

1: Crear la Librería en Angular Instalar Angular CLI si no lo tienes:

```
npm install -g @angular/cli
```

Crear el Workspace y la Librería: (Opcional puedes crear la libreria en cualquier parte sin vincularla al workspace)

```
ng new my-workspace --create-application=false
cd my-workspace
ng generate library my-library
```

Agregar código a la Librería:

Modifica los archivos en projects/my-library/src/public-api.ts para exportar tu funcionalidad. Por ejemplo, puedes agregar un servicio:

```
// projects/my-library/src/lib/my-library.service.ts
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root',
})
export class MyLibraryService {
  constructor() { }

  public sayHello(): string {
    return 'Hello from MyLibraryService!';
  }
}

// projects/my-library/src/public-api.ts
export * from './lib/my-library.service';
```

Nota: Librería ejemplo

Para construir la Librería:

```
ng build my-library
```

2. Configurar Nexus.

1. Ve a la pestaña de “Repository” en la configuración.
2. Haz clic en “Create repository” y selecciona “npm (hosted)”.
3. Configura el repositorio con un nombre, como npm-hosted, y ajusta las demás configuraciones según tus necesidades.
4. Guarda los cambios.

3. Preparar Librería para Publicar en Nexus

Para publicar librería es necesario configurar Nexus local con credenciales, entonces en una consola dentro del proyecto.

```
npm login --registry=http://10.1.40.130:8081/repository/npm-hosted/
```

Ingresa usuario y contraseña de Nexus

En el archivo C:\Users\{usuario}\.npmrc se genera la credencial como el siguiente ejemplo;

```
//10.1.40.130:8081/repository/npm-hosted/:_authToken=NpmToken.963f3fcd-2bb5-3570-aa39-acb05c1fe44f
```

Asegurate de agregar al archivo .npmrc de la librería la ruta y las credenciales.

```
@mflibs:registry=http://10.1.40.130:8081/repository/npm-hosted/  
//10.1.40.130:8081/repository/npm-hosted/:_authToken=NpmToken.963f3fcd-2bb5-3570-aa39-acb05c1fe44f
```

Asegurate de agregar el publishConfig en el package.json

```
"publishConfig": {  
  "@mflibs:registry": "http://10.1.40.130:8081/repository/npm-hosted/"  
}
```

#### 4. Publicar la Librería:

Primero, asegúrate de estar en el directorio del workspace (donde está el archivo `package.json` principal) y luego publica la librería:

```
cd dist/my-library  
  
npm publish      ó      npm publish --  
registry=http://10.1.40.130:8081/repository/npm-hosted/
```

```

$ npm publish
npm WARN publish npm auto-corrected some errors in your package.json when publishing. Please run "npm pkg fix" to address these errors.
npm WARN publish errors corrected:
npm WARN publish Removed invalid "scripts"
npm notice
npm notice 📦 @mflibs/encryption-lib@1.0.15
npm notice === Tarball Contents ===
npm notice 1.9kB esm2022/lib/encryption-lib.module.mjs
npm notice 10.1kB esm2022/lib/encryption.service.mjs
npm notice 496B esm2022/mflibs-encryption-lib.mjs
npm notice 509B esm2022/public-api.mjs
npm notice 4.1kB fesm2022/mflibs-encryption-lib.mjs
npm notice 6.1kB fesm2022/mflibs-encryption-lib.mjs.map
npm notice 127B index.d.ts
npm notice 310B lib/encryption-lib.module.d.ts
npm notice 564B lib/encryption.service.d.ts
npm notice 729B package.json
npm notice 87B public-api.d.ts
npm notice === Tarball Details ===
npm notice name: @mflibs/encryption-lib
npm notice version: 1.0.15
npm notice filename: mflibs-encryption-lib-1.0.15.tgz
npm notice package size: 6.6 kB
npm notice unpacked size: 25.1 kB
npm notice shasum: 7679af21086daf858fcbc57fa7ad323cc3268934
npm notice integrity: sha512-4jc/VJ6qp1QYo[...]skwu4ygwCmb1g==
npm notice total files: 11
npm notice
npm notice Publishing to http://10.1.40.130:8081/repository/npm-hosted/ with
tag latest and default access
+ @mflibs/encryption-lib@1.0.15

```

Paso 5: Instalar la Librería usando npm link (Para probar local) Crear un enlace simbólico global:

```

cd dist/my-library
npm link

```

Enlazar la Librería en tu Proyecto Angular:

En el directorio de tu proyecto Angular:

```

npm link my-library

```

6: Importar la Librería desde Nexus;

En el directorio de tu proyecto principal de angular Angular, debe crear .npmrc y agregar las credenciales;

```

@mflibs:registry=http://10.1.40.130:8081/repository/npm-hosted/
//10.1.40.130:8081/repository/npm-
hosted/:_authToken=NpmToken.963f3fcd-2bb5-3570-aa39-acb05c1fe44f

```

En el directorio de tu proyecto principal de Angular, ejecutar en la consola;

```

npm install @mflibs/encryption-lib@1.0.15 --
registry=http://10.1.40.130:8081/repository/npm-hosted/

```

**Nota:** Con la librería ya agregada al package.json y el archivo .npmrc con las credenciales, no es necesario volver a instalar en el proyecto, solo es para la primera construcción.

### **El orden adecuado para realizar cambios y publicar sería el siguiente;**

1. Realizar cambios
2. git add .
3. git commit -m 'fix: comentario'
4. git push (Aquí se aumenta la versión con la tarea del Hook, si no lo tiene configurado debe aumentar manual en package.json y utilizar git push -no-verify)
5. npm install
6. npm run build
7. cd dist/my-library
8. npm publish

### **Consideraciones Finales**

Asegúrate de tener correctamente configurado tu archivo .npmrc y de utilizar el token de acceso adecuado para autenticarte en el registro de Nexus. Además, asegúrate de que la URL del registro sea accesible desde tu red.

Si sigues estos pasos, deberías poder crear, publicar, y consumir tu librería Angular de manera efectiva utilizando tanto npm link como el registro privado de Nexus.

[←Regresar](#)

From:  
<http://wiki.adacsc.co/> - **Wiki**

Permanent link:  
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:front:libs>

Last update: **2024/08/06 14:23**

