

i18n Internationalisation avec Angular

Préparation du projet

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

Ajout de la librairie **localize**

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

Mise à jour de `package.json` et `polyfills.ts`

Modification du composant `app.component.html`

Ajout des messages

```
<div>Hello</div>  
<div>Yes</div>  
<div>No</div>  
<ng-container>Angular is awesome</ng-container>
```

Préparation à l'internationalisation

Ajout de la directive i18n sur les messages à internationaliser

```
<div i18n>Hello</div>  
<div i18n>Yes</div>  
<div i18n>No</div>  
<ng-container i18n>Angular is awesome</ng-container>
```

Extraction des messages

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` Définition du répertoire de destination

- `--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>

Création du fichier de messages originaux (en anglais par défaut)

```
npm run i18n:extract
```

Ajout d'attributs aux messages

Ajouter un identifiant

```
<div i18n="@messageHello">Hello</div>
<div i18n="@messageYes">Yes</div>
<div i18n="@messageNo">No</div>
<ng-container i18n="@messageAwesomeNg">Angular is awesome</ng-container>
```

Ajouter une description (xlf)

```
<div i18n="This is a message for Hello">Hello</div>
<div i18n="This is a message for Yes">Yes</div>
<div i18n="This is a message for No">No</div>
<ng-container i18n="This is a message for Awesome NG">Angular is
awesome</ng-container>
```

Ajouter un contexte donnant du sens (xlf)

```
<div i18n="Meaning value|This is a message for Hello">Hello</div>
<div i18n="Meaning value|This is a message for Yes">Yes</div>
<div i18n="Meaning value|This is a message for No">No</div>
<ng-container i18n="Meaning value|This is a message for Awesome
NG">Angular is awesome</ng-container>
```

Context, description et Id (xlf)

```
<div i18n="Meaning value|This is a message for
Hello@@messageHello">Hello</div>
<div i18n="Meaning value|This is a message for Yes@@messageYes">Yes</div>
<div i18n="Meaning value|This is a message for No@@messageNo">No</div>
<ng-container i18n="Meaning value|This is a message for Awesome
NG@@messageAwesomeNg">Angular is awesome</ng-container>
```

Création des fichiers de traduction

Création des fichiers de traduction

```
cp src/locale/messages.json src/locale/messages.fr.json
cp src/locale/messages.json src/locale/messages.es.json
```

Modifier les fichiers de traduction

version `json`

```
{
  "locale": "fr",
  "translations": {
    "messageHello": "Bonjour",
    "messageYes": "Oui",
    "messageNo": "Non",
    "messageAwesomeNg": "Angular est génial"
  }
}
```

version `xliff`

```
<?xml version="1.0" encoding="UTF-8" ?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
  <file source-language="fr" datatype="plaintext"
  original="ng2.template">
    <body>
      <trans-unit id="messageHello" datatype="html">
        <source>Hello</source>
        <target>Bonjour</target>
        <context-group purpose="location">
          <context context-
type="sourcefile">src/app/app.component.html</context>
          <context context-
type="linenumber">1</context>
        </context-group>
      </trans-unit>
      <trans-unit id="messageYes" datatype="html">
        <source>Yes</source>
        <target>Oui</target>
        <context-group purpose="location">
          <context context-
type="sourcefile">src/app/app.component.html</context>
          <context context-
type="linenumber">2</context>
        </context-group>
      </trans-unit>
      <trans-unit id="messageNo" datatype="html">
        <source>No</source>
        <target>Non</target>
        <context-group purpose="location">
```

```

                                <context context-
type="sourcefile">src/app/app.component.html</context>
                                <context context-
type="linenumber">3</context>
                                </context-group>
                                </trans-unit>
                                <trans-unit id="messageAwesomeNg" datatype="html">
                                    <source>Angular is awesome</source>
                                    <target>Angular est génial</target>
                                    <context-group purpose="location">
                                        <context context-
type="sourcefile">src/app/app.component.html</context>
                                        <context context-
type="linenumber">4</context>
                                    </context-group>
                                </trans-unit>
                                </body>
                                </file>
</xliff>

```

Compilation des traductions

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 développement (`ng serve`)

Ajouter les définitions des traductions

```

{
  "projects": {
    "i18n-project-example": {
      // ...
      "i18n": {
        "sourceLocale": "en-US",
        "locales": {
          "fr": {
            "translation":
"src/locale/messages.fr.json",
            "baseHref": ""
          },
          "es": {
            "translation":
"src/locale/messages.es.json",
            "baseHref": ""
          }
        }
      }
    }
  }
}

```

```

        },
        "architect": {
            // ...
        }
    },
    // ...
}

```

Ajouter la configuration des **builds**

```

{
    "projects": {
        "i18n-project-example": {
            // ...
            "architect": {
                "build": {
                    // ...
                    "configurations": {
                        "production": {
                            // ...
                        },
                        "fr": {
                            "localize":
["fr"],
                            "outputPath":
"dist/i18n-project-example-fr/",
                            "i18nMissingTranslation": "error"
                        },
                        "es": {
                            "localize":
["es"],
                            "outputPath":
"dist/i18n-project-example-es/",
                            "i18nMissingTranslation": "error"
                        }
                    }
                },
                // ...
            }
        },
        // ...
    },
    // ...
}

```

Ajouter la configuration du **live server**

```

{
  "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 de **build** et **serve**

Dans le fichier `package.json`

```

"scripts": {
  "start": "ng serve",
  "start:fr": "ng serve --configuration=fr",
  "start:es": "ng serve --configuration=es",
  "build": "ng build",
  "build:fr": "ng build --configuration=fr",
  "build:es": "ng build --configuration=es",
},

```

Executer les tests

```

npm run start:fr -- --port=4201
npm run start:es -- --port=4202

```

Internationaliser les messages d'un fichier TypeScript

dans le fichier `xxx.component.ts`

```
myAwesomeNgMessage = $localize`:@@messageAwesomeNg:`;
```

dans le fichier `xxx.component.html`

```
<div>{{ myAwesomeNgMessage }}</div>
```

La pluralisation

Définir le mapping des différentes possibilités

dans le fichier `xxx.component.ts`

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

messageMapping: {[k: string]: string} = {
  '=0': 'No fruit.',
  '=1': 'One fruit.',
  'other': '# fruits.'
};
```

dans le fichier `xxx.component.html`

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