

Lab

Create a simple app that displays a country name and an associated image that are specific to a language. The image will launch google maps to locate the country when clicked. Localise it for 5 languages/countries (i.e. when it runs it displays the image and text associated with the locale without programmer intervention.) There will be simple coding for this, it is mostly done in res(ource) directories.

1. Create a simple app UI that has a TextView and ImageButton.
2. Populate default directories (res/values/ and res/drawable/ no localisation yet) with a country name string in values/strings.xml and a country image in drawable/. Your app may crash if you do not have a default on device/AVD start up. Your devices are factory set to English US (en-rUS.) Choose an english speaking country.
For example you could use a map of Canada in the drawable directory named country.png and Canada in <string name="country">Canada</string> .
Or you could use a map of US in the drawable directory named country.png and USA in <string name="country">USA</string> . Use the name string and drawable in your layout xml.
3. Add code to have the button launch google maps showing a the country associated with the image, see <https://developer.android.com/guide/components/intents-common.html#Maps>
You will need to add the intent filter to the manifest and to use Uri.parse or Uri.builder to create your Uri. Keep it simple, use country name not lat/long.
4. Run your app at this point, to make sure it works.

5. English is now your default, so add other languages. You will use French and a French speaking country and choose four other languages. Check the languages on your device before choosing. Search the web for an image or a map and/or a flag or something representing the languages you choose from <http://openclipart.org/> or another site; be sure to download a png image.

"The language is defined by a two-letter [ISO 639-1](#) language code, optionally followed by a two letter [ISO 3166-1-alpha-2](#) region code (preceded by lowercase "r"). " (en-rCA, fr-rCA, en-rGB, etc) Set up a value directory, string.xml file and string country entry for each language.

See <http://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources>

6. For each alternate language create a res/drawable directory, import your images; remember your image must have the same name, different directory.
7. For each alternate language create a res/values directory, create a strings.xml for each with the appropriate text in the language.
8. When you run your app test each language by changing the locale on the device/AVD. Change the language either through the settings button in the launcher or the command line:
http://wiki.pcampbell.profweb.ca/index.php/Set_the_locale_in_Android

Once everything is working for each language modify your ImageButton so that the image fills the screen including when you rotate.

I have given you an example app <http://bit.ly/518droid> testMaps.apk this is the behaviour I expect.

Optional:

- Instead of going directly to google maps invoke another activity where you ask for a location (city or address) then invoke google maps for location + country.
- Use a different text colour for each language/country (find out the flag colours and use one.)
- Convert the png images to 9 Patch (<android sdk>/tools/draw9patch.bat, see what happens when you test on various screen resolutions.
- Add countries that have the same language different region code (en-rGB, en-rCa en-rUS) You will have to run some tests on an AVD as the SIM card is what defines the country AFAIK. If you discover differently SVP let me know. Or you may want to add further functionality to change the language and country programmatically.
- Have different layouts for landscape and portrait

Helpful (?) Material

- <http://developer.android.com/training/basics/supporting-devices/languages.html>
- <http://developer.android.com/guide/topics/resources/localization.html>
- <http://www.icanlocalize.com/site/tutorials/android-application-localization-tutorial/>
- My repo on github <https://github.com/orgs/Android518-2015> see week04-localisation & week04-resources