# Localization in Angular

# 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.

# What is Internationalization?

[Internationalization](https://en.wikipedia.org/wiki/Internationalization_and_localization) is the process of supporting multiple languages in your applications.

1. I18n , also known as internationalization, is the process of making our app support various languages
2. When the tag is marked as i18n then it is considered as translatable content

# 1: Ngx-translate

Ngx-translate [has lots of features](https://github.com/ngx-translate/core), including:

* Fast and simple integration process
* A built-in loader to fetch translation files from the given path
* Support for Ahead-of-Time compilation (which [requires a custom loader though](https://github.com/ngx-translate/core#aot)). [AoT compilation](https://angular.io/guide/aot-compiler" \t "_blank) simply means that all your TypeScript code and Angular-related HTML is turned to plain old JavaScript before the browser starts downloading the page. This approach, however, imposes a restriction: You need to serve separate apps for different locales
* Translation files are stored in JSON format, which is lightweight and easy to work with
* Support for interpolation and HTML markup inside translation messages
* Translations can be performed [using a service, a simple pipe, or a directive](https://github.com/ngx-translate/core#4-use-the-service-the-pipe-or-the-directive)
* Support for event listeners (for example, [an onLangChange event](https://github.com/ngx-translate/core#properties))
* Ability to define a [missing translation handler](https://github.com/ngx-translate/core#how-to-handle-missing-translations)
* [A handful of plugins](https://github.com/ngx-translate/core#plugins) including a special router, .po files loader, translations extractor, and some others

# 2: The Built-in I18n Module

* A special tool to extract strings into translation files (where the actual [translation](https://phrase.com/blog/localization-and-translation-glossary/translation-t9n/) takes place); note that this module works only with XML
* A special option to build an application for the given language. This enables support for the [Ahead of Time compilation](https://angular.io/guide/i18n#merge-with-the-aot-compiler)
  + *AOT AND JIT compiler IN Angular : (JIT downloads the compiler and compiles code exactly before Displaying in the browser. AOT has already complied with the code while building your application, so it doesn't have to compile at runtime. Loading in JIT is slower than the AOT because it needs to compile your application at runtime)*
* An ability to provide [context for translations](https://angular.io/guide/i18n#help-the-translator-with-a-description-and-meaning). This is very important for [translators](https://phrase.com/blog/posts/a-day-in-the-life-of-a-translator/) to adequately translate a given phrase into another language
* An ability to [mark HTML attributes for translation](https://angular.io/guide/i18n#translate-attributes)
* An ability to mix [pluralization and alternate text messages](https://angular.io/guide/i18n#regular-expressions-for-plurals-and-selections) (check [this example](https://angular.io/guide/i18n#nesting-plural-and-select-icu-expressions))

In i18n, we need to have three different serving locations for different languages.

## Script to compile the app for production

* npm run build-locale

(or)

* **"build-locale:en":** ng build --prod --i18n-locale en --i18n-format xlf --i18n-file src/translate/messages.en.xlf --output-path=dist/en --baseHref /en/
* **"build-locale:es":** ng build --prod --i18n-locale es --i18n-format xlf --i18n-file src/translate/messages.es.xlf --output-path=dist/es --baseHref /es/
* **"build-locale:hi":** ng build --prod --i18n-locale hi --i18n-format xlf --i18n-file src/translate/messages.hi.xlf --output-path=dist/hi --baseHref /hi/

## [How to run the Dist Folder on Local Machine in Angular](https://stackoverflow.com/questions/51129053/how-to-run-the-dist-folder-on-local-machine-in-angular-6)

* You can do that using [http-server](https://www.npmjs.com/package/http-server) package.
  + First install the package **globally**
    - npm install http-server -g
  + Then inside your project directory (in the terminal) just run
    - http-server dist/
* Now you can visit [http://localhost:8080](http://localhost:8080/) to view your application.

# node commands – i18n

npx @angular/cli new angular-internationalization-example --style=css --routing=false --skip-tests

cd angular-internationalization-example

npm start

ng xi18n:extract

cp src/locale/messages.xlf src/locale/messages.fr.xlf

cp src/locale/messages.xlf src/locale/messages.de.xlf

ng add @angular/localize

npm start

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

npm run start:de -- --port=4202

**References:**

https://www.freecodecamp.org/news/how-to-implement-localization-in-angular-using-i18n-tools-a88898b1a0d0/

<https://stackoverflow.com/questions/51129053/how-to-run-the-dist-folder-on-local-machine-in-angular-6>

https://www.codeandweb.com/babeledit/tutorials/how-to-translate-your-angular-app-with-ngx-translate