

E2E tests

There are four different sets of e2e tests in this folder. They are all testing different translation scenarios, but they are all built with IVY enabled.

runtime

A new `polyfills.ts` file is provided (`polyfills-runtime.ts`) which is swapped in by a file replacement in the `angular.json` configuration. In this new file: * Runtime translations are provided (`loadTranslations()`). * The current locale is set (`$localize.locale = 'fr'`) and loaded (`registerLocaleData(localeFr);`)

de and fr

The application is built (into the `dist` folder) and then two sets of translations (`src/locales/messages.(de|fr).json`) are used to generate two copies of the app, which have been translated (compile-time inlined).

These translated apps are stored in `tmp/translations/(de|fr)`.

legacy

The CLI `ng extract-i18n` tool extracts the messages from the Angular templates, into the XLIFF 1.2 format with legacy message ids (`tmp/legacy-locales/messages.legacy.xlf`).

The translation file is modified to apply a simple translation.

The app must be compiled using the `i18nLegacyMessageIdFormat` option set to ensure that the correct message ids are used to match those in the translation files.

The app is translated using the compile-time inlining tool to generate a copy that has the translated message in it.

Hosting

Since the CLI hosts from and in-memory file-system the compile-time inliner is not able to translate the output files. So the `de`, `fr` and `legacy` apps must be statically built to disk and translated there.

Since the translated app is now on disk, we cannot use the CLI to serve it. Instead we use a simple static HTTP server instead.