* Context cung cấp cách để thực hiện chia sẻ dữ liệu xuyên suốt các component tree mà không cần truyền data theo từng cấp bậc thông qua props

|  |  |
| --- | --- |
| *Lifting state up* | *Context* |

* **Context được thiết kế để** 
  + Chia sẻ "global data" của toàn bộ ứng dụng React (broadcast data)

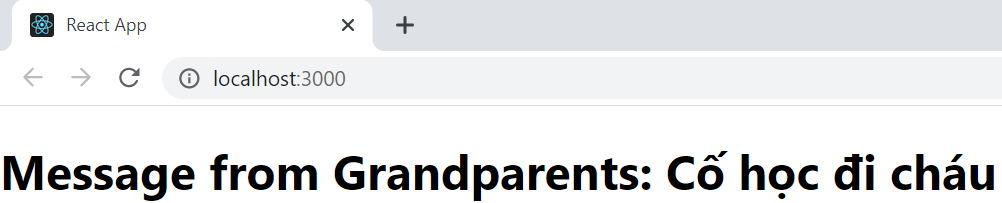
VD: thông tin user hiện tại đang đăng nhập, theme, language mà người dùng đã chọn

* + Data được sử dụng ở nhiều component (ở nhiều tầng khác nhau) trong component tree

Có thể sử dụng Component Composition để thay thế

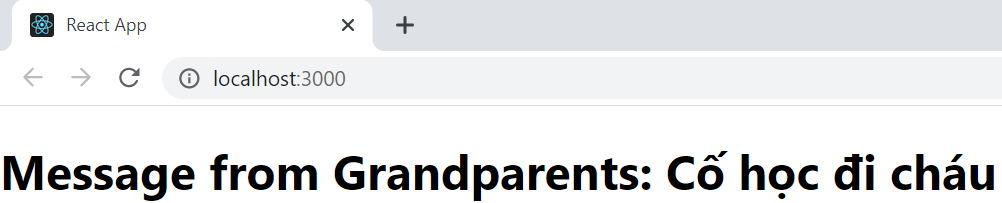
VD:

|  |  |  |
| --- | --- | --- |
|  |  |  |

**

*Send data thông qua props*

|  |  |  |  |
| --- | --- | --- | --- |
| *B1: create context object* | *B2: create provider* | *Component trung gian không cần truyền props gì cả* | *Functional component*    *Class component*  *B3: create consumer (using context)* |

**

*Broadcast data trong context*

* **Các thành phần**
  + **Context**
    - Là đối tượng lưu giá trị của context hiện tại
    - Create context object



defaultValue

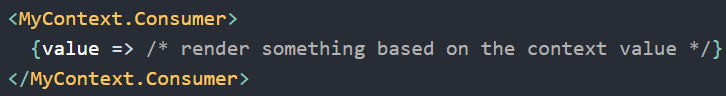
* + - * Là giá trị mặc định của props nếu không khai báo provider
      * Chỉ sử dụng khi component không có Provider nào bên trên nó trong component tree

Return 1 object bao gồm: Provider và Consumer

* + **Provider**



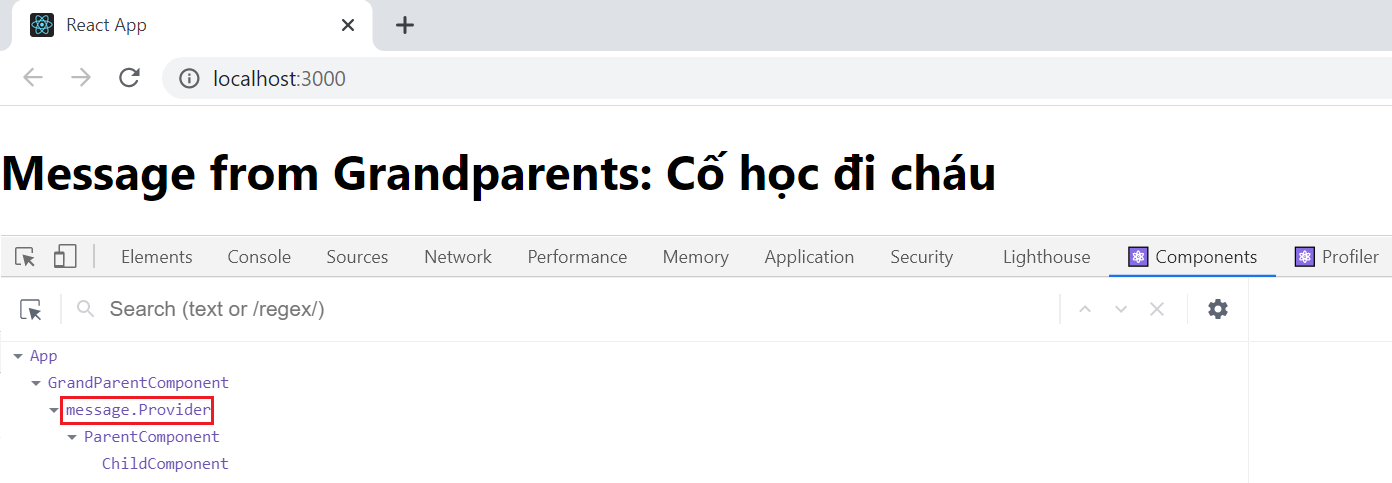
* + - Là 1 Component của React **cung cấp giá trị cho context**
    - Mỗi Context Object phải đi kèm với 1 Provider (nếu không có sẽ lấy defaultValue khi create context object)
    - Provider Component nhận 1 value (là props) và truyền tới Consuming Component
  + **Consumer**

****

* + - Là 1 component **sử dụng giá trị của context**
    - Tất cả các Consuming Component sẽ re-render khi provider thay đổi giá trị của context
    - Không phụ thuộc vào shouldComponentUpdate method
  + **Class.contextType**
    - Sử dụng để gán 1 Context vào Component hiện tại
    - Có thể sử dụng được ở mọi method trong lifecycle
    - Có 2 cách để gán contextType

|  |  |
| --- | --- |
| *Class.contextType* | *Sử dụng public class fields syntax* |

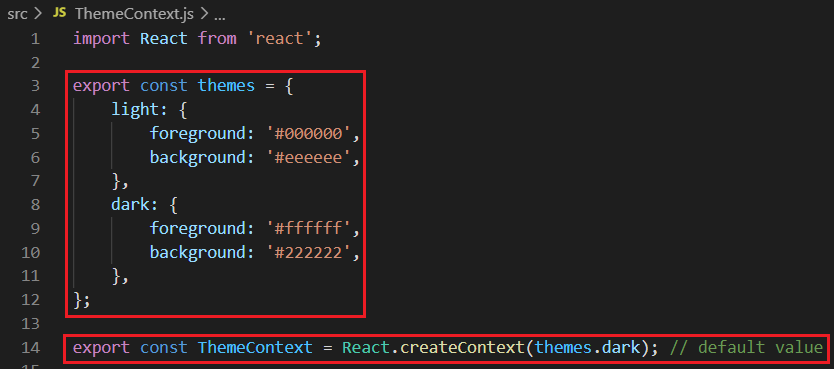
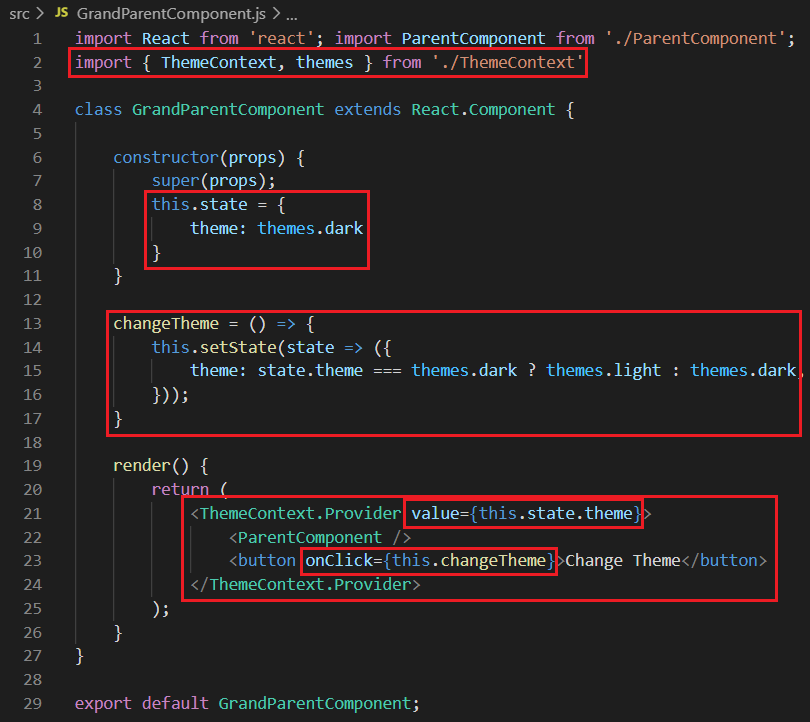
* + **Context.displayname**
    - React DevTools sử dụng để hiển thị cho context.

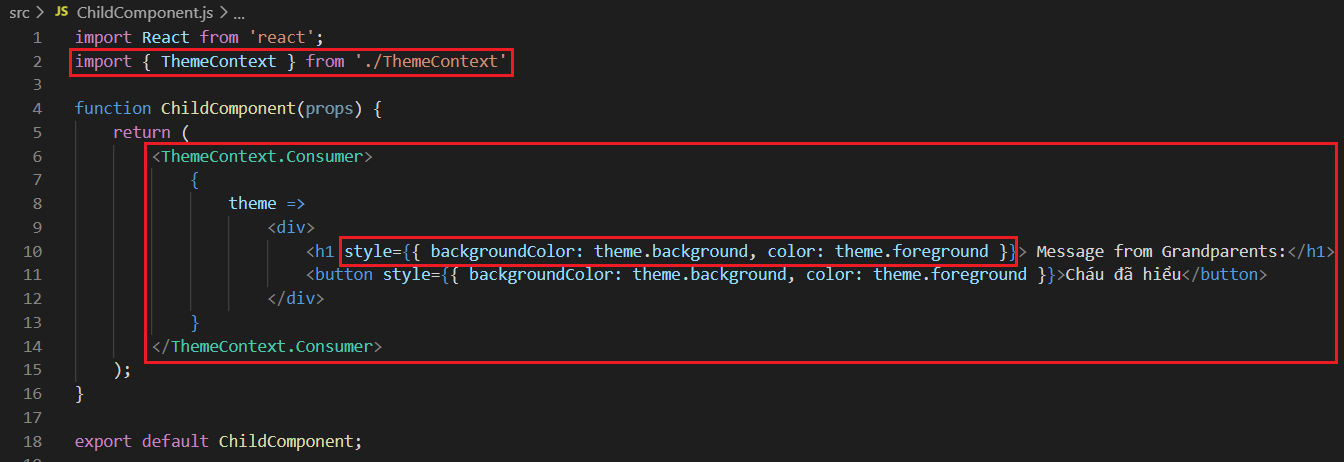


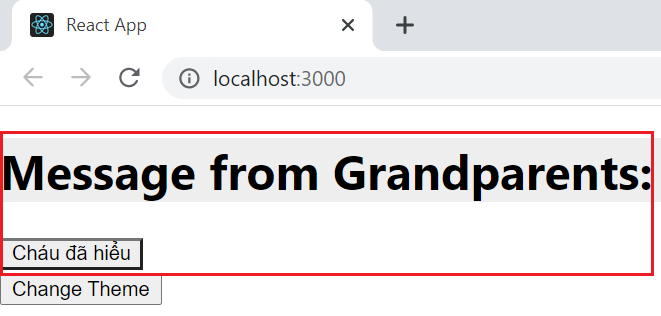
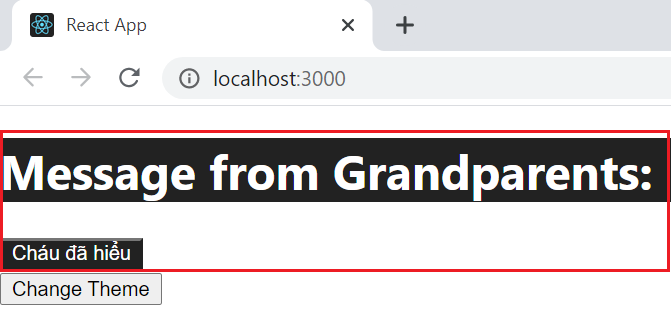
* **Technical**
  + **Context kết hợp với State**

Khi state thay đổi thì tất cả các Consumer trong Component tree sẽ update lại

VD: Change Themes

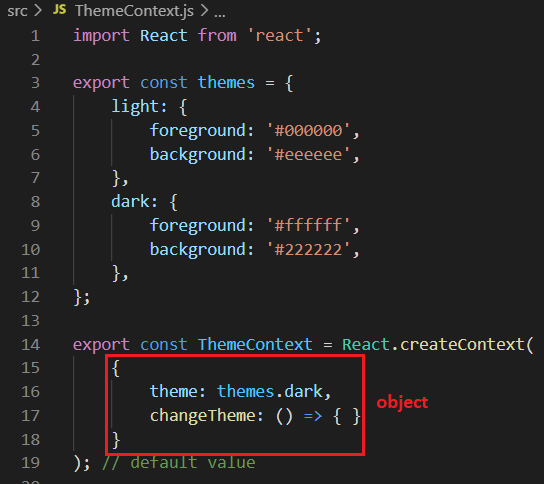
 

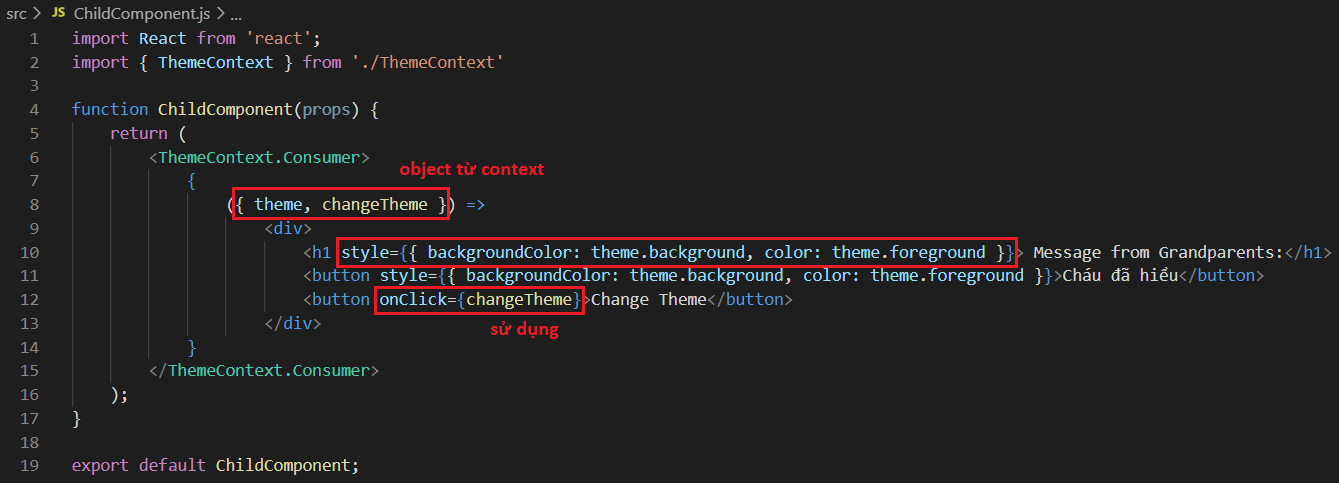
 

* + **Update Context từ Child Component**

Truyền một function xuống theo context để cho phép consumers cập nhật context đó

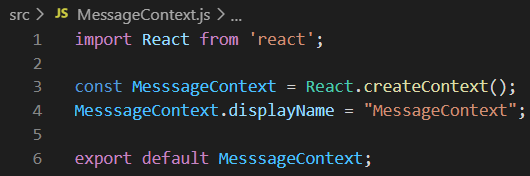
 

* + **Consuming nhiều Contexts**

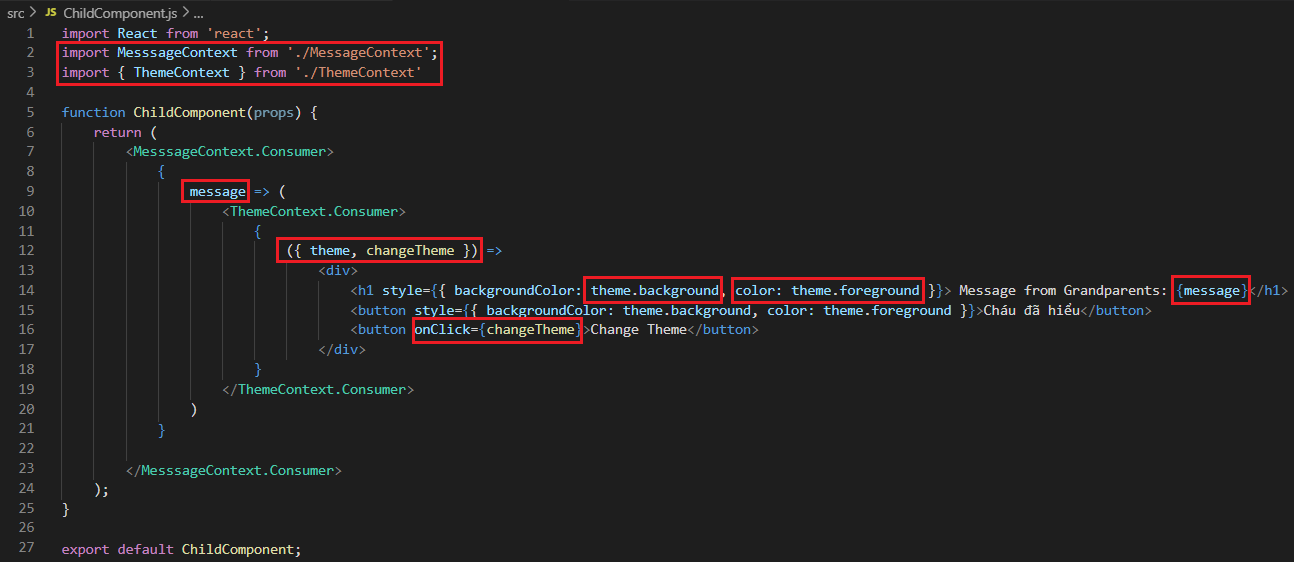
Tách ra nhiều context để re-render Component tree nhanh

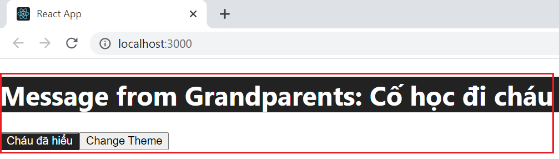
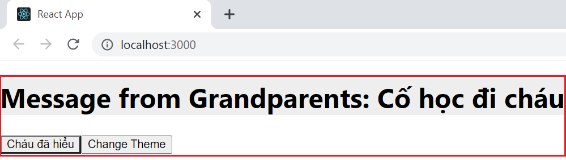
Provider và Consumer lồng nhau

*Context*

 **



* *

* **Reference**

<https://reactjs.org/docs/context.html>