 Since your backend API is ready and tested, **Day 3 is all about connecting your React frontend to your Express backend**. You’ll make your React app fetch and update todos from the backend instead of just using local state.

**NOTE::**  
**local state=> frontend local..so memory lost when u refresh** You were storing todos in React’s state (useState([])).

So on page refresh, it reset everything (because state lives in memory of the page).

when we connect react and express..it is saved in Node’s server memory and state is not lost.

 You're storing todos in your **backend's memory** (let todos = [] in Node).

 React fetches from the backend every time you load — so the todos *persist* while the server is running!

**🔥 But Warning: It's only temporary memory**

Right now:

* When you refresh React (frontend) → ✅ data stays.
* When you **restart your Node server** (Ctrl + C → node index.js again) → ❌ data is gone.

This is because:

let todos = []; // ← stored only in RAM  
  
it is An in-memory array in Express

To **make it permanent**, you'll need:

**Option 1: 🗂 Save todos to a file (like todos.json)**

**Option 2: 💾 Use a database like MongoDB or SQLite**

**Connect React Frontend to Express Backend**

**1. Make Sure Both Apps Are Running**

* Start your Express backend (e.g., node index.js on port 5000 or 8080).
* Start your React frontend (npm start on port 3000).

**2. Fetch Todos from the Backend**

**Replace your local todos state in React with data fetched from your backend.**

**Scene1:**

**Example (in App.js):**

Import React,( useState, useEffect }from ‘react’;

Function App(){

const[todos,setTodos] = useState([]);

const [input,setInput] =useState(‘ ‘);

useEffect(()=>{

fetch(‘http://localhost:5000/todos’)

.then(res=>res.json())

.then(data=>setTodos(data))

.catch(err=>console.error(‘Error: ‘ , err));

},[]);

const addTodo=()=>{

if(input.trim()===’ ‘)

return;

fetch(‘http://localhost:5000’/todos’ ,{

method:’POST’,

headers:{‘Content-type’ : ‘application/json’},

body: JSON.stringify({text:input , done:false}),

})

.then(res=>(res.json())

.then(newTodo=>setTodos([…todos,newTodo]));

setInput(‘ ‘);

const deleteTodo=(index)=>{

fetch(‘http://localhost:5000/todos/${index}’ , {

method:’DELETE’})

.then( ()=> setTodos(todos.filter( (\_,i) => i! ==index)));

}

const toggleDone=(index)=>{

fetch(‘http://localhost:5000/todos/${index}’ , {

method:’PUT’

})

.then(res=>res.json())

.then(updatedTodo =>{

const newTodos=[…todos];

newTodos[index]= updatedTodo;

setTodos(newTodos);

});

};

const clearAll=()=>{

todos.forEach( (\_,i)=>{

fetch(‘http://localhost:5000/todos/${i}’ , {

method: ‘DELETE’} );

});

setTodos([]);

};

return(

<div> <h1>todo list </h1>

<input value={input}

onChange = { e=> setInput(e.target.value)}

placeholder= ‘add a todo’ />

<button onClick ={ addTodo}>ADD </button>

<button onClick ={ clearAll}> clear </button>

<ul>

{todos.map( (todos,index) => { <li key ={index}>

{todo.done ? ‘completed’ : ‘yet to do’ } {todo.text}

<button onClick={ () => toggleDone(index) }> toggle </button>

<button onClick={ ()=> deleteTodo(index) }> delete </button>

</li>

))}

</ul>

</div>

);

}

export default App;

**Scene2:  
in TodoList.js**

//react with express for in-memory and no localstate

import React, { useState, useEffect } from 'react';

import TodoItem from './TodoItem';

function TodoList() {

  const [todos, setTodos] = useState([]);

  const [input, setInput] = useState('');

  // 🔁 Get todos on load

  useEffect(() => {

    fetch('http://localhost:5000/todos')

      .then(res => res.json())

      .then(data => setTodos(data))

      .catch(err => console.error('Error:', err));

  }, []);

  const addTodo = () => {

    if (input.trim() === '') return;

    const newTodo = { text: input, done: false };

    fetch('http://localhost:5000/todos', {

      method: 'POST',

      headers: { 'Content-Type': 'application/json' },

      body: JSON.stringify(newTodo),

    })

      .then(res => res.json())

      .then(savedTodo => {

        setTodos([...todos, savedTodo]);

        setInput('');

      });

  };

  const deleteTodo = (index) => {

    fetch(`http://localhost:5000/todos/${index}`, {

      method: 'DELETE',

    }).then(() => {

      setTodos(todos.filter((\_, i) => i !== index));

    });

  };

  const toggleDone = (index) => {

    fetch(`http://localhost:5000/todos/${index}`, {

      method: 'PUT',

    })

      .then(res => res.json())

      .then(updatedTodo => {

        const newTodos = [...todos];

        newTodos[index] = updatedTodo;

        setTodos(newTodos);

      });

  };

  const clearAll = () => {

    todos.forEach((\_, i) => {

      fetch(`http://localhost:5000/todos/${i}`, { method: 'DELETE' });

    });

    setTodos([]);

  };

  return (

    <div>

      <h1>To-do List</h1>

      <input

        value={input}

        onChange={e => setInput(e.target.value)}

        placeholder='add a todo'

      />

      <button onClick={addTodo}>ADD</button>

      <button onClick={clearAll}>clear</button>

      <ul>

        {todos.map((todo, index) => (

          <TodoItem

            key={index}

            todo={todo}

            index={index}

            toggleDone={toggleDone}

            deleteTodo={deleteTodo}

          />

        ))}

      </ul>

    </div>

  );

}

export default TodoList;

**Test Everything**

* **Add, delete, and toggle todos from the frontend and see the changes reflected from the backend.**
* **Use browser dev tools (Network tab) to debug API calls.**