

Copilot

D'accord, voici les parties des deux codes qui sont différentes :

Premier code :

```
const [last, setLast] = useState(null);

useEffect(() => {
  if (data) {
    const sortedEvents =
      data &&
      data.events &&
      [...data.events].sort((a, b) => new Date(b.date) - new Date(a.date));

    setLast(sortedEvents && sortedEvents[0]);
    return;
  }
  getData();
});

return (
  <DataContext.Provider
    // eslint-disable-next-line react/jsx-no-constructed-context-values
    value={{
      data,
      error,
      last,
    }}
  >
    {children}
  </DataContext.Provider>
);
```

Deuxième code :

```
useEffect(() => {
  if (data) return;
  getData();
});

return (
  <DataContext.Provider
    // eslint-disable-next-line react/jsx-no-constructed-context-values
    value={{
      data,
      error,
    }}
  >
    {children}
  </DataContext.Provider>
);
```

Dans le premier code, un état supplémentaire `last` est défini et utilisé pour stocker le dernier événement trié par date. Cette logique de tri et de stockage n'est pas présente dans le deuxième code. De plus, le `Provider` du `DataContext` dans le premier code fournit `last` en plus de `data` et `error`, tandis que dans le deuxième code, il ne fournit que `data` et `error`.