

תרגיל תכנות:

react-router

העברת מידע באמצעות ה-url

מטרות התרגיל

- תקשורת בין קומפוננטות באמצעות ה-router
- החזרת המשתמש לאותו מצב גם כאשר הוא טוען מחדש את העמוד
- מימוש חלק מהתרגיל באמצעות קומסוננטות מבוססות פונקציה כדי לתרגל hooks ולהבין את היתרונות של hooks

תרגיל:

בואו ניצור את האפליקציה הבאה...

תרגיל: todo עם חיפוש

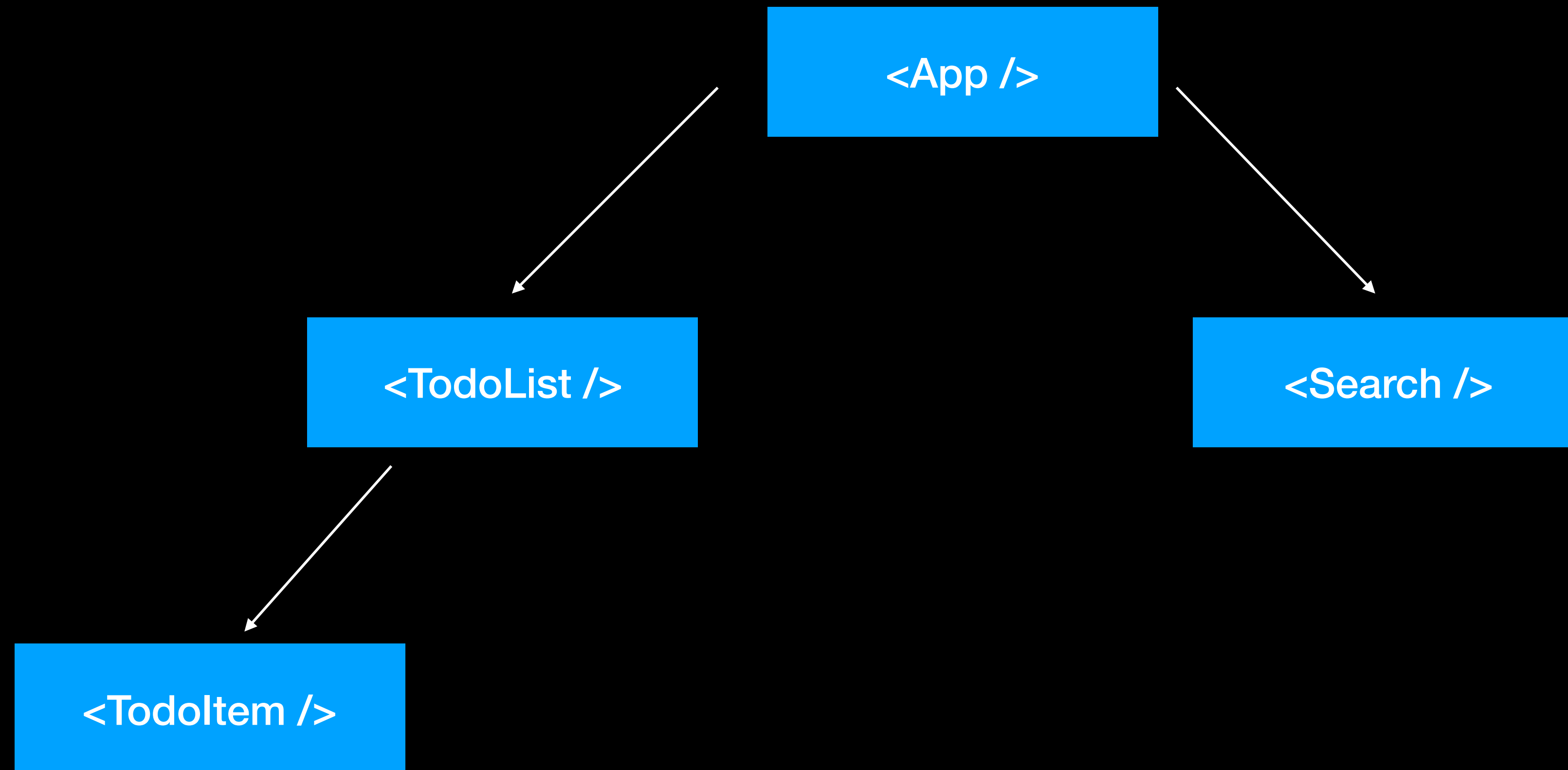
- בתרגיל זה אתם צריכים ליצור את ארבעת הקומפוננטות הבאות:

- `<App />`

- `<TodoList />`

- `<TodoItem />`

- `<Search />`



הערות

- לצורך עיצוב האפליקציה אנחנו נשתמש ב- bootstrap
- את ה- `<TodoList />` נסו לעשות פעם אחת עם `class component` ופעם אחת עם `function component`
- השרת יושב בכתובת הבאה:
- `https://nztodo.herokuapp.com/api/task/?format=json&search=<search-string>`

תרגיל:

נסו לפתור את התרגיל בעצמכם.

הפתרון בהמשך...

<App />

- צריך להיות עטוף ב-BrowserRouter

<Search />

- צריך לשנות את הנתיב דרך אובייקט ה- history שצריך להגיע מה- router
- לצורך זה נצטרך לעטוף את ה- component עם withRouter

<TodoList />

- כדי שה- component ירונדר מחדש בכל שינוי ב- url וכדי שנוכל לשלוף את משתנה ה- query param אנחנו צריכים לעטוף את ה- component ב- withRouter
- כדי לשלוח בקשה לשרת באתחול של ה- component נצטרך לממש את componentDidMount
- כדי לשלוח בקשה בכל עדכון של ה- url נצטרך לממש componentDidUpdate
- את הפיכת ה- query params לאובייקט אנחנו נשתמש בספרייה שנקראת:
 - query-string
- בואו ננסה לממש את ה- component כפונקציה עם hooks וגם כ- class

<TodoItem />

- מקבל אובייקט json שמייצג task יחיד
- מדפיס את הכותרת והתיאור של ה-task

סיכום

- דברים כמו חיפוש ופילטור אנחנו צריכים לסמן אותם ב- url כדי לשמור על המשתמש באותו המצב כאשר הוא עושה reload
- אם המידע צריך לעבור במילא דרך ה- url אני יכול לנצל את העובדה כדי לתקשר בין קומפוננטות באמצעות ה- url שזה יוריד לי את הצורך לתקשר בין הקומפוננטות בדרך נוספת כמו redux