**Opção 1: Usar um Alias no Webpack**

Essa abordagem é recomendada se você quiser organizar melhor os caminhos dos imports no seu projeto.

**Passo 1: Verificar o uso do Webpack**

Certifique-se de que você está usando o Webpack diretamente ou tem acesso à sua configuração (webpack.config.js).  
Se você usa create-react-app, precisará ejetar o projeto ou usar o pacote react-app-rewired.

**Passo 2: Configurar o Alias**

Edite o arquivo webpack.config.js (ou crie um, se necessário) e adicione a configuração do alias. Aqui está o código completo:

const path = require('path');

module.exports = {

resolve: {

alias: {

utils: path.resolve(\_\_dirname, 'src/utils'), // Ajuste 'src/utils' para o caminho correto da sua pasta utils

},

},

};

Isso cria um alias chamado utils que aponta para a pasta src/utils.

**Passo 3: Ajustar o Import**

Depois de configurar o alias, ajuste o import no arquivo Form.js:

import handleSubmit from 'utils/handleSubmit';

**Passo 4: Configuração com React App (opcional)**

Se você estiver usando create-react-app e não quiser ejetar, instale o react-app-rewired:

npm install react-app-rewired

Crie ou edite o arquivo config-overrides.js no diretório raiz do seu projeto:

const path = require('path');

module.exports = function override(config) {

config.resolve.alias = {

...config.resolve.alias,

utils: path.resolve(\_\_dirname, 'src/utils'),

};

return config;

};

No arquivo package.json, substitua os scripts por:

json

Copy code

"scripts": {

"start": "react-app-rewired start",

"build": "react-app-rewired build",

"test": "react-app-rewired test"

}

Reinicie o servidor (npm start) e teste os imports.

**Opção 2: Criar um Symlink**

Esta é uma solução rápida e menos comum, mas funciona bem em certos cenários.

**Passo 1: Criar o Symlink**

No terminal, no diretório raiz do projeto, execute o comando abaixo para criar um link simbólico:

ln -s ../../utils/handleSubmit.js ./node\_modules/handleSubmit.js

**Passo 2: Ajustar o Import**

Depois de criar o symlink, ajuste o import no arquivo Form.js para usar o atalho:

import handleSubmit from 'handleSubmit';

**Passo 3: Testar a Configuração**

Reinicie o servidor do React (npm start) e certifique-se de que o problema foi resolvido.

**Qual escolher?**

1. **Alias no Webpack:** Ideal para projetos grandes ou equipes que seguem padrões. É organizado e escalável.
2. **Symlink:** Uma solução rápida para projetos pequenos ou problemas pontuais.