



ANGULAR

Traducción i18n

Alexandru Kovacs

Contenido

1. Instalamos dependencias necesarias	2
2. Configuramos las variables de entorno.....	2
3. Preparamos la configuración de la aplicación	2
4. Creamos los archivos JSON de traducciones	4
5. Usamos las traducciones en el componente.....	5
6. Resultado.....	5
7. Bibliografía.....	5

1. Instalamos dependencias necesarias

Instalamos el compilador de traducciones:

```
npm install @ngx-translate/core
```

Instalamos el loader de archivos de traducciones:

```
npm install @ngx-translate/http-loader
```

Instalamos el compilador de pluralizaciones y formatos:

```
npm install ngx-translate-messageformat-compiler @messageformat/core
```

2. Configuramos las variables de entorno



3. Preparamos la configuración de la aplicación



●●● app.component.ts

```
1  import { Component } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3  import { RouterOutlet } from '@angular/router';
4
5  // Import the language files
6  import englishLanguage from '../assets/i18n/en.json';
7  import greekLanguage from '../assets/i18n/gr.json';
8
9  import { TranslateModule, TranslateService } from '@ngx-translate/core';
10
11  @Component({
12    selector: 'app-root',
13    standalone: true,
14    imports: [CommonModule, RouterOutlet, TranslateModule],
15    templateUrl: './app.component.html',
16    styleUrls: ['./app.component.css']
17  })
18  export class AppComponent {
19
20    constructor(private translate: TranslateService) {
21      this.initLanguage();
22    }
23
24    initLanguage(): void {
25
26      const default_lang = sessionStorage.getItem('default_lang');
27
28      this.translate.setTranslation('en', englishLanguage);
29      this.translate.setTranslation('gr', greekLanguage);
30
31      if (default_lang) {
32        this.translate.setDefaultLang(default_lang);
33        this.translate.use(default_lang);
34      } else {
35        this.translate.setDefaultLang('en');
36        this.translate.use('en');
37      }
38    }
39
40    useLanguage(language: string): void {
41      this.translate.use(language);
42      sessionStorage.setItem('default_lang', language);
43    }
44  }
45
```

Snipped

4. Creamos los archivos JSON de traducciones

Los archivos de las traducciones son en formato JSON y se deberán crear en la ruta **src/assets/i18n**

```
enjson
1 {
2   "demo": {
3     "greeting": "Hello {name}!",
4     "text": "This is a simple demonstration app for ngx-translate",
5     "title": "Translation demo"
6   },
7   "icu": {
8     "pluralization": "There {count, plural, =0{is no apple} one{is one apple} other{there are several apples}}.",
9     "select": "{gender, select, male{His uses} female{She uses} other{They use }} {product}"
10  },
11  "messagebox": {
12    "warning": {
13      "text": "Warning!"
14    }
15  }
16 }
17
```

Snipped

```
grjson
1 {
2   "demo": {
3     "greeting": "Γεια σου {name}!",
4     "text": "Αυτή είναι μια απλή εφαρμογή επίδειξης για το ngx-translate",
5     "title": "Επίδειξη μετάφρασης"
6   },
7   "icu": {
8     "pluralization": "Υπάρχει {count, plural, =0{δεν υπάρχει μήλο} one{ένα μήλο} other{υπάρχουν αρκετά μήλα}}.",
9     "select": "{gender, select, male{Ο χρήστης} female{Η χρήστρια} other{Χρησιμοποιούν }} {product}"
10  },
11  "messagebox": {
12    "warning": {
13      "text": "Προειδοποίηση!"
14    }
15  }
16 }
17
```

Snipped

Importante añadir la siguiente configuración en el archivo ts.config.json para poder cargar archivos JSON directamente desde el componente:

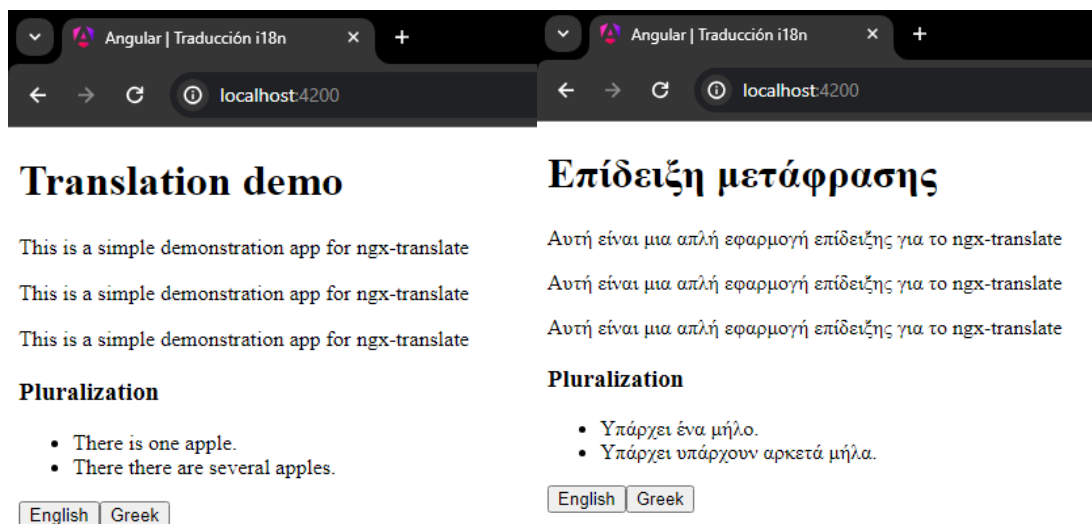
```
"resolveJsonModule": true
```

5. Usamos las traducciones en el componente

```
app.component.html

1 <div>
2   <h1>{{ 'demo.title' | translate }}</h1>
3
4   <!-- translation: translation pipe -->
5   <p>{{ 'demo.text' | translate }}</p>
6
7   <!-- translation: directive (key as attribute)-->
8   <p [translate]='demo.text'></p>
9
10  <!-- translation: directive (key as content of element) -->
11  <p translate>demo.text</p>
12
13  <h3>Pluralization</h3>
14  <ul>
15    <li translate="" [translateParams]='{ count: 1 }'>icu.pluralization</li>
16    <li>{{ 'icu.pluralization' | translate:{ 'count': 2 } }}</li>
17  </ul>
18
19 </div>
20
21 <!-- Selecciona el idioma -->
22 <button (click)="useLanguage('en')>English</button>
23 <button (click)="useLanguage('gr')>Greek</button>
24
```

6. Resultado



Demo en vivo 1: [HERE](#)

Demo en vivo 2: [HERE](#)

7. Bibliografía

Enlace 1: [How to translate your Angular app with ngx-translate](#)