**

```
const userState = useSelector(state => state.userReducer)
const userName = userState.userName
```

## שינוי מידע ב store ברכיב ריאקט

```
import { useDispatch } from "react-redux"
import { actions } from "../store/index.js"
function SomeComponent(props) {

  const dispatch = useDispatch()
  const setName = (name) => {dispatch(actions.setName(name))}

  return (
    <div>
      <button onClick={() => setName('Shlomi')}>Change name to Shlomi!</button>
    </div>
  )
}
export default SomeComponent
```

שימוש בשיטות מהstore יבוצעו  
דרך הhook, **useDispatch**  
אם נרצה להגדיר שיטה שניגשת לשיטות אחרות בstore  
נקרא לה בצורה הבאה:  

```
const dispatch = useDispatch()
const myAction = dispatch(actions.myAction())
```

# תרגיל - Liked Photos

א. בנה 2 עמודים באמצעות Browser

- בעמוד הראשון יהיה רשימת תמונות מלאה (4) עם כפתור לכל אחת להוספה למועדפים.
- בעמוד השני יופיעו רק המועדפים, אם אין אז תופיע רשימה ריקה.

השתמש ב redux store כדי לעשות זאת