**PUBLISH PACKAGES NPM**

**SCOPE PACKAGE**

**Creating package**

1. Create account on npm

2. Create folder with package name

3. Inside that folder, run -> npm init --scope=@my-username

Example:

npm init --scope=@muneebwaseem78 (create package.json file)

4. Create some index.ts file and write any function

5. npm i @types/node (optional)

6. tsc index.ts

**Publishing**

1. npm adduser
2. npm publish --access public (package will be created with @username/foldername)

Example: @muneebwaseem78/muneeb-npm-package@1.0.0

Testing package in some other folder

1. Create folder
2. Inside that folder, run npm init first to get package.json
3. Install package (npm i @muneebwaseem78/muneeb-npm-package@1.0.0 (version optional as only first version))
4. Import function from package

Example:

import {print} from '@muneebwaseem78/muneeb-npm-package';

print()

We can find further information from this tutorial:

<https://docs.npmjs.com/creating-and-publishing-scoped-public-packages>

**PUBLISHING GLOBAL PACKAGES**

1. Create folder (muneeb-npm-global)
2. npm init
3. npm i @types/node
4. create folder names ‘lib’ and make files in it (greet.ts). Make sure function and filename should be same

Example:

export function greet():string{

    return 'Hey! Good evening my friend.'

}

1. first do tsc in lib folder to generate .js files (tsc lib/greet.ts)
2. create another bin folder at root
3. In bin folder, create index.ts file and call function in that file

Example:

import { greet } from "../lib/greet";

let greeting:string = greet();

console.log(greeting);

Now till now this cannot be run from cmd, for that we use npm package process

So bin index.ts file will be:

#!/usr/bin/env node

import { greet } from "../lib/greet";

let greeting:string = greet();

console.log(greeting);

let s:string[] = process.argv.splice(2);

console.log(s)

Make sure to add below command at first line of index.ts in bin folder

|  |  |
| --- | --- |
|  | #!/usr/bin/env node |

now time to compile it:

tsc bin/index.ts

node bin/index.js a b (it will return [‘a’, ‘b’])

1. Now to tell system to run index.ts file under bin directory, we need to do some changes in package.json file (we need to update path of main and add bin key.

"main": "./lib/index.js",

  "bin": {"muneeb": "./bin/index.js"},

When we say Muneeb, it will run index.js file under bin folder

Updated Package.json file:

{

  "name": "muneeb-package-global",

  "version": "1.0.0",

  "description": "",

  "main": "./lib/index.js",

  "bin": {"muneeb": "./bin/index.js"},

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1"

  },

  "author": "",

  "license": "ISC",

  "dependencies": {

    "@types/node": "^16.10.1"

  }

}

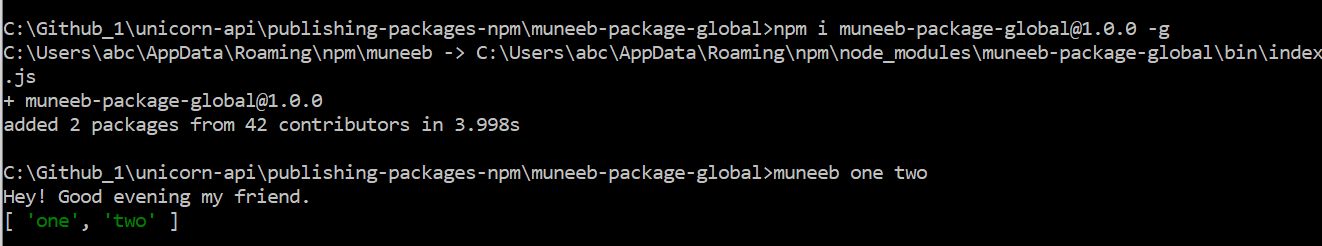
1. Make sure to compile index.ts file in bin folder before publishing (node bin/index.ts)
2. Now publish package (npm publish --access public)
3. After that install package globally in your system with –g

Example:

npm i [muneeb-package-global@1.0.0](mailto:muneeb-package-global@1.0.0) –g

1. Now Run CLI with name passed in bin under package.json file

Example:



**To Update Package:**

1. Update your work in either lib folder (make sure to do tsc lib/index.ts) or in bin folder in index.ts file (make sure to do compile tsc bin/index.ts)
2. Make sure to compile both files
3. Now in package.json, update version under version key like 1.0.1.

"version": "1.0.0",

1. Now publish package again (npm publish --access public)
2. Again install package

npm i muneeb-package-global@1.0.1 –g

1. Run in CLI