



# **Chapter – 12**

## **Localization**

# Localization



- Localization is the process of making an app available in multiple languages.
- To sell an application worldwide successfully, you must target it to a broad and diverse audience.
- English enjoys a broad acceptance and practice, and therefore you can expect that an English language application will sell well globally, but why give up on sales opportunities to the broader non-English-speaking Android market?
- In its simplest approach, this means translation into multiple languages. But there's more! Beyond language translation, an application needs to properly handle dates and times, number and currency formats, and for some applications unit of measure.
- The reasons for localizing an application are manifold. Your application may be bound for some cultural reasons to a specific region.

# String resource



- Keeping your labels and other bits of text outside the main source code of your application is generally considered to be a very good idea. [strings.xml]
- In particular, it helps with Internationalization (I18N) and localization (L10N).
- Even if you are not going to translate your strings to other languages, it is easier to make corrections if all the strings are in one spot instead of scattered throughout your source code.

# Android Way



- The way Android currently handles this is by having multiple resource directories, with the criteria for each embedded in their names.
- Suppose, for example, you want to support strings in both English and Spanish.
- Normally, for a single-language setup, you would put your strings in a file named `res/values/strings.xml`.
- To support both English and Spanish, you would create two folders, `res/values-en/` and `res/values-es/`, where the value after the hyphen is the ISO 639–1 two-letter code for the language you want.
- Your English-language strings would go in `res/values-en/strings.xml` and the Spanish ones in `res/values-es/strings.xml`. Android will choose the proper file based on the user's device settings.

# Android Way

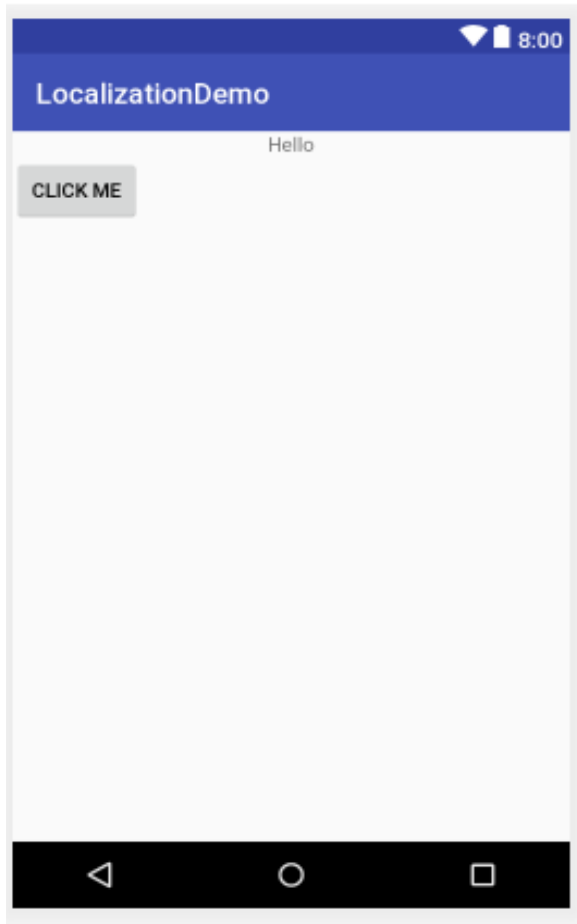


- An even better approach is for you to consider some language to be your default, and put those strings in `res/values/strings.xml`.
- Then, create other resource directories for your translations (e.g., `res/values-es/strings.xml` for Spanish).
- Android will try to match a specific language set of resources; failing that, it will fall back to the default of `res/values/strings.xml`.
- This way, if your app winds up on a device with a language that you do not expect, you at least serve up strings in your chosen default language.
- Otherwise, if there is no such default, you will wind up with a `ResourceNotFoundException`, and your application will crash.

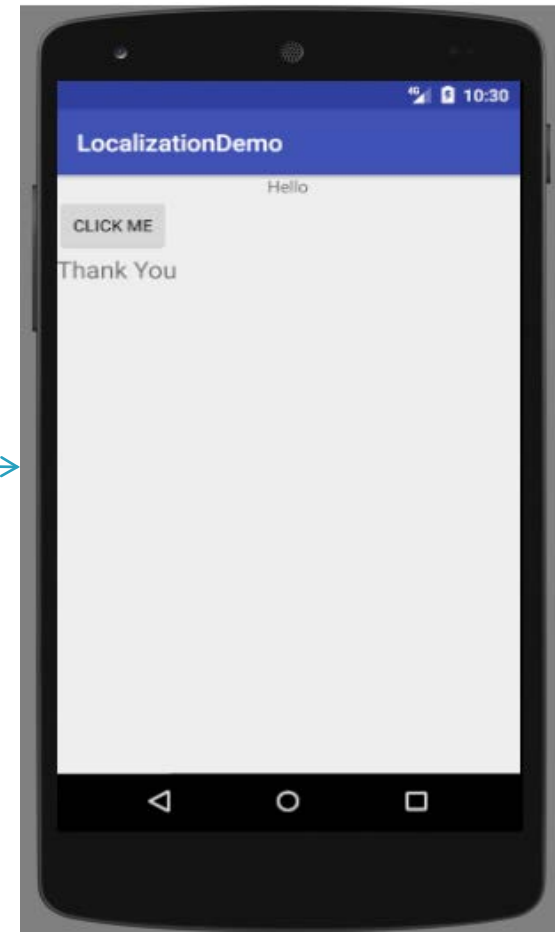
# Hands on Example(English)



This for the default Language(English), you have store everything in String Resources. To make localization here we use French.



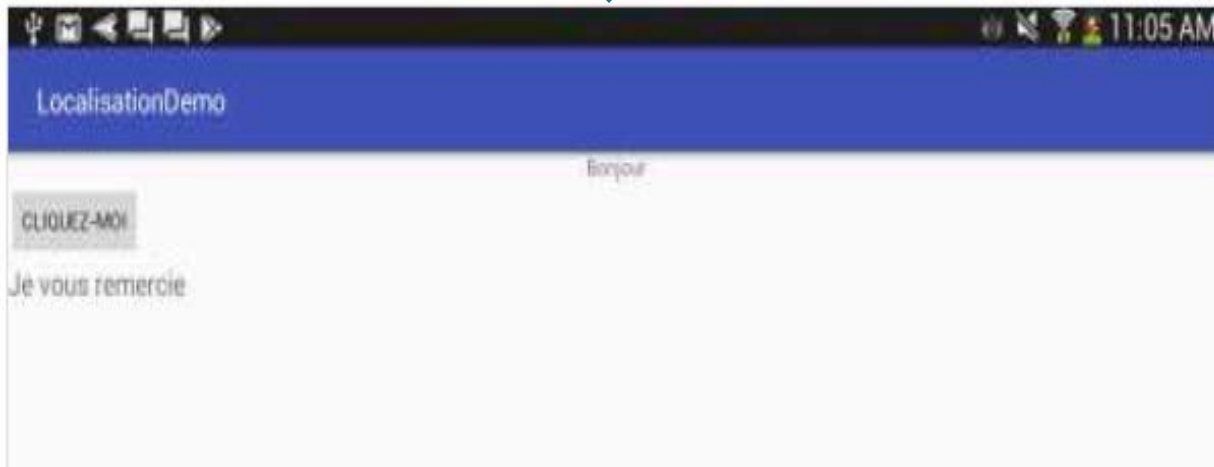
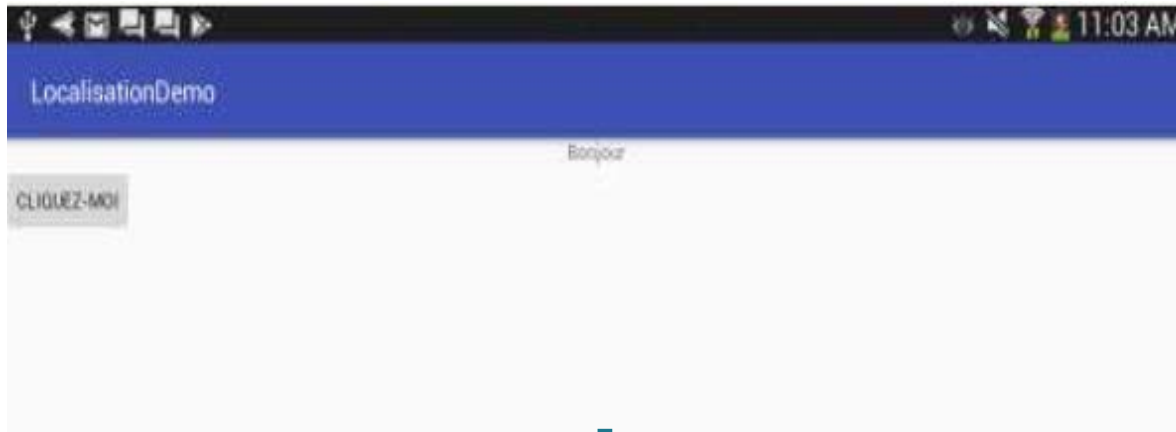
After CLICK ME



# Hands on Example(France)



Once you changed the Language settings to France, you will get the following output



# activity\_main.xml



LocalizationDemo app src main res layout activity\_main.xml

activity\_main.xml x MainActivity.java x values\strings.xml x gradle.properties x

1: Project  
2: Structure  
Captures  
Build Variants  
2: Favorites

app

- manifests
- java
  - com.example.rmohanraj
    - MainActivity
- res
  - drawable
  - layout
    - activity\_main.xml
  - mipmap
  - values
    - colors.xml (2)
    - strings.xml (2)
      - strings.xml
      - strings.xml (fr)
    - styles.xml (2)

Gradle Scripts

LinearLayout

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:orientation="vertical" >
6     <TextView
7         android:layout_width="fill_parent"
8         android:layout_height="wrap_content"
9         android:gravity="center"
10        android:text="Hello" />
11    <Button
12        android:id="@+id/button1"
13        android:layout_width="wrap_content"
14        android:layout_height="wrap_content"
15        android:onClick="displayResponse"
16        android:text="Click Me" />
17    <TextView
18        android:id="@+id/textView1"
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:text=""
22        android:textSize="20sp" />
23 </LinearLayout>
```

Design Text

Logcat Android Profiler Run TODO Terminal Messages

LocalizationDemo

8:00

Hello

CLICK ME



# String Resources



## **strings.xml**

```
<resources>
  <string name="app_name">LocalizationDemo</string>
  <string name="title">Localization</string>
  <string name="hello">Hello </string>
  <string name="button"> Click Me</string>
  <string name="response">Thank You</string>
</resources>
```

## **strings.xml(fr)**

```
<resources>
  <string name="app_name">LocalisationDemo</string>
  <string name="title">Localisation</string>
  <string name="hello">Bonjour </string>
  <string name="button"> Cliquez-moi</string>
  <string name="response">Je vous remercie</string>
</resources>
```

# MainActivity.java

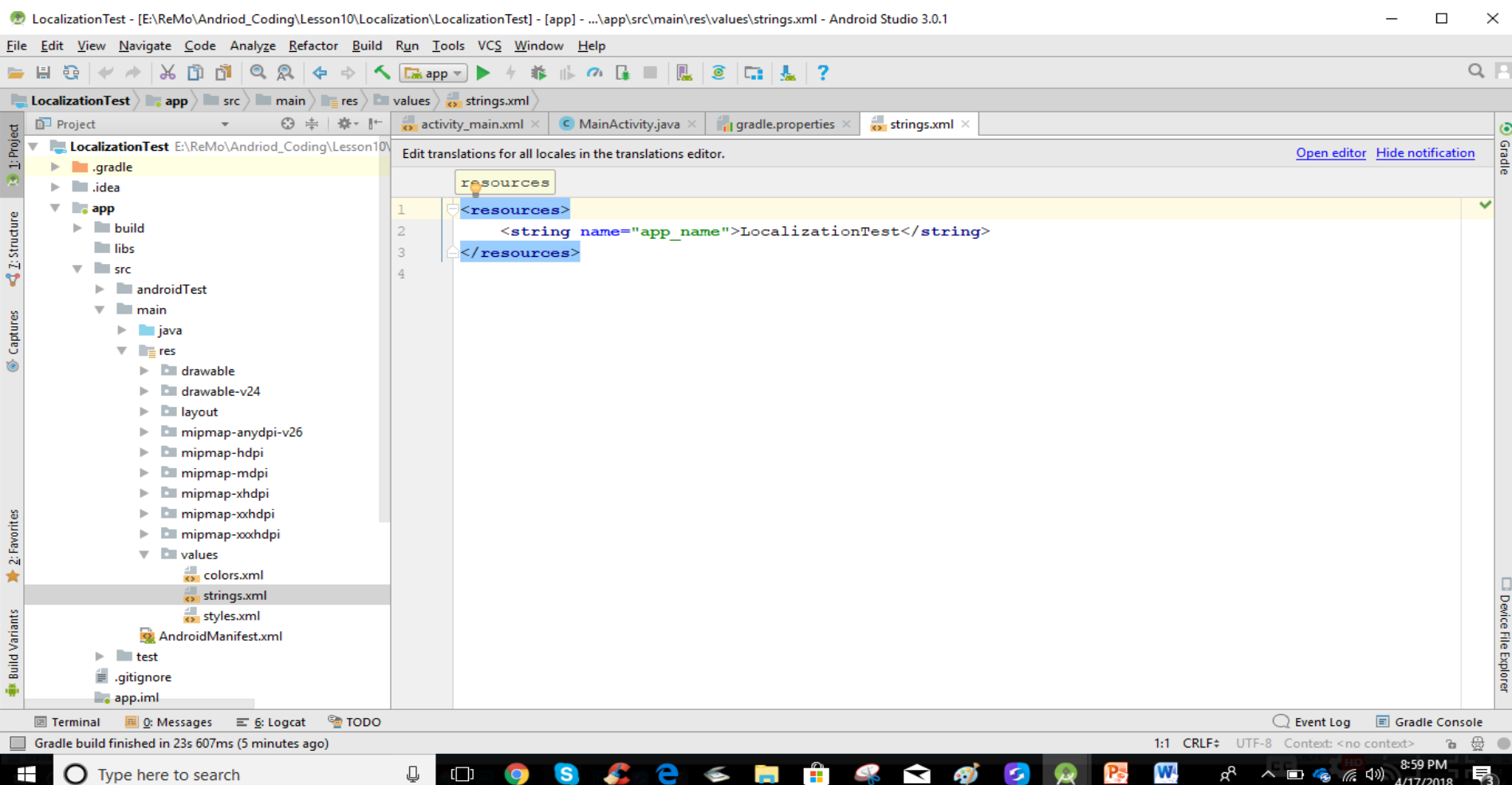


```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
    public void displayResponse(View v) {  
        textview1.setText(R.string.response);  
    }  
}
```

Demo Code: DaysApp, LocalizationDynammic

# Localization Step by Step Screens

## Method 1 : Enter Data into String resource directly.



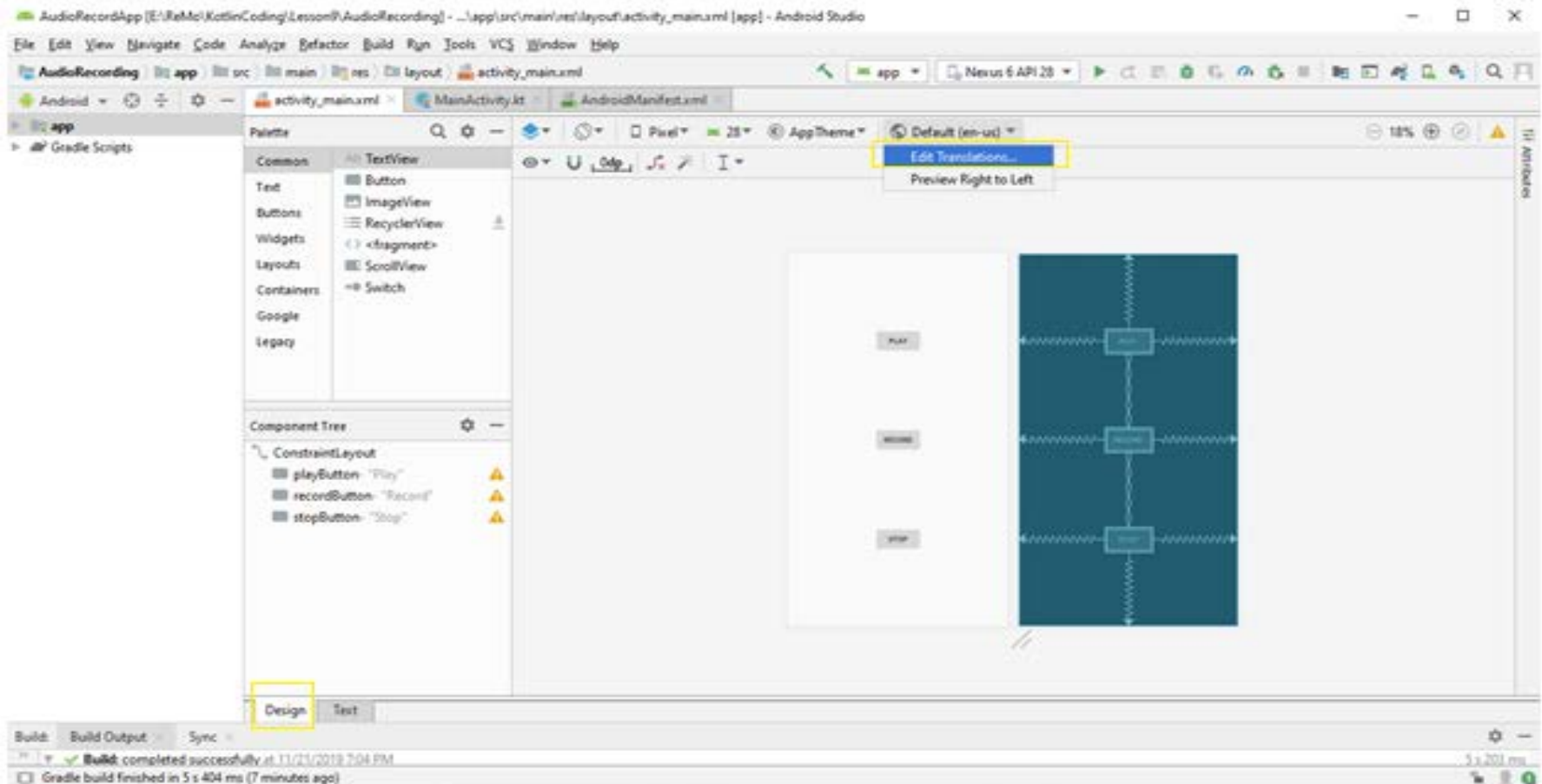
The screenshot displays the Android Studio 3.0.1 interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The toolbar below the menu contains icons for file operations, running, and debugging. The Project Explorer on the left shows the project structure for 'LocalizationTest', with the 'res' folder expanded to show the 'values' directory. The central editor displays the 'strings.xml' file, which contains the following XML code:

```
<resources>
  <string name="app_name">LocalizationTest</string>
</resources>
```

The bottom of the screen features a toolbar with icons for the Terminal, Messages, Logcat, and TODO. The status bar at the very bottom indicates the Gradle build status: 'Gradle build finished in 23s 607ms (5 minutes ago)'. The system tray at the bottom right shows the time as 8:59 PM on 4/17/2018.

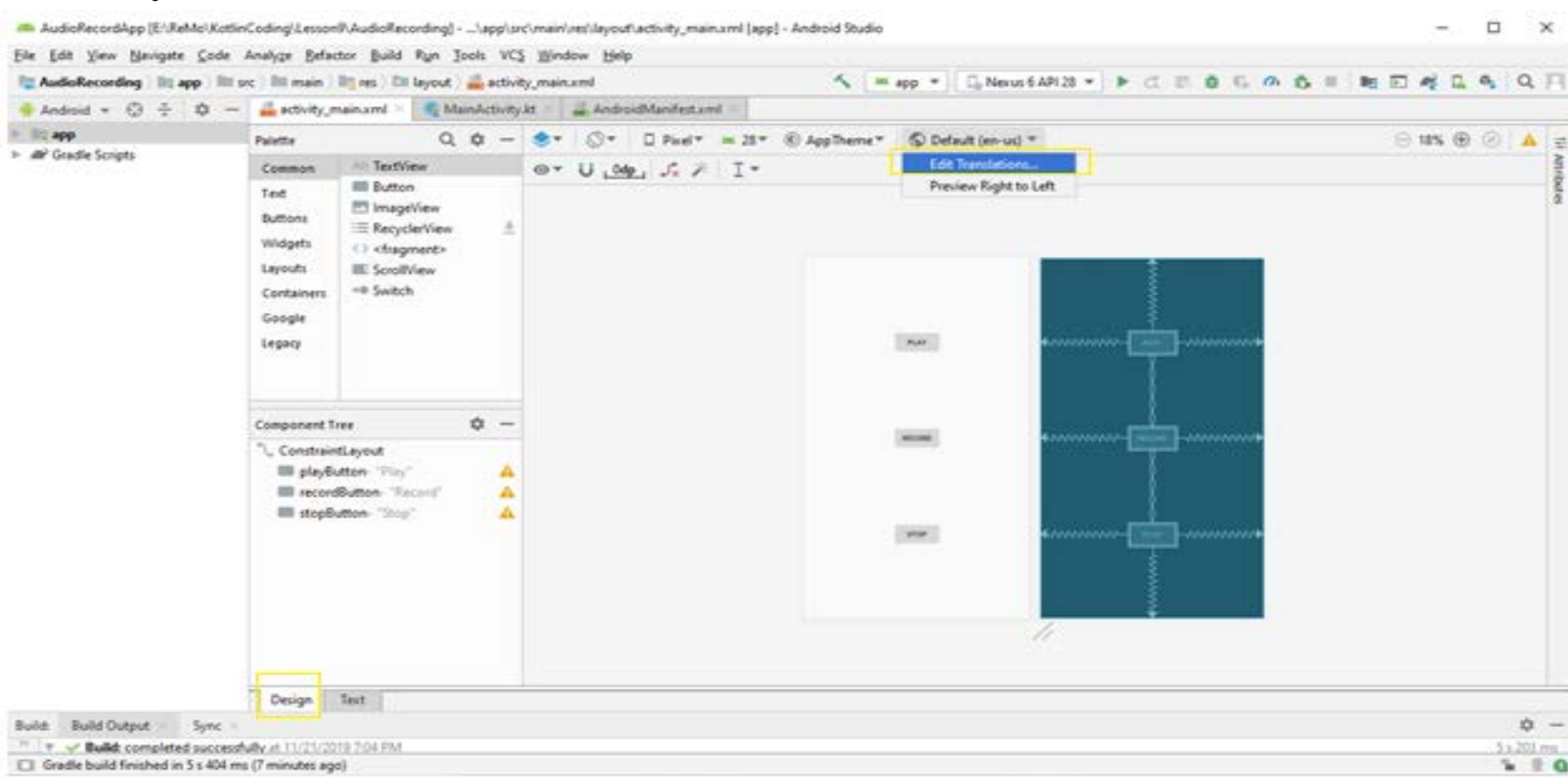
# Localization Step by Step Screens

Method 2 : Step 1 : Go to your Layout Design View and Select Default(en-us) → Edit Translation



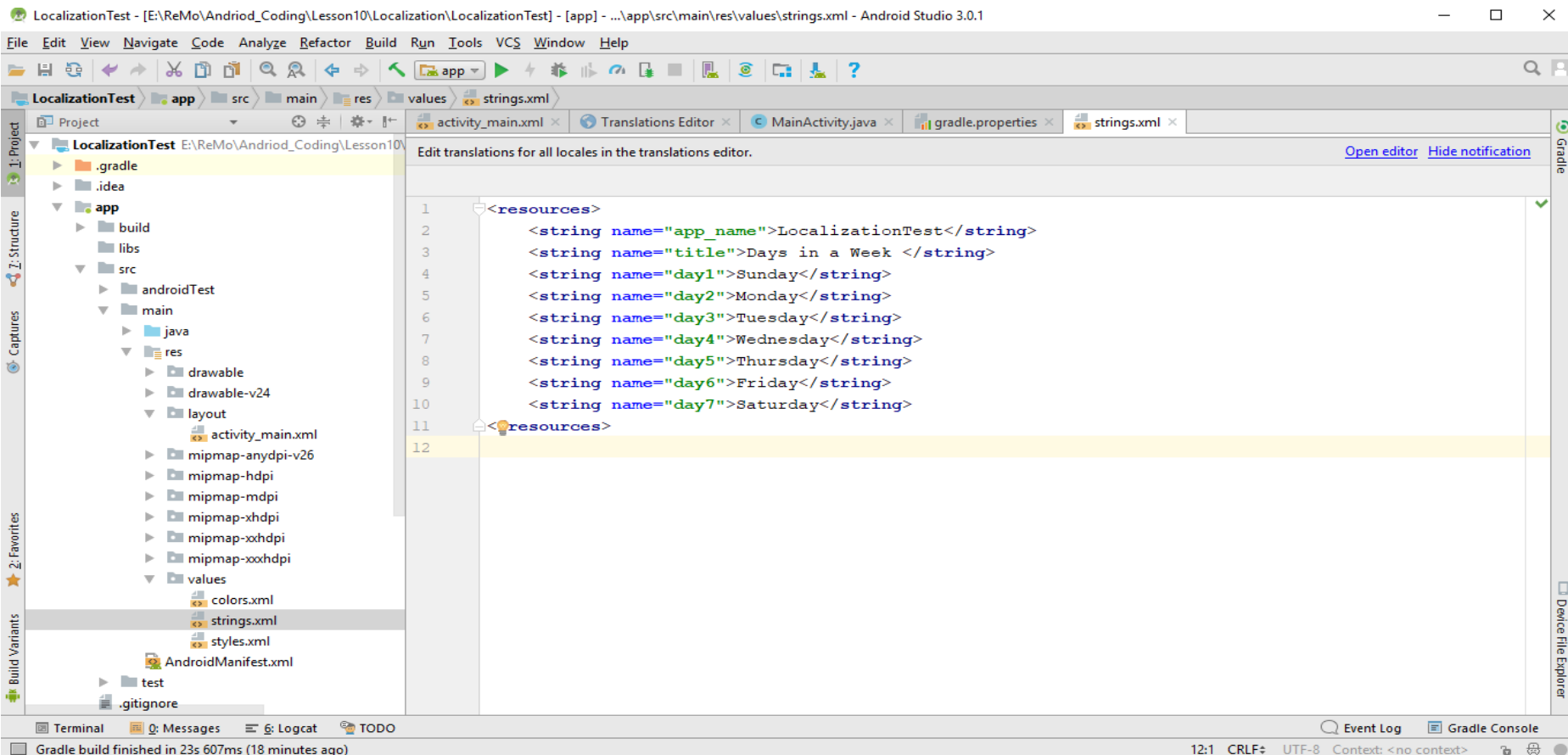
# Localization Step by Step Screens

Step 2 :Click + to add more String inputs as a Key/Value Pair into your resource.



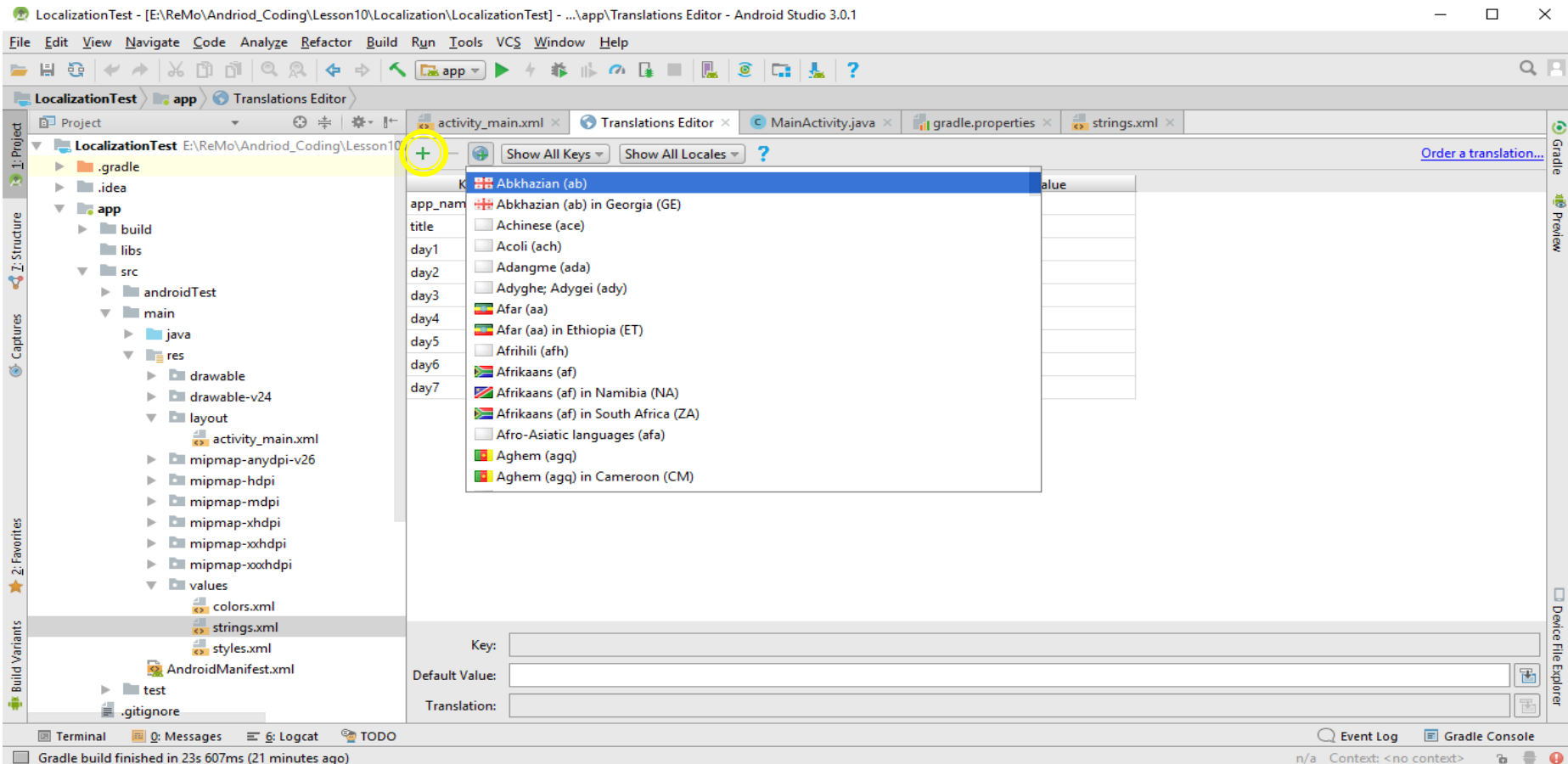
# Localization Step by Step Screens

Step 3 :If you click on your strings.xml, you will notice with all the values.



# Localization Step by Step Screens

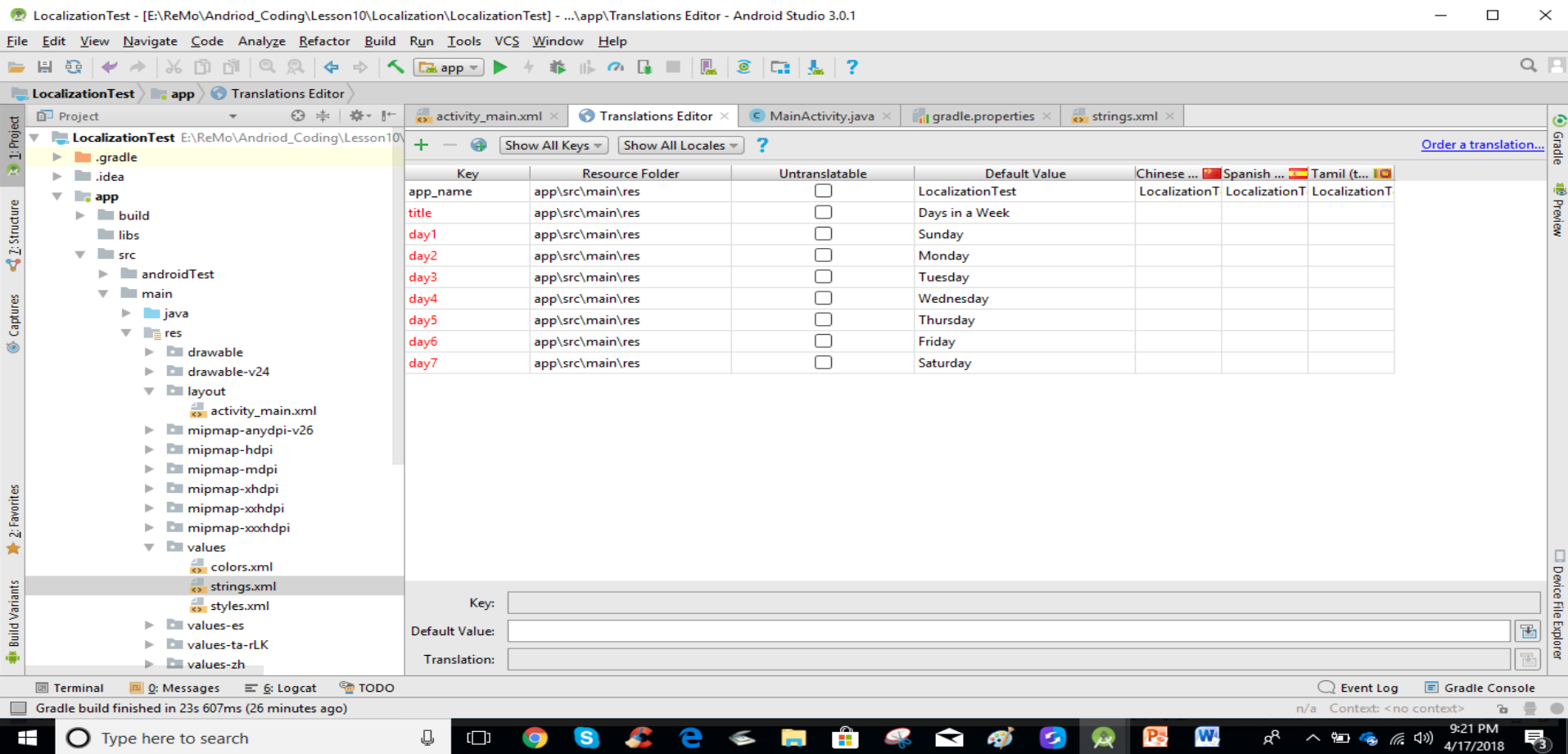
**Step 4 :** To add Locale support, go back to your Translation Editor and click the (+)highlighted icon and choose the desired language.





# Localization Step by Step Screens

**Step 5 :** After Choosing the Language, you will get screen looks like below. Red color indicates that need to provide terms for each Locale.





# Localization Step by Step Screens

**Step 6 :Use Google Language Translator to fill out the values in each locale.**

The screenshot shows a web browser window with multiple tabs. The active tab is 'google language translator'. The address bar shows the URL: [https://www.google.com/search?source=hp&ei=d6bWWtyhJ4mDjwTvrr\\_wDA&q=google+language+translator&oeq=google+language+tra&gs\\_l...](https://www.google.com/search?source=hp&ei=d6bWWtyhJ4mDjwTvrr_wDA&q=google+language+translator&oeq=google+language+tra&gs_l...). The Google logo is on the left, and the search bar contains 'google language translator'. Below the search bar, there are tabs for 'All', 'News', 'Videos', 'Images', 'Maps', 'More', 'Settings', and 'Tools'. The search results show 'About 235,000,000 results (0.61 seconds)'. The main content area displays the translation of 'Saturday' from English to Spanish. The English text 'Saturday' is on the left, and the Spanish text 'sábado' is on the right. Below the English text, it says 'Edit' and '-dē,ˈsə-tər, dā'. Below the Spanish text, it says 'sábado'. At the bottom of the page, there are links to 'Google Translate', 'Languages', 'English', 'Show your site in another', and 'Google Cloud Translation API'. The Windows taskbar is visible at the bottom, showing the Start button, search bar, and various application icons.

Google language translator

About 235,000,000 results (0.61 seconds)

English - detected Spanish

Saturday sábado

Edit

-dē,ˈsə-tər, dā

Open in Google Translate Feedback

Google Translate

<https://translate.google.com/>

Google's free service instantly translates words, phrases, and web pages between English and over 100 other languages.

Languages

Explore the world in over 100 languages with Google ...

English

Google's free service instantly translates words, phrases, and ...

Show your site in another

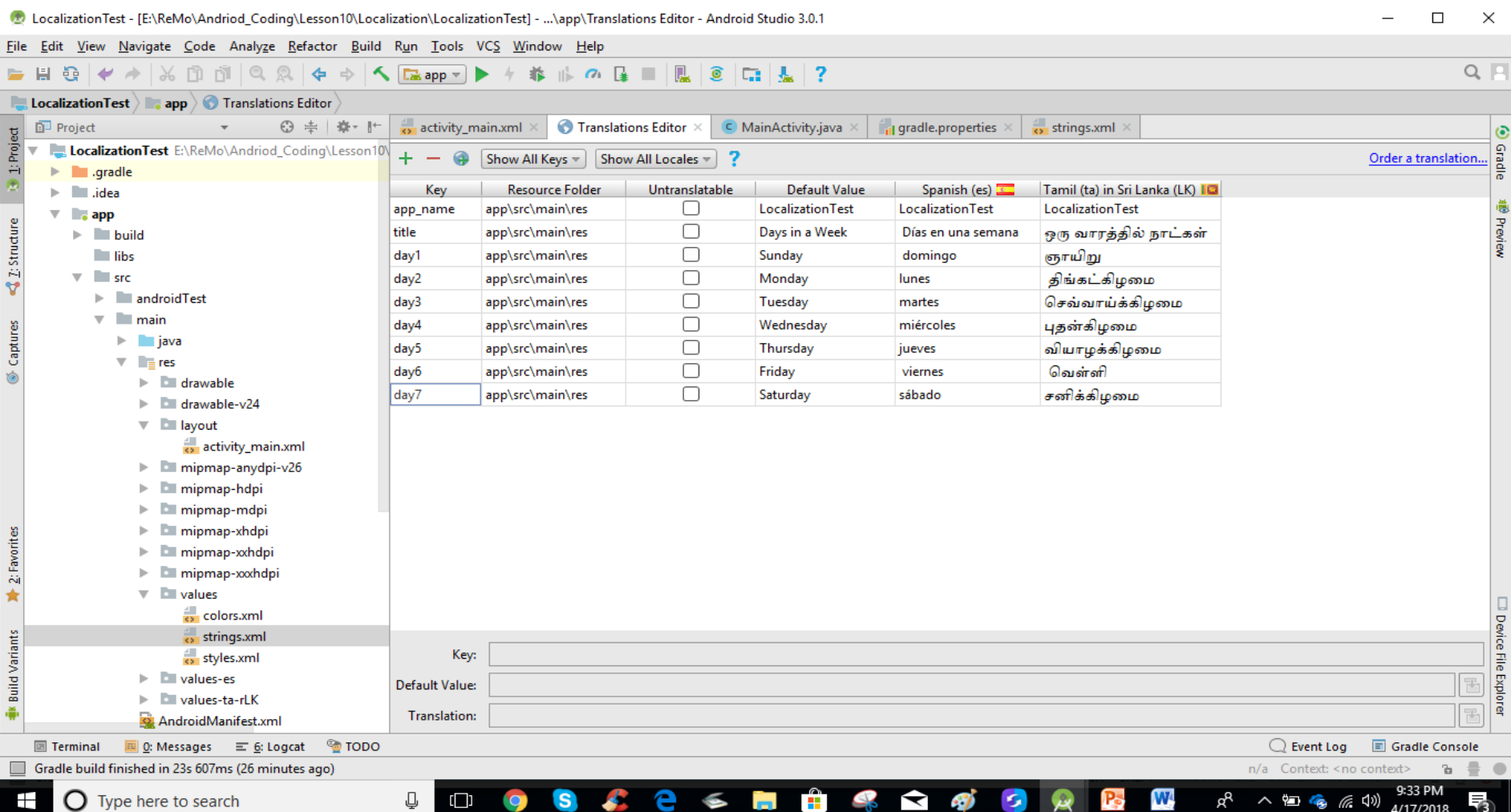
You can show visitors a translated version of your webpage using ...

Google Cloud Translation API

Quickstart - APIs & Reference - Sample Applications - ...

# Localization Step by Step Screens

**Step 7 :Once everything is done, the screen looks below.**



# Localization Step by Step Screens

Step 8 : Open your res -> values->strings.xml, the screen as looks as below

LocalizationTest - [E:\ReMo\Andriod\_Coding\Lesson10\Localization\LocalizationTest] - ...app\Translations Editor - Android Studio 3.0.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

LocalizationTest app src main res values-es

activity\_main.xml Translations Editor MainActivity.java gradle.properties strings.xml

Show All Keys Show All Locales

Key	Resource Folder	Untranslatable	Default Value	Spanish (es)	Tamil (ta) in Sri Lanka (LK)
app_name	app/src/main/res	<input type="checkbox"/>	LocalizationTest	LocalizationTest	LocalizationTest
title	app/src/main/res	<input type="checkbox"/>	Days in a Week	Días en una semana	ஒரு வாரத்தில் நாட்கள்
day1	app/src/main/res	<input type="checkbox"/>	Sunday	domingo	ஞாயிறு
day2	app/src/main/res	<input type="checkbox"/>	Monday	lunes	திங்கட்கிழமை
day3	app/src/main/res	<input type="checkbox"/>	Tuesday	martes	செவ்வாய்க்கிழமை
day4	app/src/main/res	<input type="checkbox"/>	Wednesday	miércoles	புதன்கிழமை
day5	app/src/main/res	<input type="checkbox"/>	Thursday	jueves	வியாழக்கிழமை
day6	app/src/main/res	<input type="checkbox"/>	Friday	viernes	வெள்ளி
day7	app/src/main/res	<input type="checkbox"/>	Saturday	sábado	சனிக்கிழமை

Key:

Default Value:

Translation:

Terminal Messages Logcat TODO

Gradle build finished in 23s 607ms (42 minutes ago)

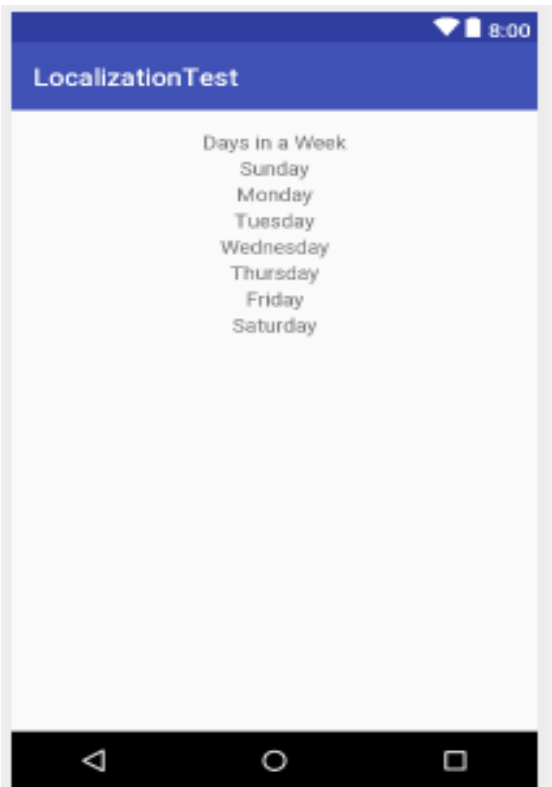
Event Log Gradle Console

n/a Context: <no context>

9:35 PM 4/17/2018

# Localization Step by Step Screens

**Step 9 :Go to your activity\_xml and design layout as shown below. Add eight TextView components and make all component text attribute reference with its string resource.**



```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center"
    android:text="@string/title"
    android:textSize="16sp"/>
```

```
<TextView
```

```
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center"
    android:text="@string/day1"
    android:textSize="16sp"/>
```

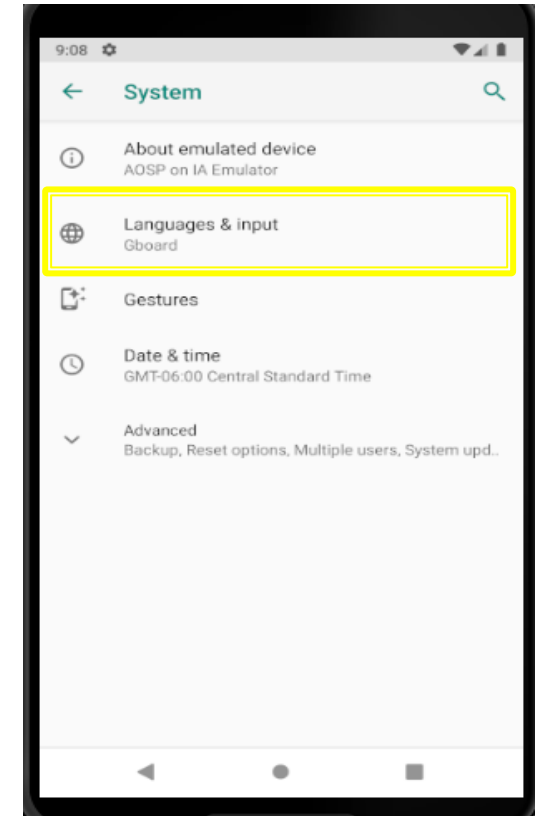
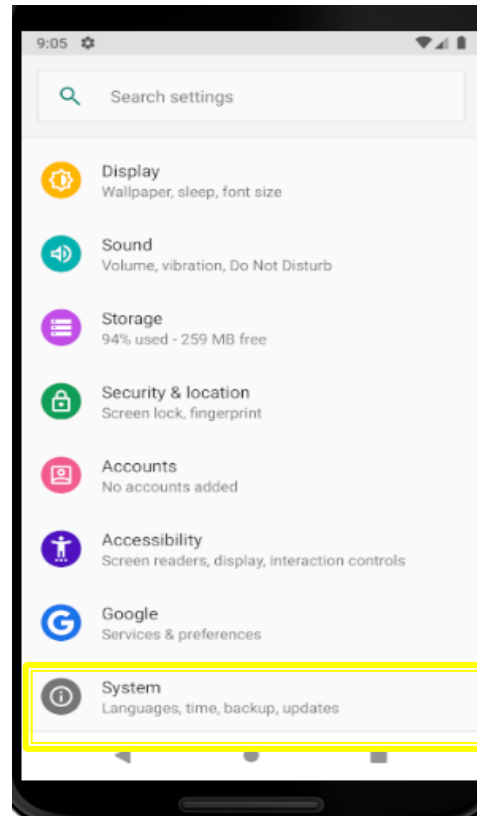
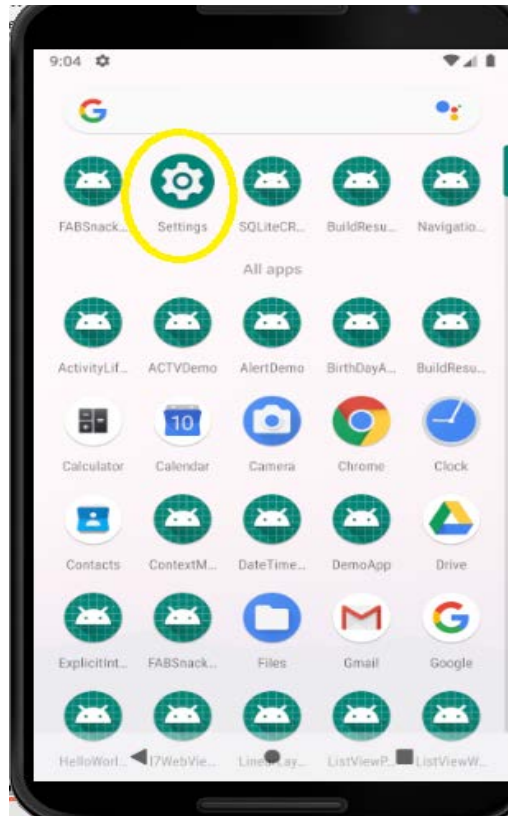
# How to change the Language Settings

From your emulator or real device,

Choose Settings

→ System

→ Language & Input



# How to change the Language Settings

Continuation of Previous slide Screenshots,  
Choose Languages, English is the first priority now. Drag your preferences and click  
Add a language to insert more languages. Make sure that, you should have the String  
resources for the selected language in your app.

