

ANKIT SHARMA PRESENTS

Internationalization **in** **Angular**



Ankit Sharma

- SDE III at Cisco Systems, India
- Author & Speaker
- GDE for Angular, Microsoft MVP
- Blog <https://ankitsharmablogs.com/>



@ankitsharma_007

What is Internationalization?

- Internationalization is the process of making our app support various languages to extend the reach to a worldwide audience.
- Shorthand – i18n



What is Localization??

- Localization is the process for translating the app to a particular language.
- We need to apply internationalization to the application and after that we can localize it.
- Localization allows us to serve our application in different languages.
- Shorthand – `l10n`



Add the localize package

- We will add the **@angular/localize** package to our project. It will allow us to use the localization features provided by Angular.

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



Add the localize package

- This command will do the following
 - install the package in your angular app.
 - update the **polyfills.ts** file which allows the project to take advantage of Angular's localization features.



The i18n Attribute

- The Angular framework provides us with the i18n attribute, which can be used to mark an HTML element for translation.

```
<h1 i18n>Hello World</h1>
```



The i18n Attribute

- We can add a description to the **i18n** attribute which will provide additional information or context for the translator.

```
<h1 i18n="An introduction header">Hello World</h1>
```



The i18n Attribute

- If you want to translate an attribute of an HTML tag, then you can use the i18n attribute as **i18n-x**, where x is the name of the HTML attribute you wish to translate.

```
<input i18n-placeholder placeholder="Name" name="name" />
```



Create a Translation Source File

- After preparing the template file for translation, we will extract the text marked for translation to a Translation Source File.

```
ng extract-i18n --output-path src/translate
```



Create a Translation Source File

- It will create a folder called `translate` inside the `src` folder and create a `messages.xlf` file inside it.
- The `--output-path` option is used to set the output path for the generated translation file.



Serve App in Multiple Languages

- Make the following changes in the **angular.json** file
 - Configure the locale and the path of translated file.
 - Configuration for locale definition in the build configuration.
 - Custom locale-specific configuration using a single locale.



Execute in Local Env

- We can use only one locale at a time while executing the app in the local environment.
- Run the following command to execute the app in local environment for French language.

```
ng serve --configuration=fr
```

Demo Time



in/ankitsharma-007



@ankitsharma_007

Deploy to Firebase

- Deploy to Firebase to see how i18n works in the production environment.
- Run the following command to create the production build of the app.

```
ng build --localize
```



Resources

- GitHub: <https://github.com/AnkitSharma-007/angular-localization>
- Personal blog - <https://ankitsharmablogs.com>

