i18n avec Angular

Préparation du projet

1. Création du projet

```
ng new i18n-project-example
> routing: no
> stylesheet: CSS
```

2. Pointer le terminal dans le répertoire du projet

```
cd i18n-project-example
```

3. Ajouter la libraire de gestion des d'internationalisation d'Angular

```
ng add @angular/localize
```

> **Note**
>
> Les commandes `ng add` procèdent à une installation compléte (téléchargement + modification de la configuration du projet).

Ajout des messages originaux

Dans une vue HTML

Prenons le fichier `app.component.html` pour notre exemple.

```
<h1>{{ title }}</h1>
<div>Hello {{ name }}</div>
<ng-container>The dog is very funny</ng-container>
```

Dans un fichier TypeScript

Prenons le fichier `app.component.ts` pour notre exemple.

```
title:string = "Angular is awesome";
name:string = "John"
```

Préparer l'identification des éléments à internationaliser

Dans la vue HTML, avec la directive `i18n`

la directive `i18n` va permettre l'identification des éléments à internationaliser:

- pour l'extraction vers un fichier dedier aux messages
- pour l'application de l'internationalisation

```
<h1>{{ title }}</h1>
<div i18n>Hello {{ name }}</div>
<ng-container i18n>The dog is very funny</ng-container>
```

Dans un fichier TypeScript, avec le tag `$localize`

comme pour la directive `i18n` au HTML, le tag `$localize` va permettre l'identification des éléments à internationaliser:

- pour l'extraction vers un fichier dedier aux messages
- pour l'application de l'internationalisation

```
title:string = $localize`Angular is awesome`;
name:string = "John"
```

Ajouter du contexte, une description et un identifiant pour chaque message

Indentifiant *(id)*:
Lors de l'extraction, Angular génère - *sur la base SHA-1* - un identifiant por chaque message. Il est possible d'attibuer ses proprs identifiants afin de les rendre plus lisible.

Description *(desc)*:
Permet d'ajouter une information de description sur le message à internationaliser.
Cette information sera utile sur des logiciels de traduction.

Contexte *(context)*:
Permet de decrire le contexte dans leque le message est utilisé.
Cette information sera utile au traducteur en cas d'ambiguité sur les termes à choisir lors de la traduction.

Aperçu de la syntaxe dans une vue HTML :

```
<div i18n="context|desc@@id">message original</div>
```

Aperçu de la syntaxe dans un fichier TypeScript :

```
title:string = $localize`:context|desc@@id:message original`;
```

Application dans une vue HTML

```
<h1>{{ title }}</h1>
<div i18n="@@sayHello">Hello {{ name }}</div>
<ng-container i18n="@@funnyDog">The dog is very funny</ng-container>
```

Application dans un fichier TypeScript

```
title:string = $localize`:@@pageTitle:Angular is awesome`;
name:string = "John"
```

Extraction des messages

Préparation de la commande d'extraction

Ajout de la commande d'extraction des messages dans le fichier `package.json`

```
"scripts": {
"i18n:extract": "ng extract-i18n --output-path=src/locale --format=json"
},
```

- `--output-path=src/locale`
  Défnition du répertoire de destination des fichier de traduction

- `--format=json`
  Définition du format d'extraction (arb, json, xlf, xmb)
  https://angular.io/guide/i18n-common-translation-files#change-the-source-language-file-format

Execution de la commade d'extraction

```
npm run i18n:extract
```

Exemple de sortie `json`

fichier `src/locale/messages.json`

```
{
        "locale": "en",
        "translations": {
                "pageTitle": "Play with Angular",
                "sayHello": "Hello {$INTERPOLATION}",
                "funnyDog": "The dog is very funny"
```

```
        }
}
```

Internationaliser les messages

Créer les fichiers de traduction

Dupliquer le fichier des messages originaux vers les fichiers de traduction

```
cp src/locale/messages.json src/locale/messages.fr.json
cp src/locale/messages.json src/locale/messages.no.json
```

Modifier les fichiers de traduction

fichier `src/locale/messages.fr.json`

```
{
        "locale": "fr",
        "translations": {
                "pageTitle": "Jouez avec Angular",
                "sayHello": "Bonjour {$INTERPOLATION}",
                "funnyDog": "Le chien est très drôle"
        }
}
```

fichier `src/locale/messages.no.json`

```
{
        "locale": "no",
        "translations": {
                "pageTitle": "Lek med Angular",
                "sayHello": "Hallo {$INTERPOLATION}",
                "funnyDog": "Hunden er veldig morsom"
        }
}
```

Build de l'application internationalisée

Modifier le fichier `angular.json` pour

- définir l'emplacement des fichiers de traduction source
- définir la configuration du rendu de build (`ng build`)
- définir la configuration du rendu de developpement (`ng serve`)

Ajouter les definitions des traductions

```
{
        "projects": {
                "i18n-project-example": {
                        // ...
                        "i18n": {
                                "sourceLocale": "en-US",
                                "locales": {
                                        "fr": {
                                                "translation":
"src/locale/messages.fr.json",
                                                "baseHref": "fr/"
                                        },
                                        "no": {
                                                "translation":
"src/locale/messages.no.json",
                                                "baseHref": "no/"
                                        }
                                }
                        },
                        "architect": {
                                // ...
                        }
                }
        },
        // ...
}
```

Configuration des sorties `ng build`

```
{
        "projects": {
                "i18n-project-example": {
                        // ...
                        "architect": {
                                "build": {
                                        // ...
                                        "configurations": {
                                                "production": {
                                                        // ...
                                                },
                                                "fr": {
                                                        "localize":
["fr"],
                                                        "outputPath":
"dist/i18n-project-example-fr/",

"i18nMissingTranslation": "error"
                                                },
```

```
                                        "no": {
                                                "localize":
["no"],
                                                "outputPath":
"dist/i18n-project-example-no/",

"i18nMissingTranslation": "error"
                                        }
                                }
                        },
                        // ...
                }
        }
},
// ...
}
```

Configuration des sorties `ng serve`

```
{
        "projects": {
                "i18n-project-example": {
                        // ...
                        "architect": {
                                "serve": {
                                        // ...
                                        "configurations": {
                                                "production": {
                                                        "browserTarget":
"i18n-project-example:build:production"
                                                },
                                                "fr": {
                                                        "browserTarget":
"i18n-project-example:build:fr"
                                                },
                                                "es": {
                                                        "browserTarget":
"i18n-project-example:build:es"
                                                }
                                        }
                                },
                                // ...
                        }
                }
        },
        // ...
}
```

Ajouter les commandes `ng build` et `ng serve` internationalisées

Dans le fichier `package.json`

```
"scripts": {
        "start": "ng serve",
        "start:fr": "ng serve --configuration=fr",
        "start:no": "ng serve --configuration=no",
        "build": "ng build",
        "build:fr": "ng build --configuration=fr",
        "build:no": "ng build --configuration=no",
},
```

Executer les tests (sur des porst different)

```
npm run start:fr -- --port=4201
npm run start:no -- --port=4202
```

Gestion de la pluralisation

Définition du mapping de pluralisation

dans le fichier `xxx.component.ts`

```
fruits: any[] = ['apple', 'banana'];

fruitsPlural: {[k: string]: string} = {
        '=0': $localize`:@@noFruit:No fruit`,
        '=1': $localize`:@@oneFruit:1 fruit`,
        'other': $localize`:@@manyFruit:# fruits`
};
```

Application de la pluralisation avec le pipe `i18nPlural`

dans le fichier `xxx.component.html`

```
<div>{{ fruits.length | i18nPlural: fruitsPlural }}</div>
```