**Publish React components as an npm package.**

There are several reasons why you might want to publish React components as an npm package:

**1. Reusability:** By packaging your React components into a standalone npm package, you make it easier for other developers to reuse those components in their own projects. This can save time and effort, as developers don't have to reinvent the wheel and can instead use your pre-built components.

**2.Modularity**: Packaging your React components as an npm package can also help you break down your code into smaller, more manageable modules. This can make your codebase easier to maintain and update, as you can make changes to individual modules without affecting the rest of the codebase.

**3.Collaboration:** Publishing your React components as an npm package can also make it easier to collaborate with other developers. By sharing your code in a standardized format, you can facilitate communication and reduce friction when working with others.

**4.Versioning:** When you publish your React components as an npm package, you can use the npm versioning system to manage updates and changes to your code. This makes it easy to track changes, revert to previous versions, and ensure that your code is always up-to-date.

**Steps to create a react component as an npm package.**

1. Create an react app using command **“npx create-react-app app-name”.**
2. In Src folder remove every file except “App.css , App.js , Index.js”.
3. Add a folder “lib” that will store everything which we need to publish in npm.
4. Inside lib folder there is a folder component in which all components will come which we want in npm.
5. Inside component I have created a file “Cta.js” or any name you like and respective css file too.

**Index.js**

1. import React from 'react';
2. import './Cta.css';
3. export const Cta = (props) => {
4. return (
5. <button className={`btn btn--${props.kind} CTA`}
6. data-id={props.id}
7. type={props.type}
8. name={props.name}
9. value={props.value}
10. disabled={props.disabled}
11. onClick={props.handleClick}>
12. {props.label}
13. </button>
14. );
15. }

**Cta.css**

.btn--primary {

  color: #fff;

  background-color: #007bff;

  border-color: #007bff;

}

.btn--primary:hover {

  color: #fff;

  background-color: #0069d9;

  border-color: #0062cc;

}

.btn--secondary {

  color: #fff;

  background-color: #6c757d;

  border-color: #6c757d;

}

.btn--secondary:hover {

  color: #fff;

  background-color: #5a6268;

  border-color: #545b62;

}

.btn--success {

  color: #fff;

  background-color: #28a745;

  border-color: #28a745;

}

.btn--success:hover {

  color: #fff;

  background-color: #218838;

  border-color: #1e7e34;

}

.btn--danger {

  color: #fff;

  background-color: #dc3545;

  border-color: #dc3545;

}

.btn--danger:not(:disabled):not(.disabled):active {

  color: #fff;

  background-color: #bd2130;

  border-color: #b21f2d;

}

.btn--warning {

  color: #212529;

  background-color: #ffc107;

  border-color: #ffc107;

}

.btn--warning:hover {

  color: #212529;

  background-color: #e0a800;

  border-color: #d39e00;

}

.btn {

  display: inline-block;

  font-weight: 400;

  text-align: center;

  white-space: nowrap;

  vertical-align: middle;

  -webkit-user-select: none;

  -moz-user-select: none;

  -ms-user-select: none;

  user-select: none;

  border: 1px solid transparent;

  padding: 0.375rem 0.75rem;

  font-size: 1rem;

  line-height: 1.5;

  border-radius: 0.25rem;

  transition: color 0.15s ease-in-out, background-color 0.15s ease-in-out,

    border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;

}

**6.** In next step you can install

**npm install --save-dev @babel/core @babel/cli @babel/preset-env npm install -save @babel/polyfill**

**7.** Next create **babel.config.json.**

{

  "presets": [

    [

      "@babel/env",

      {

        "targets": {

          "edge": "17",

          "firefox": "60",

          "chrome": "67",

          "safari": "11.1"

        },

        "useBuiltIns": "usage",

        "corejs": "3.6.5"

      }

    ],

    "@babel/preset-react"

  ]

}

**8.** Next in **Package.json** in **script** section add

**For windows**

"build": "del -rf dist && mkdir dist &&  babel src/lib/components -d dist --copy-files"

**For Mac**

"build": "rm -rf dist && mkdir dist &&  babel src/lib/components -d dist --copy-files"

**9.** run the command **“npm run build”**

**10.** Alter the **package.json** for **publishing**

**11. Set private is “false” in Package.json**

  "name": "dynamic-cta",

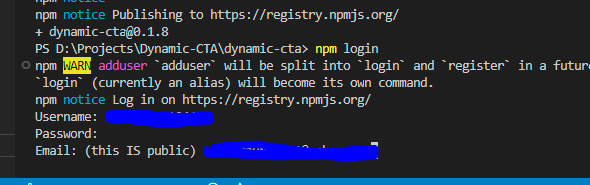
  "version": "0.1.0",

  "private": false,

**12. description**, **keywords**, and **author** are all optional fields that will give potential end users a better idea of the package.

13. for Publishing the react component run command **“npm login”.**

**PS D:\Projects\Dynamic-CTA\dynamic-cta> npm login**



**After that run “npm publish”**

**Now your react component is publicly access from NPM. Use the npm module as suggested in the url :**

<https://www.npmjs.com/package/dynamic-cta?activeTab=dependencies>

Codebase: https://github.com/rahulgaur1812/dynamic-cta

Thank You