

# Summary Of Context Api

🔥🤖! حلو جدًا يا صاحبي، ركز معايا كدا علشان نلخص الموضوع بطريقة سهلة وسريعة

## ✂️ أول حاجة يا معلم.. هنرتب الدنيا صح!

ات بتاعتنا بدل ما context علشان نلم فيه كل الـ `contexts` هنعمل فولدر في المشروع اسمه `contexts`. نرميها في أي حته

شكل الفولدر هيبقى كدا:

```
/src
├── /contexts
│   ├── AuthContext.tsx (اللي مسؤول عن تسجيل الدخول والخروج)
│   ├── ThemeContext.tsx (اللي بيبدل بين الوضع الليلي والعادي)
│   └── index.ts (علشان يسهل الاستيراد)
```

## ✂️ ثاني حاجة.. هنعهد البيانات اللي هنتخزن

لازم يكون فيه حاجتين مهمين Context كل:

✅ نوع البيانات اللي هنخزنها (مثلاً: `user`, `theme`)

✅ الفانكشنات اللي هنتعامل بيها مع البيانات (مثلاً: `login`, `logout`, `toggleTheme`)

## مثال سريع على `AuthContext`

```
import { createContext, useContext, useState, ReactNode } from "react";

interface User {
  id: number;
  name: string;
}

interface AuthContextType {
  user: User | null;
  login: (userData: User) => void;
}
```

```
logout: () => void;  
}
```

```
const AuthContext = createContext<AuthContextType | null>(null);
```

## **Provider** ثالث حاجة.. هنظبط الـ

هو اللي بيوزع البيانات دي في التطبيق كله، فهنعمله بالشكل ده **Provider** الـ

```
export const AuthProvider: React.FC<{ children: ReactNode }> = ({ children })  
=> {  
  const [user, setUser] = useState<User | null>(null);  
  
  const login = (userData: User) => setUser(userData);  
  const logout = () => setUser(null);  
  
  return <AuthContext.Provider value={{ user, login, logout }}>{children}</AuthContext.Provider>;  
};
```

## **useAuth** رابع حاجة.. هنظبط

جاهز عشان نسهل الشغل **Hook** هنعمل `useContext(AuthContext)` بدل ما كل مرة نكتب

```
export const useAuth = () => {  
  const context = useContext(AuthContext);  
  if (!context) throw new Error("useAuth must be used within an AuthProvider");  
  return context;  
};
```

## عشان الاستيراد **index.ts** في Contexts خامس حاجة.. هندمج كل الـ يكون أسهل

```
export { AuthProvider, useAuth } from "./AuthContext";
export { ThemeProvider, useTheme } from "./ThemeContext";
```

## 📌 **main.tsx أو \_app.tsx في Providers سادس حاجة.. هنستخدم الـ**

```
import { AuthProvider, ThemeProvider } from "@/contexts";

const MyApp = ({ Component, pageProps }) => {
  return (
    <AuthProvider>
      <ThemeProvider>
        <Component {...pageProps} />
      </ThemeProvider>
    </AuthProvider>
  );
};

export default MyApp;
```

## 📌 **سابع حاجة.. ازاي نستخدمه في أي مكون؟**

دلوقتي نقدر نستخدمه بسهولة كدا:

```
import { useAuth } from "@/contexts";

const Profile = () => {
  const { user, login, logout } = useAuth();

  return user ? (
    <div>
      <h2>أهلاً, {user.name} 🙌</h2>
      <button onClick={logout}>تسجيل خروج</button>
    </div>
  ) : (
```

```
<button onClick={() => login({ id: 1, name: "Abdallah" })}>تسجيل دخول</button>  
);  
};
```

## 🎯 الخلاصة في كلمتين

- 1 ونظبط الملفات جواه `contexts` هنعمل فولدر
  - 2 هنعدد البيانات والفانكشنات اللي هنستخدمها
  - 3 جاهز Hook `useContext` و `Provider` هنعمل
  - 4 `_app.tsx` أو `main.tsx` في Providers هنخط ال
  - 5 اللي عملناها في أي مكون بسهولة Hooks هنستخدم ال
- 🔥!محترم، منظم، وسهل التوسع Context كذا بقى عندك نظام 🤖👏  
👊🔥 لو محتاج أي تعديل أو إضافة، قولني وأنا معاك