

Custom hooks

Os ***custom hooks*** são funções JavaScript que utilizam hooks do React, e podem ser reutilizadas em vários componentes funcionais. Suas principais características são:

1. Prefixo com ***use***, exemplo: `useFetch`, `useTheme`, etc;
2. Devem utilizar outros hooks do React, caso contrário podem ser funções normais;
3. Um custom hook pode gerenciar seu próprio estado, lógica e efeitos colaterais;
4. Não renderiza componentes, apenas encapsula lógica.

São, resumidamente, funções que utilizam hooks do React, e podem ser reutilizadas em vários componentes funcionais.

Comumente utilizados quando possuímos lógica compartilhada entre vários componentes, por exemplo, uma função que realiza uma requisição HTTP:

```
import React from 'react'

const useFetch = (url) => {
  const [data, setData] = React.useState(null)
  const [loading, setLoading] = React.useState(true)
  const [error, setError] = React.useState(null)

  React.useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await fetch(url)
        const result = await response.json()
        setData(result)
      } catch (err) {
        setError(err)
      } finally {
        setIsLoading(false)
      }
    }
    fetchData()
  }, [url])

  return { data, loading, error }
}
```

```
import React from 'react'
import { View, Text } from 'react-native'
import useFetch from './useFetch' // Importe o hook

const ReactComponent = () => {
  const { data, isLoading, error } = useFetch('https://api.example.com/data')

  if (isLoading) {
    return <Text>Carregando...</Text>
  }

  if (error) {
    return <Text>Houve um erro: {error.message}</Text>
  }

  return (
    <View>
      <Text>Retorno da chamada: {JSON.stringify(data)}</Text>
    </View>
  )
}

export default ReactComponent
```

Links úteis:

Reusando logica com custom hooks