

Localization, part 2

In Part 2 of the Localization unit, you are going to localize the Road Signs app for the `fr_FR` locale. Your app will display French descriptions for US signs and English descriptions for French signs. This practice will simulate an app meant for US tourists visiting France and French tourists visiting the US.

All localized resources should have the same names as their corresponding default resources so that Android can swap them as necessary depending on which locale is in use at the moment. Example: `yield.png`

You may find that some resources cannot be localized because there is no French equivalent. Delete those resources or simply do not display them in your localized version.



Figure 1: Beware dust storms!

Creating French resource folders

Create `drawables-fr/` and `values-fr/` folders in your app's `res/` folder.

You can create these folders either within Android Studio or within your operating system's file manager. Android Studio should be able to access them either way.

Duplicating drawables

Copy the `.png` graphic files for the US signs from `drawables/` to `drawables-fr/`.

When you choose a French locale for your device, your app should display French text describing US signs, as a French tourist to the US might need. The reverse should be true when you select the default locale, which is functionally the same as `en_US`. That is why US signs are saved in the `drawables-fr/` folder.

Duplicating strings

Copy the `strings.xml` file from `values/` to `values-fr/`.

You will replace the English-language contents of these strings with French text.

Localizing graphics

Replace the US road sign graphics in `drawables/` with corresponding French signs.

References

- [Comparison of European road signs - Wikipedia](#)
- [Road signs in France - Wikipedia](#)

Localizing text

Replace the English text in `values-fr/strings.xml` with the corresponding French text.

You can probably find appropriate road sign names wherever you found the French road sign images. However, you might need to use an online translation service for the user interface text if you don't read and write French.

Keep the *variable names* of the strings the same, so that Android can automatically replace the variable names in the user interface with French text when you choose a French locale.

References

- [Bing Translator](#)
- [Google Translate](#)

Further localization

Replace “USA” with “France” in English strings and add “USA” to the French strings.



Figure 2: French “No motorcycles” sign

Remember, the signs in `drawables/` will be French signs, and the signs in `drawables_fr` will be from the US. Your instructor will demonstrate this in case you need clarification.

Localizing the UI

Localize the name of the app and any built-in documentation, as well as controls such as buttons.

Once again, an online translation service might be useful here.

Test your app

Run your app in Android Studio’s emulator or on your own device. Demonstrate it for a friend with both the default or `en_US` locale and the `fr_FR` locale. How does it look? Do your translations and functional localizations make sense?

Questions

How would you localize this app for more than one language or locale?

Multi-language localization would depend in part on whether it were one-to-many (English to French, Spanish, Mandarin, and so on) or many-to-many (English to French and Spanish, Spanish to English and French, and French to English and Spanish, for example).