Angular Libraries via ng-packagr in CLI v6

Vancouver Angular Meetup - May 11, 2018

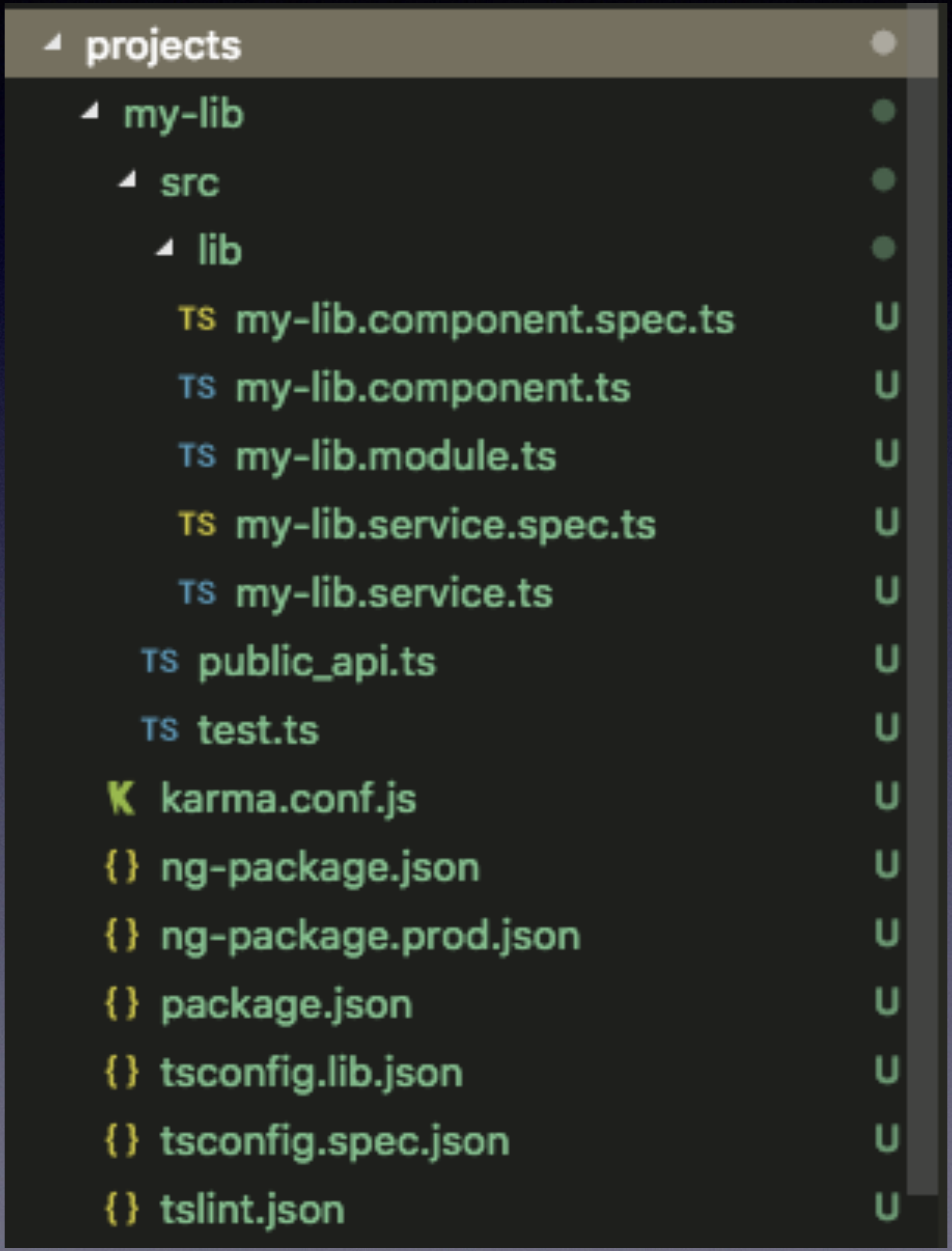
Libraries and NPM

- Not an alternative to NPM (or is it?)
- Easier to share and maintain?

Generate library

- ng generate library my-lib
    - Goes into projects/my-lib folder.
- ng build my-lib
- ng build my-lib --prod
    - Goes into dist/my-lib folder.

Structure

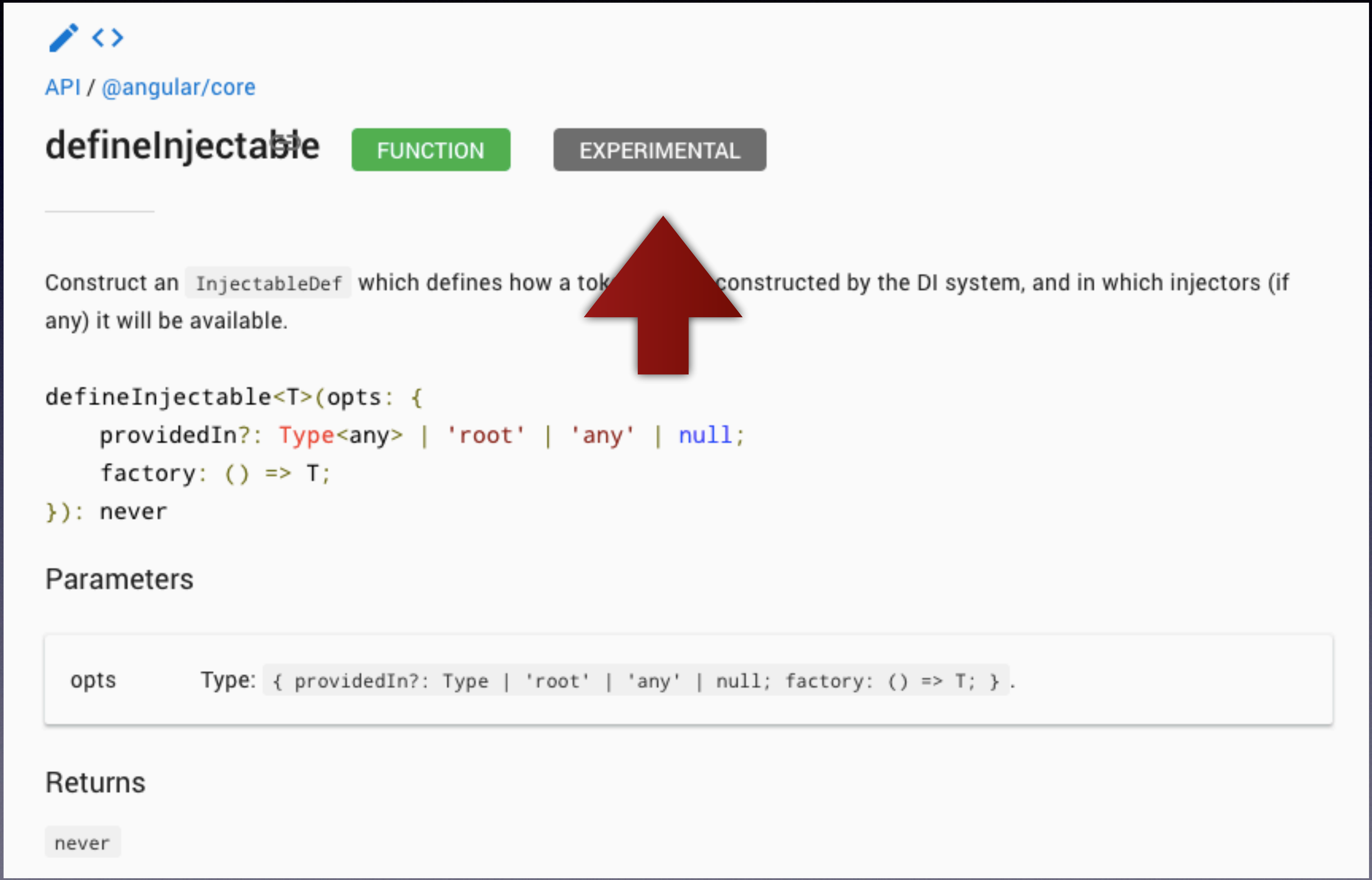


```
projects
  my-lib
    src
      lib
        my-lib.component.spec.ts      U
        my-lib.component.ts           U
        my-lib.module.ts              U
        my-lib.service.spec.ts        U
        my-lib.service.ts             U
      public_api.ts                   U
      test.ts                         U
    karma.conf.js                     U
    ng-package.json                   U
    ng-package.prod.json              U
    package.json                      U
    tsconfig.lib.json                 U
    tsconfig.spec.json                U
    tslint.json                       U
```

Import your library

- Angular 6 is required to import
- import { something } from 'library-name';

More experimental



API / @angular/core

defineInjectable FUNCTION EXPERIMENTAL

Construct an `InjectableDef` which defines how a tok constructed by the DI system, and in which injectors (if any) it will be available.

```
defineInjectable<T>(opts: {
    providedIn?: Type<any> | 'root' | 'any' | null;
    factory: () => T;
}): never
```

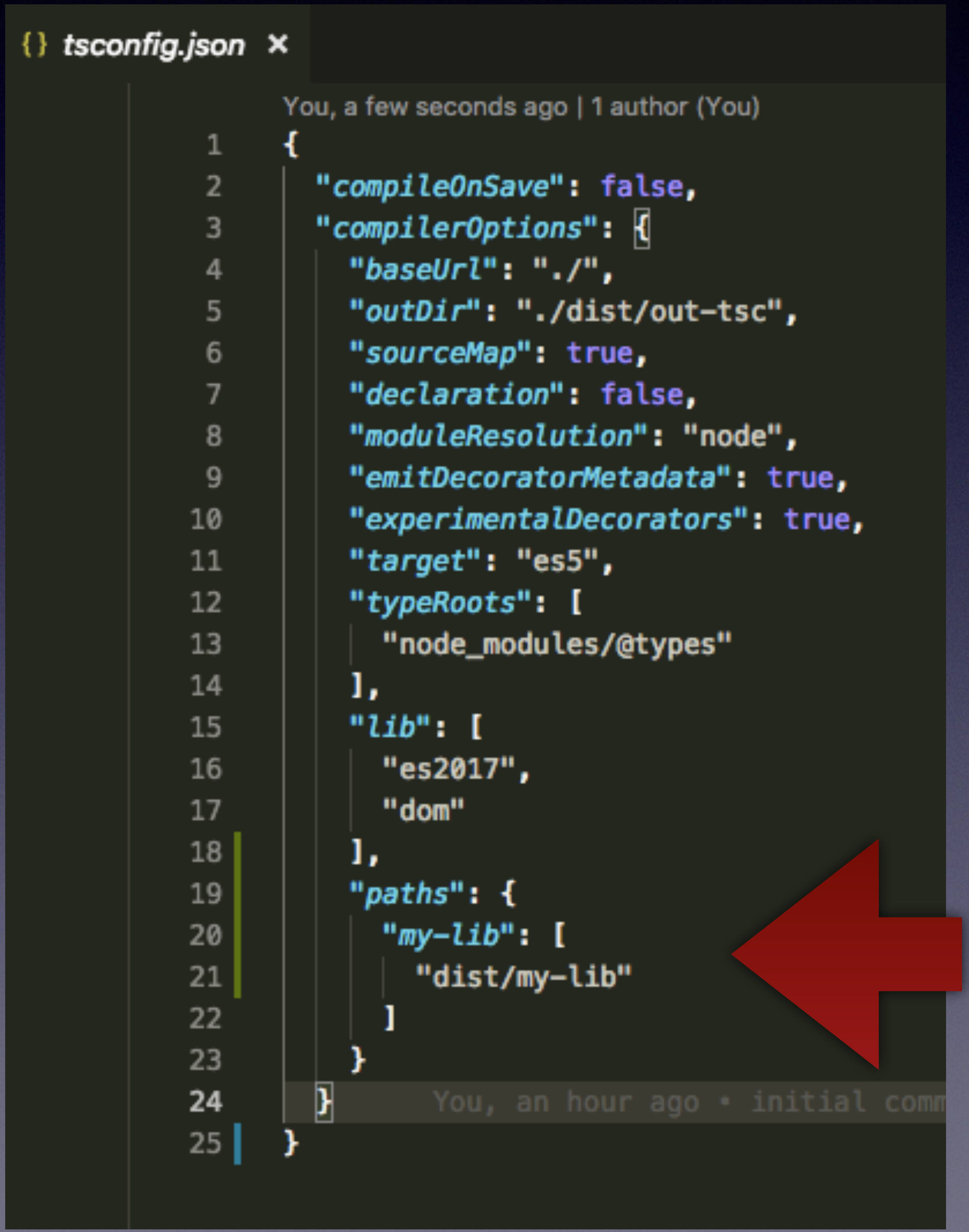
Parameters

opts	Type: `{ providedIn?: Type | 'root' | 'any' | null; factory: () => T; }`.

Returns

`never`

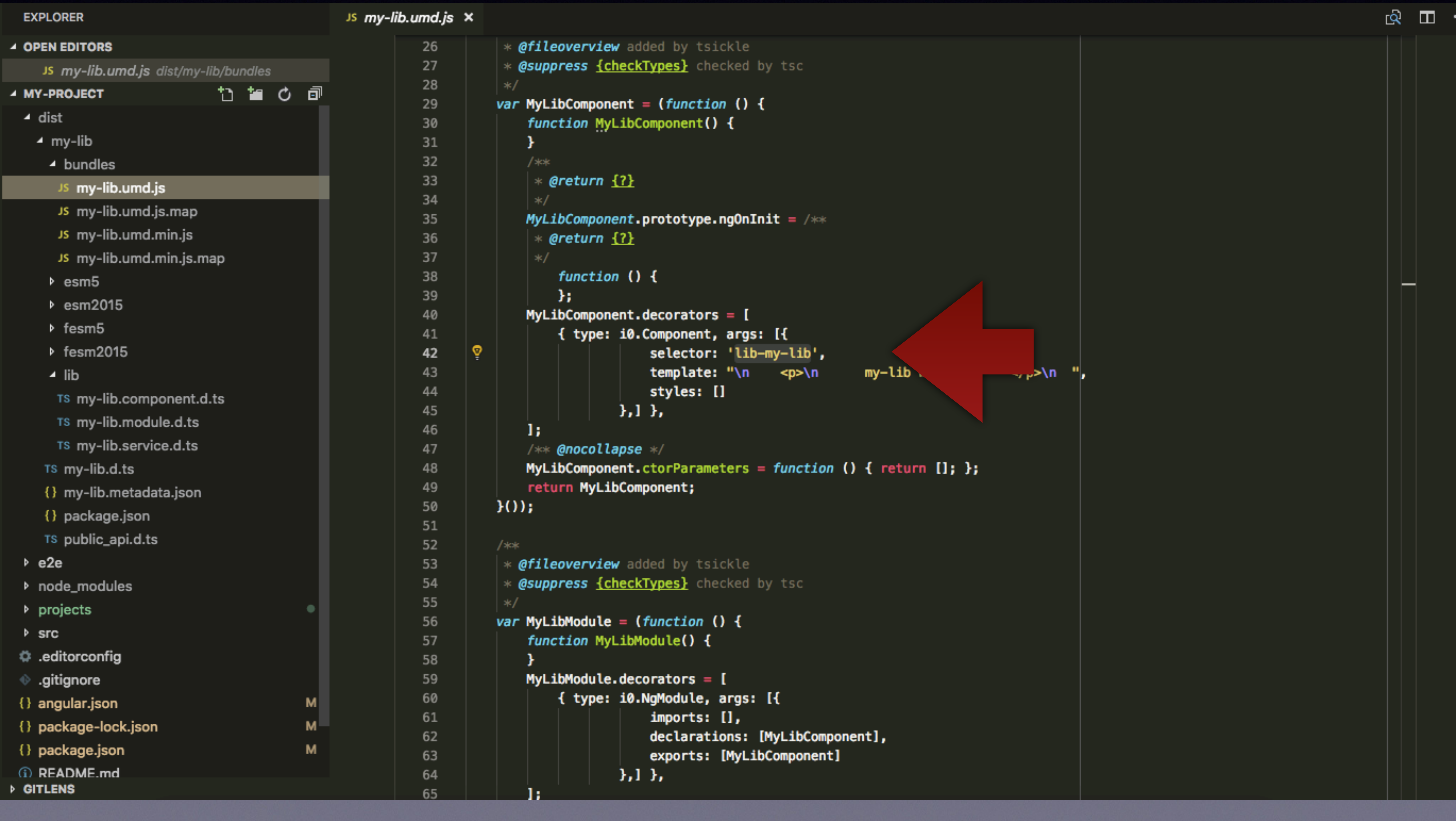
Using CLI build



```
{
  "compileOnSave": false,
  "compilerOptions": {
    "baseUrl": "./",
    "outDir": "./dist/out-tsc",
    "sourceMap": true,
    "declaration": false,
    "moduleResolution": "node",
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "target": "es5",
    "typeRoots": [
      "node_modules/@types"
    ],
    "lib": [
      "es2017",
      "dom"
    ],
    "paths": {
      "my-lib": [
        "dist/my-lib"
      ]
    }
  }        You, an hour ago • initial comm
}
```

you can also put it in node_modules as a hack

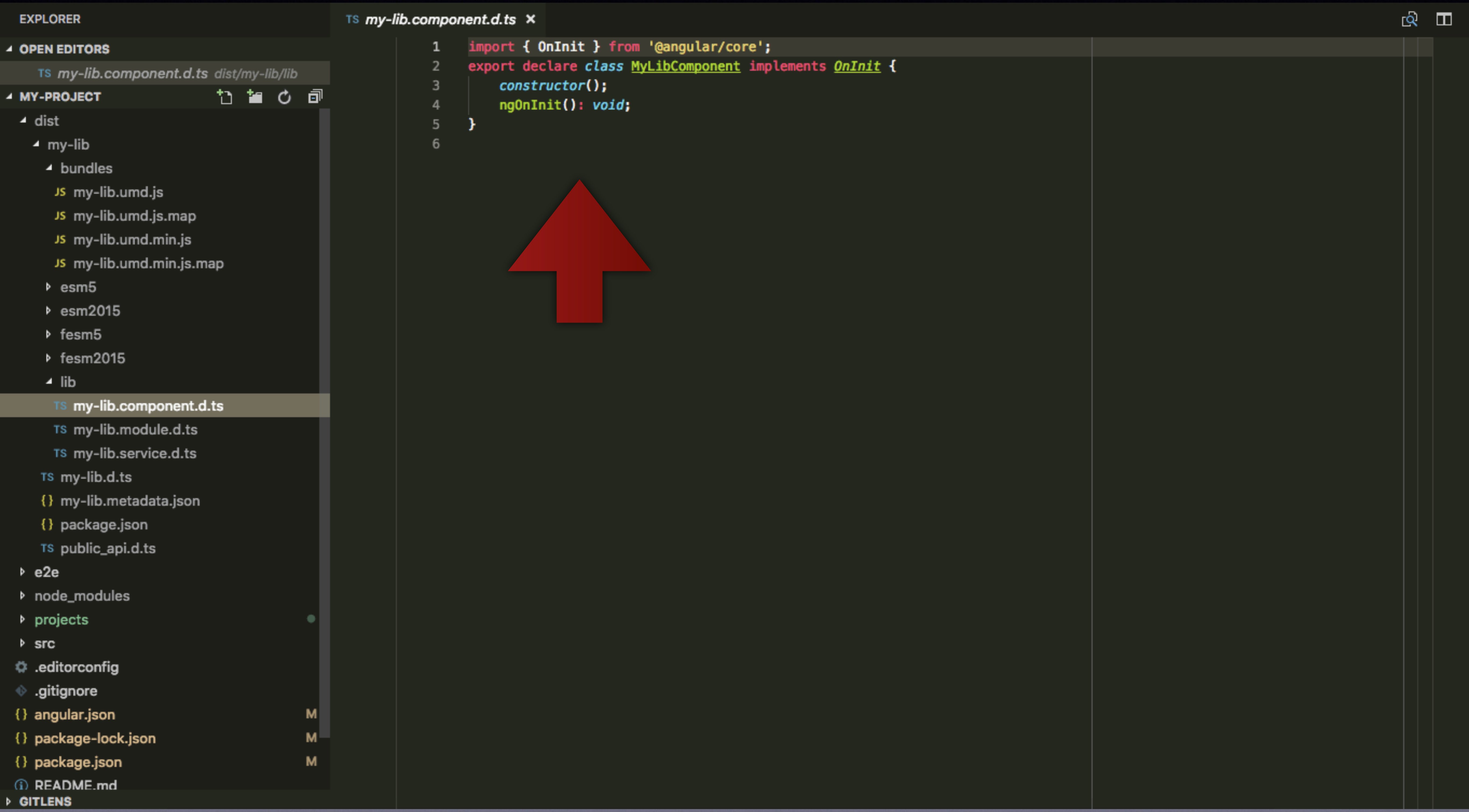
my-lib.umd.js



```
 * @fileoverview added by tsickle
 * @suppress {checkTypes} checked by tsc
 */
var MyLibComponent = (function () {
    function MyLibComponent() {
    }
    /**
     * @return {?}
     */
    MyLibComponent.prototype.ngOnInit = /**
     * @return {?}
     */
        function () {
        };
    MyLibComponent.decorators = [
        { type: i0.Component, args: [{
                    selector: 'lib-my-lib',
                    template: "\n    <p>\n      my-lib                 </p>\n  ",
                    styles: []
                },] },
    ];
    /** @nocollapse */
    MyLibComponent.ctorParameters = function () { return []; };
    return MyLibComponent;
}());

/**
 * @fileoverview added by tsickle
 * @suppress {checkTypes} checked by tsc
 */
var MyLibModule = (function () {
    function MyLibModule() {
    }
    MyLibModule.decorators = [
        { type: i0.NgModule, args: [{
                    imports: [],
                    declarations: [MyLibComponent],
                    exports: [MyLibComponent]
                },] },
    ];
```

my-lib.component.d.ts



```
import { OnInit } from '@angular/core';
export declare class MyLibComponent implements OnInit {
    constructor();
    ngOnInit(): void;
}
```