



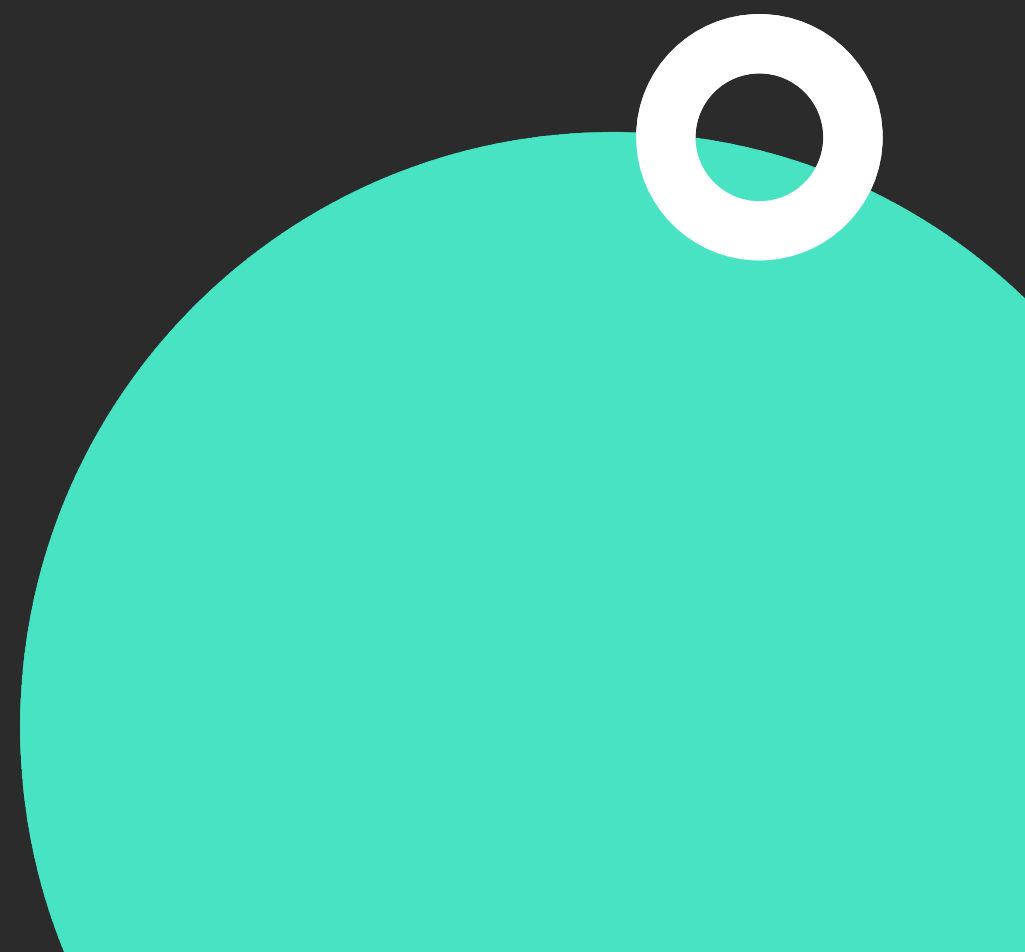
Localization in Android Jetpack Compose: Crafting Multilingual User Experiences



Antonio Leiva

 @lime_cl

 in/antoniolg



Overview

- Importance of localization for reaching **global audiences**.
- How to **implement localization** in Jetpack Compose
 - String formatting
 - Quantity Strings (Plurals)
 - Date localization
 - Number localization
- Support per-app language selection in Android 13+
- Strategies for **managing localized resources** efficiently and optimizing app performance.



Course Demo App

10:51



Cart: 0 products

<div>Name</div> <div></div>	<div>Price</div> <div></div>	<div>Add</div>
-----------------------------	------------------------------	----------------

Total: 0.0



Course Demo App

- This is the [starter project](#) that you can find in the repository
- Cart App that allows to:
 - Add items to the cart
 - Remove items from the cart
 - See the number of products at the top
 - The total at the bottom

10:51



Cart: 0 products

Name	Price	Add
<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>

Total: 0.0



Course Demo App

- It has several issues
 - All strings are hardcoded
 - There are no translations
 - The number of products is not using plurals
 - Dates are not properly formatted
 - Numbers are not properly formatted to region

10:51



Cart: 0 products

Name	Price	Add
<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>

Total: 0.0





Why is localization important?



Why is localization important?

- It's the process of adapting an app to a specific language, culture or region
- Key factor
 - Reaching global audiences
 - Increasing user engagement
- You can start with the most used languages
- Then read your analytics





Localization in Jetpack Compose



Localization in Jetpack Compose

- How to create Strings
- How to use them from Jetpack Compose
- How to translate them to other languages
- Build dynamics Strings
 - With format
 - Quantity Strings (Plurals)
 - Date formatting
 - Numbers Localization



Formatted Strings

- You can add variables to the definition of String resources
 - If there's one value, you just need to define the type
 - If there are several, you also need to specify the order



Formatted Strings

- You can add variables to the definition of String resources
 - If there's one value, you just need to define the type
 - If there are several, you also need to specify the order

```
<string name="total">Total: %s</string>
```



Formatted Strings

- You can add variables to the definition of String resources
 - If there's one value, you just need to define the type
 - If there are several, you also need to specify the order

```
<string name="total">Total: %s</string>
```

```
<string name="total">Cart: %1$d - Total: %2$s</string>
```



Formatted Strings

- You can add variables to the definition of String resources
 - If there's one value, you just need to define the type
 - If there are several, you also need to specify the order

```
<string name="total">Total: %s</string>
```

```
<string name="total">Cart: %1$d - Total: %2$s</string>
```

```
stringResource(id = R.string.total, total.toString())
```



Formatted Strings

- String resources accept several HTML tags
 - They don't work in Compose
 - There are several "hacks" we won't cover



Quantity Strings (Plurals)

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <plurals name="product_quantity">
        <item quantity="one">product</item>
        <item quantity="other">products</item>
    </plurals>
</resources>
```



Quantity Strings (Plurals)

Value	Description
zero	When the language requires special treatment of the number 0 (as in Arabic).
one	When the language requires special treatment of numbers like one (as with the number 1 in English and most other languages; in Russian, any number ending in 1 but not ending in 11 is in this class).
two	When the language requires special treatment of numbers like two (as with 2 in Welsh, or 102 in Slovenian).
few	When the language requires special treatment of "small" numbers (as with 2, 3, and 4 in Czech; or numbers ending 2, 3, or 4 but not 12, 13, or 14 in Polish).
many	When the language requires special treatment of "large" numbers (as with numbers ending 11-99 in Maltese).
other	When the language does not require special treatment of the given quantity (as with all numbers in Chinese, or 42 in English).



Date Formatting

- Dates show different depending on the language
- There's not an easy API to do the localization
- The most advanced API is only available on SDK 26+
- Conversion must be done as part of Compose code



Date Formatting

Value	Description
FULL	Tuesday, October 10, 2023 4:40:34 PM Central European Summer Time
LONG	October 10, 2023 4:40:34 PM GMT+02:00
MEDIUM	Oct 10, 2023 4:40:34 PM
SHORT	10/10/23 4:40 PM



Numbers formatting

- Decimal and thousand separators
 - 🇺🇸 1,000.99
 - 🇪🇸 1.000,99



Numbers formatting

To convert a Double into a String

```
NumberFormat.getInstance().format(double)
```



Numbers formatting

To convert a Double into a String

```
NumberFormat.getInstance().format(double)
```

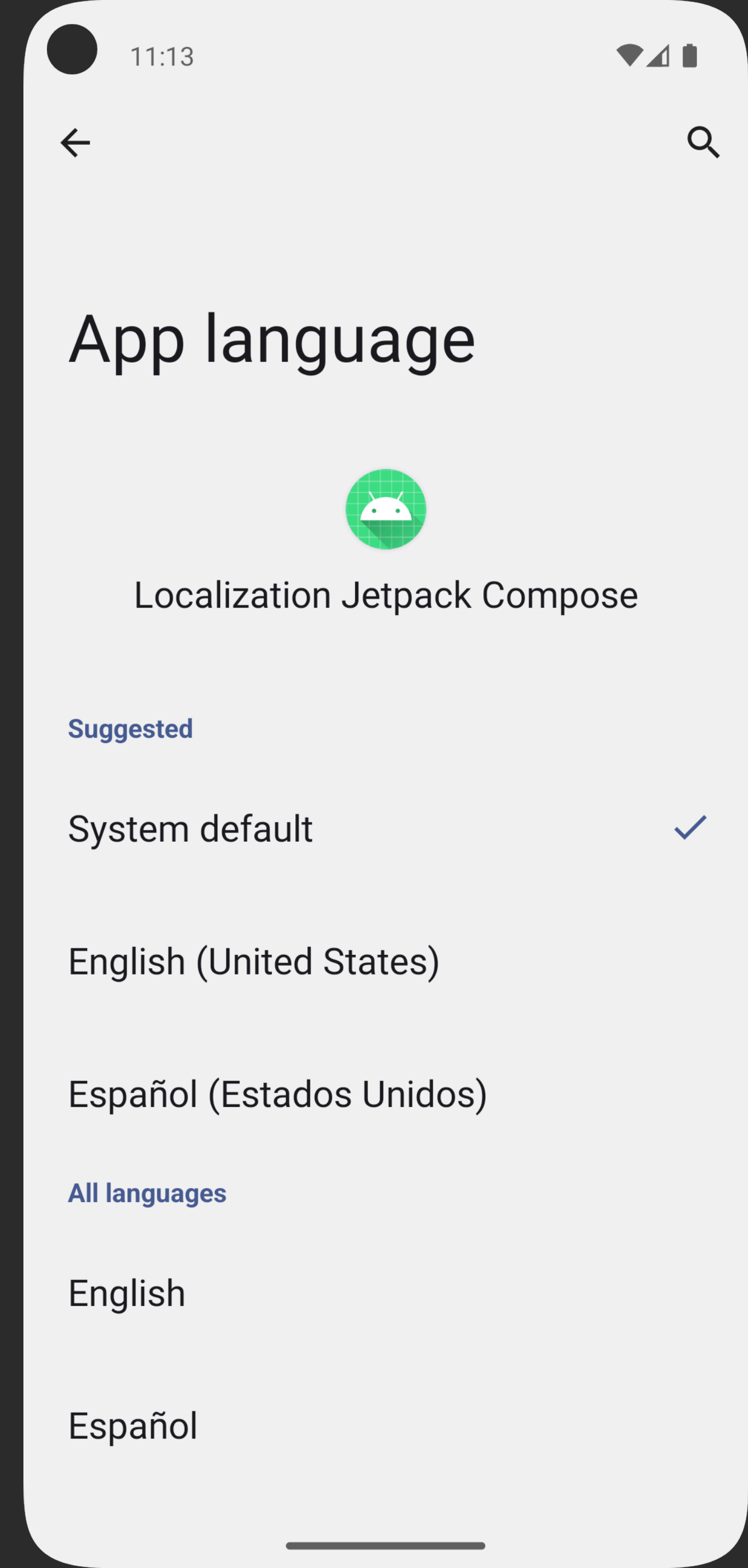
To convert a user input into a double

```
string.replace(",", ".").toDoubleOrNull()
```



Per-App language support

- Since Android 13, users can choose the language per App
- Not by default, your App needs to enable it

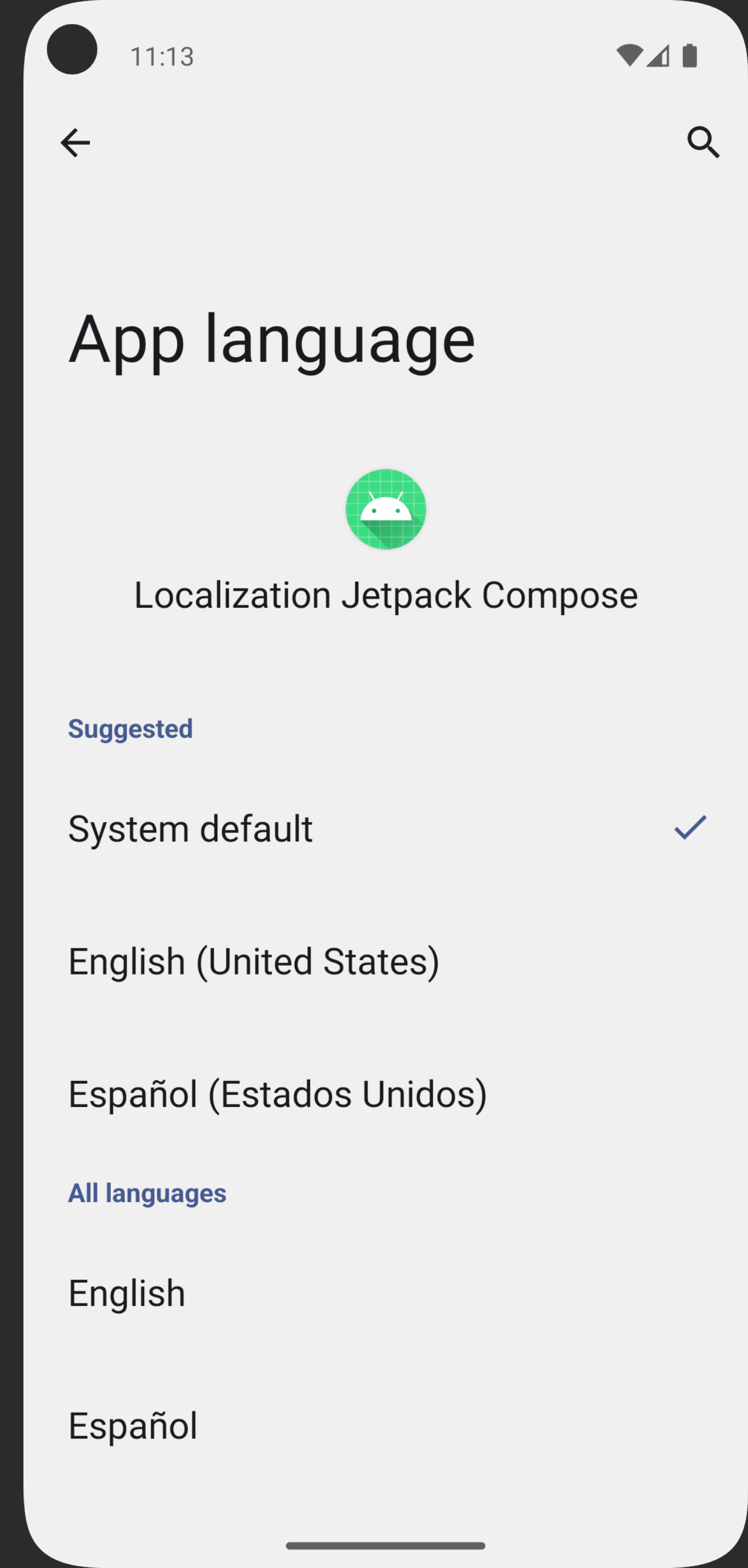


Per-App language support

- Since Android 13, users can choose the language per App
- Not by default, your App needs to enable it

Enable it in build.gradle

```
androidResources {  
    generateLocaleConfig = true  
}
```



Per-App language support

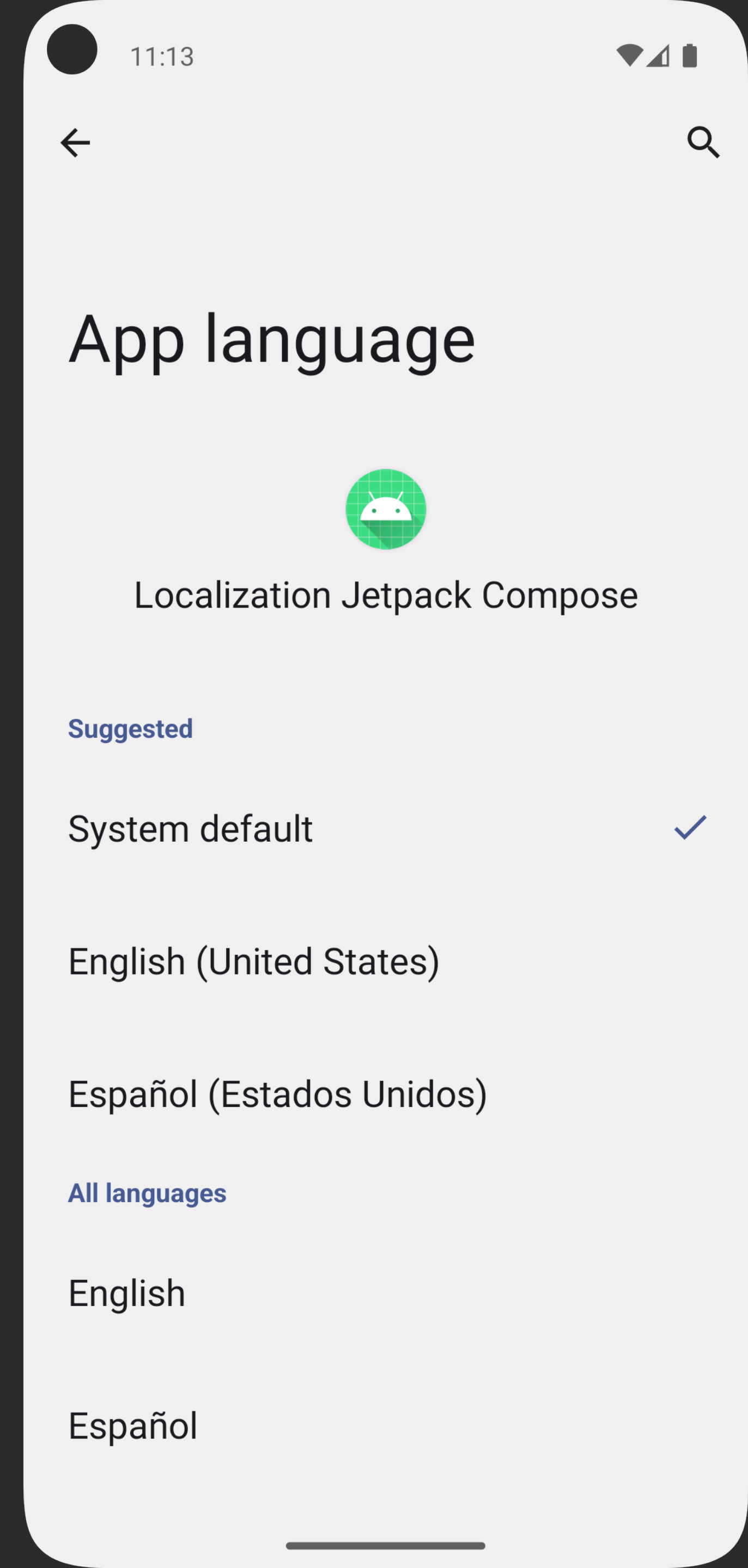
- From Android 13, users can choose the language per App
- Not by default, your App needs to enable it

Enable it in build.gradle

```
androidResources {  
    generateLocaleConfig = true  
}
```

Create the file resources.properties in res folder, and add this

```
unqualifiedResLocale=en
```



Manage localized resources efficiently

- You will probably have String resources in a localized App
 - Try not to duplicate Strings
 - Make use of plurals
- If you use localized drawables
 - Try to prevent the need of one drawable per language
 - If required, make sure to optimise the weight
 - Consider using vectors
 - Try to minimize the part that needs to be different for each language, and use two combined images.



Coding Challenge

- Finish doing App localization
 - Replace remaining hard-coded String
 - Update plurals if required
 - Localize product price in the products list

10:51



Cart: 0 products

<input type="text" value="Name"/>	<input type="text" value="Price"/>	<input type="button" value="Add"/>
-----------------------------------	------------------------------------	------------------------------------

Total: 0.0



Overview

- Importance of localization for reaching **global audiences**.
- How to **implement localization** in Jetpack Compose
 - String formatting
 - Quantity Strings (Plurals)
 - Date localization
 - Number localization
- Support per-app language selection in Android 13+
- Strategies for **managing localized resources** efficiently and optimizing app performance.





Thank You!



Antonio Leiva

 @lime_cl
 in/antoniolg

