

# Traducción con dropdown de idiomas

## Servicio

Servicio nativo para angular que permite multilinguaje de manera sencilla y adicionalmente, el dropdown de lenguaje para Bosch.

## Instrucciones

### Implementación del servicio multilinguaje

A. Instalación de los módulos necesarios

- **npm install @ngx-translate/core --save**
- **npm install @ngx-translate/http-loader --save**

B. Crear la función del loader en app.module.ts, al lado de la importación de las librerías

```
/* Librerías para las traducciones */ import { HttpClient,
HttpClientModule } from '@angular/common/http'; import {
TranslateLoader, TranslateModule} from '@ngx-translate/core'; import
{ TranslateHttpLoader} from '@ngx-translate/http-loader';
/*Carga el archivo de las traducciones para el loader de ngx-translate*/
export function createTranslateLoader(http:HttpClient)
{
return new TranslateHttpLoader(http,'./assets/translations/i18n/', '.json');
}
```

C. Añadir el modulo en los imports de app.module.ts

```
Imports:[
  TranslateModule.forRoot(
    loader:{
      provide:TranslateLoader,
      useFactory:(createTranslateLoader),
      deps:[HttpClient]
    }
  ) ...
]
```

**Pd:** Si el routing esta modulado, entonces hay que añadir solamente “TranslateModule” en los imports de los modulos hijos.

D. Declararlo en app.component.ts con el lenguaje default

```
import {TranslateService} from '@ngx-
translate/core';
export class AppComponent
{
  constructor(translate: TranslateService)
  {
    translate.setDefaultLang('en');
    translate.use('en');
  }
}
```

## Implementación del dropdown de Bosch con el multilinguaje

E. Añadir los atributos necesarios en el app.component.ts

```
//Atributos para el dropdown de lenguaje dropdownLanguages: any[] = [ { id:
"en", name: "English" },
{ id: "es", name: "Español" }
];
dropdownLanguageSelected:any;
```

F. Añadir el componente en app.component.html con los bindings correspondientes

```
<rb-language-selector [languages]="dropdownLanguages"
[(language)]="dropdownLanguageSelected" (languageChange)="changeLang($event)">
</rb-language-selector>
```

G. Iniciar lo asociado con el servicio de lenguaje en el constructor de app.component.ts

```
import {TranslateService} from '@ngx-translate/core';
... constructor(public
translateService:TranslateService,private){
    //inicializa el lenguaje    let
    language=this.translateService.getBrowserLang(); //Lenguaje default
    this.dropdownLanguageSelected=language;
    this.translateService.setDefaultLang(language);
    this.translateService.use(language);        this.setAvailableLangs();
    }
}
```

H. Añadir el método que añade los lenguajes disponibles para el servicio de traducción

```
//Set the langs in the translate service from the
array setAvailableLangs(){ let langs=[];
for (let i=0; i<this.dropdownLanguages.length;++i)
{ let target =
this.dropdownLanguages[i];
langs.push(target.id);
}
this.translateService.addLangs(langs);
}
```

I. Añade el método que se dispara cuando cambia un lenguaje en app.component.ts

```
//Cada vez que cambia de lenguaje  changeLang(event:
string) {    this.translateService.use(event)
}
```

## Continuación del multilenguaje

J. Crear los archivos .json dentro de la carpeta assets/i18n con los lenguaje de interés

- es.json
- en.json

## Uso

### En HTML

1. Añadir el binding en el .html con el id de la traducción

```
<div>{{ 'KEY' | translate }}</div>
```

2. Añadir las traducción con el id en los .json que creamos

- es.json

```
{
  "KEY": "¡Bienvenido!"
}
```

- en.json

```
{
  "KEY": "Welcome!"
}
```

## En Typescript

Es importante entender que el servicio multilenguaje hace peticiones http internas, es por esto que se esperar al dato que este nos devuelve.

1. Iniciar el servicio en el constructor y añadir la traducción cuando se usa mediante un await

```
let {KEY} = await this.translateService.instant(["KEY"] {});  
console.log(KEY);
```

2. Añadir las traducción con el id en los .json que creamos

- es.json

```
{  
  "KEY": "¡Bienvenido!"  
}
```

- en.json

```
{  
  "KEY": "Welcome!"  
}
```

**Pd:** Se puede añadir parámetros o incluso tags HTML en las traducciones, para esto revisar la fuente de referencia.

## Vista del servicio



## Repositorio de ejemplo

[https://sourcecode.socialcoding.bosch.com/projects/TSP/repos/cr\\_tools\\_otrs/browse](https://sourcecode.socialcoding.bosch.com/projects/TSP/repos/cr_tools_otrs/browse)

## Referencia

<https://www.npmjs.com/package/@ngx-translate/core>