

DI Providers

Examples and utilities

DI PROVIDERS

providers: [LanguageService]

```
{ provide: LanguageService, useClass: LanguageService }
    DITOKEN PROVIDER
```

TOKEN: key for locating a dependency value

PROVIDER: definition object to create dependency value

DI TOKEN

Class type

```
{ provide: LanguageService, useClass: LanguageService }
```

• string

```
{ provide: 'api', useValue: 'http://myapp/api' }
```

InjectionToken

```
{
  provide: HTTP_INTERCEPTORS,
  useClass: CustomHttpInterceptor,
  multi: true,
},
```

PROVIDERS

- useValue, use native value, object
- useClass, creates a new dependencies
- useExisting, reuses an existing dependencies
- useFactory, function to create a dependencies based on runtime values/instances

useValue

Configuration

```
APP_CONFIG = new InjectionToken<MyConfig>('config');
{
   provide: APP_CONFIG,
   useValue: importedConfig,
}
```

Testing: mock, stubs, spies

```
const translateServiceStub = {
  instant: (label: string): string => {
    return prefixes[label];
  },
};

{ provide: TranslateService, useValue: translateServiceStub }
```

useClass

Use testing

{provide: TranslateService, useClass: TranslateServiceStub}

Specialization/Overrides behavior

CustomDatepickerI18n

Configuration

WebpackTranslateLoader

• Defines a dependencies for a fake service in lib

CustomI18nService extends LibI18nService

useExisting

• Fake service in library, defining in app with already used service

DateUtils extends LibDateUtils

useFactory

Setup Angular based on own service

useFactory: LocaleFactory

Fake service in library with default dependency

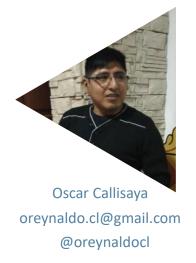
useFactory: LibI18nServiceFactory

Resources

- https://angular.io/guide/dependency-injection-providers
- https://medium.com/swlh/angular-dependency-providers-60a0400f370

Repository

https://github.com/oreynaldocl/di-examples



THANK YOU