Creating library

Here are the steps to create a library. The structure of the library is the same as a regular Angular project.

Create new project

You can create a new project by executing the following command.
”ng new (name)”

Generate library

You can generate you’re library by executing the following command.
“ng generate library (name)”

The library is now located at ../projects/(name)

Public components/services

Here I will explain how you can make components/services available for use outside the library.

Components

First you need declare the components in the library module. Next you want to export the components in the module. And finally you will have to export everything from the component in the public-api.

Once you have done these things you will be able to use the component outside the library once it is installed into you’re application by using the selector name of the component between element tags in the html.

Services

To make a service available for use outside the library you will need to make the following function inside the exported class NameModule.

```
static forRoot(): ModuleWithProviders<NameModule> {

  return {

   ngModule: NameModule,

   providers: [ NameService, NameService]

  };

 }
```

You will have to put all the services you want exported in the providers array. Once you have done this you will be able to import the service from the library in you’re application and use it.

Make sure when you import the module in the app.module file from your application that you call the forRoot() function. So the import will look like this: EcareLibModule.forRoot()

Include external dependencies

Here i will explain how you can include dependencies for your library if needed. For example the ngx-translate library. This will cause ngx-translate to automatically install with our library.

To include a dependency in our library the following must be added to our package.json.

```
"dependencies": {
   "@ngx-translate/core": "^12.1.2"
 }
```

To allow the dependency to be packaged with our library we must whitelist the dependency in the ng-package.json file as follows.

```
"whitelistedNonPeerDependencies": [
   "@ngx-translate/core"
 ]
```

Building, publishing and installing a library

Here I will explain how to build you're library and how to publish them to the ecare devops server.

Build library

Make sure you are in the folder of the library that you want to publish. (../projects/name)
In the package.json of the library change the version to the correct version as you cannot publish the same version twice.

Once you are ready to build the library execute the command “ng build (name) --prod”.

Publish library

When you have build the library correct there will be a new folder called ../dist/(name), open the terminal here.
Now make a new file in the library folder called “.npmrc” with the following contents.

```
registry=https://pkgs.dev.azure.com/ecaredevops/Giant/_packaging/E.care/npm/registry/
always-auth=true
```

Now that you added the file you need to get credentials in order to be allowed to publish the library to the server. Execute the following command to get new credentials
“vsts-npm-auth -config .npmrc”.

If you have received the credentials you can execute the the following command to publish the library “npm publish”

Installing library

Add the same .npmrc file to the root of you're project and get credentials so you are allowed to install the library from the server.

If you wish to install a library execute the following command “”npm install name@0.0.0”

Error Solving

Attention!

Some libraries will not build unless you load them dynamically. In order to do so you will have to add “//dynamic” above @NgModule in the module file.

When importing a module, service or component make sure you always import one item per line and always write the full path to the item. (else may cause library compile issues)

Remove unused imports ngc tries to connect everything this may cause build errors if you do not remove them.

ERROR: Resource busy or locked restart the editor. If you have published the library once you cannot rebuild the library until you restarted the editor.

ERROR: Function expressions are not supported in decorators Consider changing the function expression into an exported function.
Example:
BEFORE:

```
{
  provide: APP_INITIALIZER,
  multi: true,
  deps: [AppConfigService],
  useFactory: (appConfigService: AppConfigService) => {
    return () => {
      appConfigService.loadAppConfig();
    }
  }
```

AFTER:

```
{
  provide: APP_INITIALIZER,
  multi: true,
  deps: [AppConfigService],
  useFactory: LoadAppConfig
  }
export function LoadAppConfig(appConfigService: AppConfigService) {
  return () => appConfigService.loadAppConfig();
}
```

```
{
  "$schema": "../../node_modules/ng-packagr/ng-package.schema.json",
  "dest": "../../dist/ecarelib",
  "lib": {
    "entryFile": "src/public-api.ts"
  },
  "whitelistedNonPeerDependencies": [
    "@ngx-translate/core",
    "@ngx-translate/http-loader"
  ]
}
```