

React useContext

#React

#Guide

Good Practices

In React is a good practice to create a custom hook for useContext instead of use directly. The following videos explain how to create a custom react useContext hook:

[Always Use a Custom Hook for Context API](#)

Performance TIP

Use a two context per hook, one for the items that changes and one for the items that remains the same, the one that not change must be in a top level because parents components trigger re-renders in children components when something change.

Example Code

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

type SearchContextProviderProps = {
  children: React.ReactNode;
};

type SearchContext = {
  searchText: string;
  isLoading: boolean;
  countProducts: number;
};

type ActionsContext = {
  setSearchText: React.Dispatch<React.SetStateAction<string>>;
  setIsLoading: React.Dispatch<React.SetStateAction<boolean>>;
  setCountProducts: React.Dispatch<React.SetStateAction<number>>;
};

export const SearchContext = createContext<SearchContext | null>(null);
export const ActionsContext = createContext<ActionContext | null>(null);

export default function SearchContextProvider({ children }:
SearchContextProviderProps) {
```

```

const [searchText, setSearchText] = useState("");
const [isLoading, setIsLoading] = useState(false);
const [countProducts, setCountProducts] = useState(0);

return (
  <ActionsContext.Provider
    value={{
      setSearchText,
      setIsLoading,
      setCountProducts,
    }}
  >
    <SearchContext.Provider
      value={{
        searchText,
        isLoading,
        countProducts,
      }}
    >
      {children}
    </SearchContext.Provider>
  </ActionsContext.Provider>
);
}

```

```

export const useSearchContext = () => {
  const context = useContext(SearchContext);
  if (!context) throw new Error("useSearchContext must be used within SearchContextProvider");
  return context;
};

```

```

export const useSearchActionsContext = () => {
  const context = useContext(ActionsContext);
  if (!context) throw new Error("useActionsSearchContext must be used within ActionsContextProvider");
  return context;
};

```