

```
Copy code
```

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

function FetchExample() {
  const [data, setData] = useState([]);
  const [error, setError] = useState<string | null>(null);

  useEffect(() => {
    fetch("https://jsonplaceholder.typicode.com/todos")
      .then((response) => {
        if (!response.ok) {
          throw new Error("Network response was not ok");
        }
        return response.json();
      })
  })
}
```

```

        .then(setData)
        .catch(setError);
    }, []);

    if (error) return <div>Error: {error}</div>;

    return (
      <ul>
        {data.map((todo: any) => (
          <li key={todo.id}>{todo.title}</li>
        ))}
      </ul>
    );
  }
}

```

Sử dụng Axios:

Axios cung cấp API thân thiện hơn với các tính năng như timeout, interceptors.

Ví dụ:

```

tax
Copy code

import axios from "axios";
import { useEffect, useState } from "react";

function AxiosExample() {
  const [data, setData] = useState([]);
  const [error, setError] = useState<string | null>(null);

  useEffect(() => {
    axios
      .get("https://jsonplaceholder.typicode.com/todos")
      .then((response) => setData(response.data))
      .catch((error) => setError(error.message));
  }, []);

  if (error) return <div>Error: {error}</div>;

  return (
    <ul>
      {data.map((todo: any) => (
        <li key={todo.id}>{todo.title}</li>
      ))}
    </ul>
  );
}

```

3. Quản lý Dữ liệu Toàn Cục

React Context API:

Phù hợp cho ứng dụng nhỏ, đơn giản.

Ví dụ:

```

tax
Copy code

import React, { createContext, useContext, useState } from "react";

const TodoContext = createContext<any>(null);

export function TodoProvider({ children }: { children: React.ReactNode }) {
  const [todos, setTodos] = useState<string[]>([]);

  const addTodo = (todo: string) => setTodos([...todos, todo]);

  return (
    <TodoContext.Provider value={{ todos, addTodo }}>
      {children}
    </TodoContext.Provider>
  );
}

export function useTodo() {
  return useContext(TodoContext);
}

// Sử dụng
function TodoList() {
  const { todos } = useTodo();
  return (
    <ul>
      {todos.map((todo, index) => (
        <li key={index}>{todo}</li>
      ))}
    </ul>
  );
}

```

Redux:

Dùng khi cần quản lý dữ liệu lớn với nhiều state phức tạp.

```

bash
Copy code

# Cài đặt Redux Toolkit
npm install @reduxjs/toolkit react-redux

```

Ví dụ:

```

tax
Copy code

import { configureStore, createSlice } from "@reduxjs/toolkit";
import { useDispatch, useSelector, useReducer } from "react-redux";

```

```
import { Provider, useDispatch, useSelector } from "react-redux";

const todoSlice = createSlice({
  name: "todos",
  initialState: [] as string[],
  reducers: {
    addTodo: (state, action) => {
      state.push(action.payload);
    },
  },
});

const store = configureStore({
  reducer: {
    todos: todoSlice.reducer,
  },
});

const { addTodo } = todoSlice.actions;

export function TodoApp() {
  const dispatch = useDispatch();
  const todos = useSelector((state: any) => state.todos);

  return (
    <div>
      <button onClick={() => dispatch(addTodo("New Task"))}>Add Task</button>
      <ul>
        {todos.map((todo: string, index, number) => (
          <li key={index}>{todo}</li>
        ))}
      </ul>
    </div>
  );
}

export function App() {
  return (
    <Provider store={store}>
      <TodoApp />
    </Provider>
  );
}
```

Zustand:

Nhẹ hơn Redux, dễ dùng.

bash

Copy code

```
npm install zustand
```

Ví dụ:

tsx

Copy code

```
import create from "zustand";

const useStore = create((set) => ({
  todos: [] as string[],
  addTodo: (todo: string) =>
    set((state) => ({ todos: [...state.todos, todo] })),
}));

function ZustandExample() {
  const { todos, addTodo } = useStore();

  return (
    <div>
      <button onClick={() => addTodo("New Task")}>Add Task</button>
      <ul>
        {todos.map((todo, index) => (
          <li key={index}>{todo}</li>
        ))}
      </ul>
    </div>
  );
}
```

Lựa chọn công cụ nào sẽ phụ thuộc vào quy mô và yêu cầu của ứng dụng. Nếu bạn cần hỗ trợ sâu hơn hoặc triển khai cụ thể, hãy cho mình biết! 🤗

🗨️ 📄 📌 🔄 🗑️

Is this conversation helpful so far?

👍

👎

✕

Previous 30 Days



Upgrade plan

More access to the best models

Message ChatGPT



ChatGPT can make mistakes. Check important info.

?