Steps involved in the process of translate an Ionic App

1. Translation Language
    a. Detect the language you want to translate into.
    b. It can either be done automatically by detecting phone or browser's language (Globalization)
    c. It can be done with user actions(dropwon, buttons,...)
    d. For this case, we will detect device's language using both:
        i. Cordova globalization plugin
        ii. Browser's internationalization API
2. Prepare Language Text
    a. We need to have a pre-translated dictionary (Json file) which stores the translations
    b. We can use this url to translate our json files
        i. https://www.logisticinfotech.com/translate-language-files-online-json/
3. Translate
    a. We will use ngx-translate library for translating our texts

Development

1. **Create a starter Ionic 5 tab app**
    a. ionic start ionicTranslate tabs --type=angular --cordova
        i. with --cordova we tell the CLI to integrate cordova in the app

2. **Prepare multiple language Json files in assets**
    a. We will create these Json files in "src/assets/i18n" folder
    b. Note for the json files for the example:
        i. The {{ value }} and {{ name_value }} are kind of variable/constants we can pass from our component.

3. **Implement ngx-transtale library and Cordova Globalization Plugin**
    a. Cordova plugin is used to detect device's default language/locale, Unfortunately, this plugin is deprecated, but still supported by Ionic
    b. However, the latest way of detecting the language / locale of the browser is by using browser's default internationalization API
        i. https://cordova.apache.org/news/2017/11/20/migrate-from-cordova-globalization-plugin.html
    c. We install Cordova globalization plugin using
        i. ionic cordova plugin add cordova-plugin-globalization
        ii. npm install @ionic-native/globalization
    d. We install ngx-translate library
        i. npm install --save @ngx-translate/core
        ii. npm install @ngx-translate/http-loader --save
            1. http-loader is used for loading the translation json files via Angular's HttpClient module.
    e. We need to define a function that loads the external json files to the app using http-loader. It is in the app.module.ts

        i. export function HttpLoaderFactory(http: HttpClient) { return new TranslateHttpLoader(http, "./assets/i18n/", ".json"); }
        ii. Pay attention to TranslateModule.forRoot() , is definded in case of a Tabbed aplicateion, or general non lazy-loaded module. For a tab child, however, we will have to use TranslateModule.forChild().
    f. Import and setup the translate library in child component.
        i. in “src/app/tab1”

4. **The Directive**
    a. to use the directive like this
        i. <h1 translate>TITLE</h1><p [translate]="'description'"></p>
    b. we need to import the translation module in child modules as well for everything to work correctly.
    c. It wll be in “tab1.module.ts”

5. **Setup stand alone translations**
    a. The process of setting up separate language files in “assets” for each language is the standard way of translation in Angular. But sometimes it becomes a little cumbersome, especially when don’t have that much data to translate
    b. Quick translation in tab2.page
        i. We can declare the variables in the component itself instead of reading from the JSON files from “assets”
        ii. tab2.html is similar to tab2.html and the same to tab2.scss and
        iii. We make changes in tab2.ts constructor

Source:
https://enappd.com/blog/how-to-translate-in-ionic-internationalization-and-localization/143/