

REACTJS BEST PRACTICE



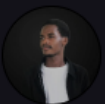
@máriobatalhasilva

CENTRALIZE EXTERNAL COMPONENTS



```
1 //Don't import directly
2 import { Button } from 'styled-components';
3 import { Typeform } from 'react-typeform';
4
5 //import your component from a global folder
6 import { Button, TypeForm } from '@share/components';
```

in case you use components from others packages or library, is a best practise centralize all the components in shared folder, because when you import in your local component, you'll not need to know from where it comes.

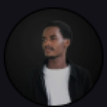


@mariobatalhasilva

IMAGE ITERATION

iterate images is the most common problem when you're import images from isolated file or folder. the best practise is import images from a shared folder or create an array of imported images.

```
1  import React from "react";
2
3  import Logo from "../logo.png";
4  import User from "../user.png";
5  import ImgBackground from "../background.png";
6
7  const images = [Logo, User, ImgBackground];
8
9  const Home = () => {
10    return (
11      <>
12        <div className="card">
13          {images.map((item, index) => {
14            <img alt="" src={item} key={index} />;
15          })}
16        </div>
17      </>
18    );
19  };
```



EVITE SHORT-CIRCUIT OPERATOR

```
1 //Try do not use short-circuit operator
2 const home;
3
4 const MyHomeComponent = () => {
5     return <div>{home && <p>Home: {home}</p>}</div>
6 }
7
8 //Instead use ternary operator
9 const MyHomeComponent = () => {
10     return home ? <p>Home: {home}</p> : null
11 }
12
13 //Or double negation
14 const MyHomeComponent = () => {
15     return <div>{!!home && <p>Home: {home}</p>}</div>
16 }
```

short-circuit operator &&, sometimes can return 0 as value. it's happen because react doesn't treat 0 as false, for that we have to use ternary operator or double negation to solve our problem.

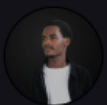


OBJECT LITERALS INSTEAD OF SWITCH

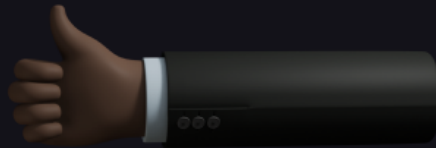
most case when we need to use conditional statement we prefer to use if/else or switch/case. the problem with switch/case is that it's doesn't perform so well.

object literals is more simple, and we can use for example to instantiate a component.

```
1  const PageComponent = ({componentType}) => {
2    const MyComponent = {
3      "home": <HomeComponent />,
4      "profile": <ProfileComponent />,
5      "product": <ProductComponent />,
6      "default": <ErrorPageComponent />,
7    }
8
9    return MyComponent[componentType] || MyComponent["default"];
10 }
```



SHARE AND LEAVE A COMMENT



@máriobatalhasilva