

Zasoby statyczne aplikacji!

Poniższa tabelka obrazuje dostępne w aplikacji zasoby statyczne

Types of Resources

The following are the most common types of resources within Android apps:

Name	Folder	Description
Property Animations	animator	XML files that define property animations.
Tween Animations	anim	XML files that define tween animations.
Drawables	drawable	Bitmap files or XML files that act as graphics
Layout	layout	XML files that define a user interface layout
Menu	menu	XML files that define menus or action bar items
Values	values	XML files with values such as strings, integers, and colors.

In addition, note the following key files stored within the `values` folder mentioned above:

Name	File	Description
Colors	<code>res/values/colors.xml</code>	For color definitions such as text color
Dimensions	<code>res/values/dimens.xml</code>	For dimension values such as padding
Strings	<code>res/values/strings.xml</code>	For string values such as the text for a title
Styles	<code>res/values/styles.xml</code>	For style values such as color of the AppBar

My skupimy się na: colors, dimensions. strings, styles. Na pierwszy ogień bierzemy kolory!

colors.xml

Będziemy korzystać ze strony:

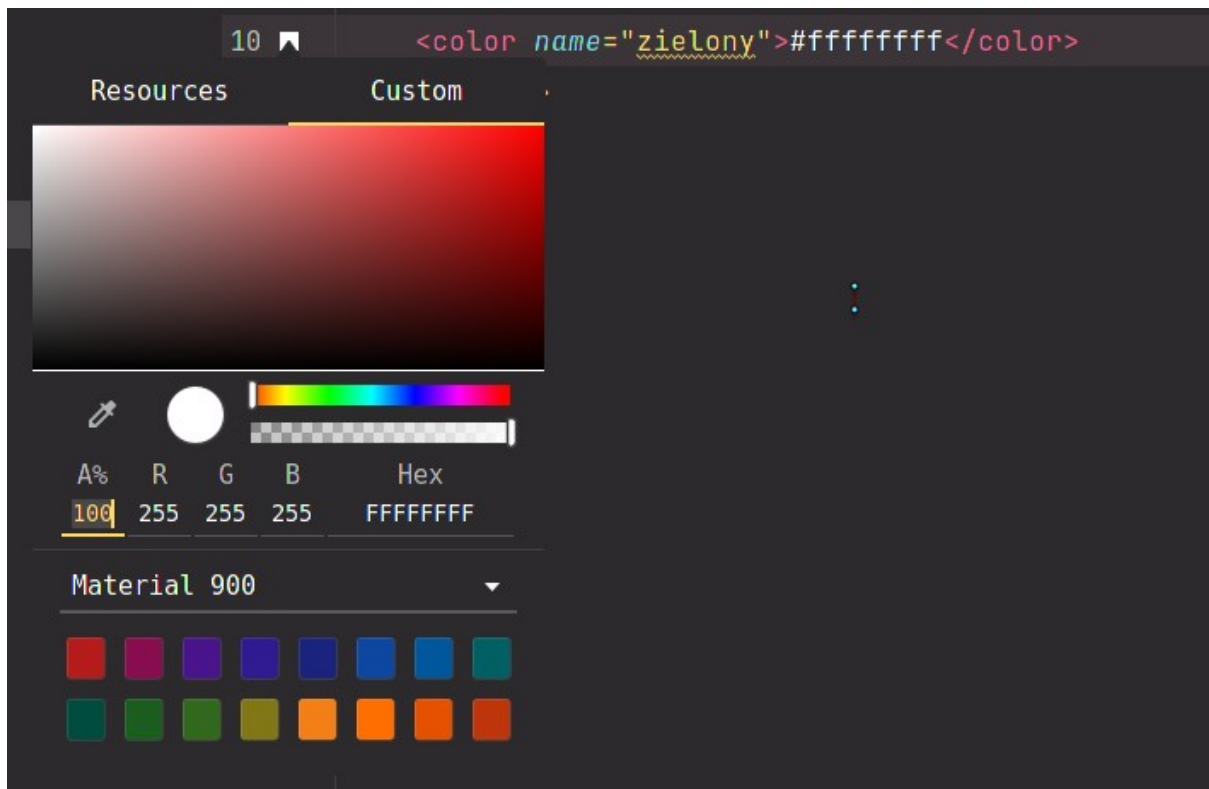
<https://material.io/design/color/the-color-system.html#tools-for-picking-colors>

aby dobrać sobie odpowiednie kolory.

Definiujemy w pliku `res/values/colors.xml` odpowiednie nazwy oraz ustawiamy dla tej nazwy kolor.

Kolor można wybrać także klikając na kwadracik koloru – pojawi się okno wyboru koloru. (JAK NIŻEJ)

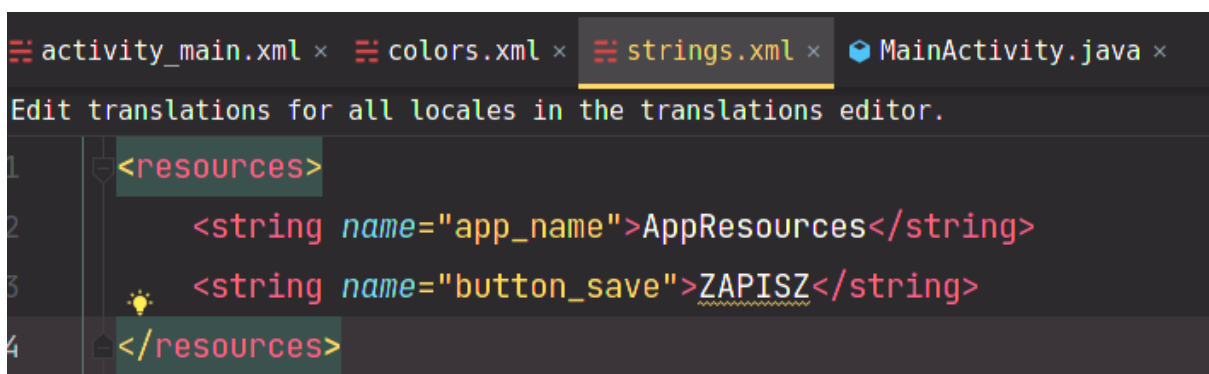
Skoro nazwa jest zielony to wybieramy sobie kolor zielony!



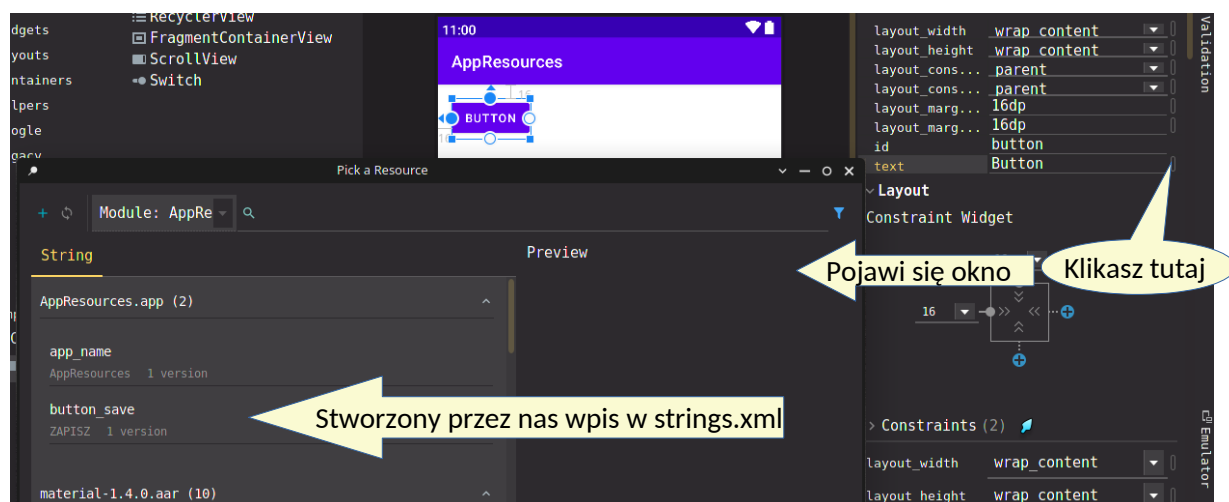
strings.xml

Ten plik jest o wiele bardziej ciekawy. Tutaj definiujemy wszystkie teksty, które mają się pojawiać na elementach stałych naszej aplikacji.

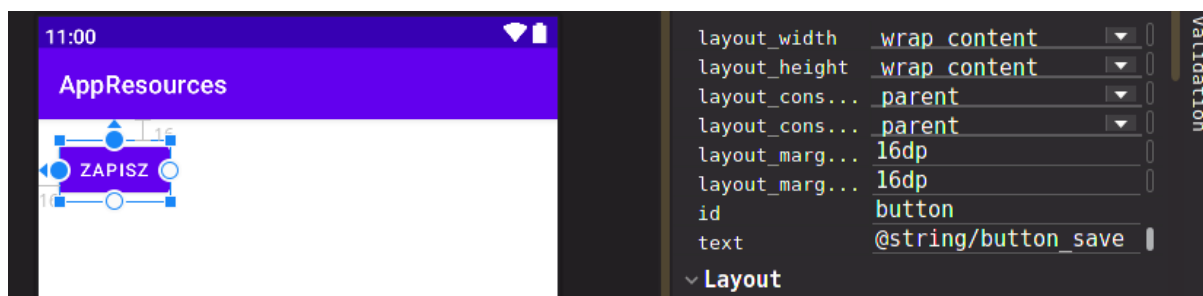
Tworzymy nowy wpis:



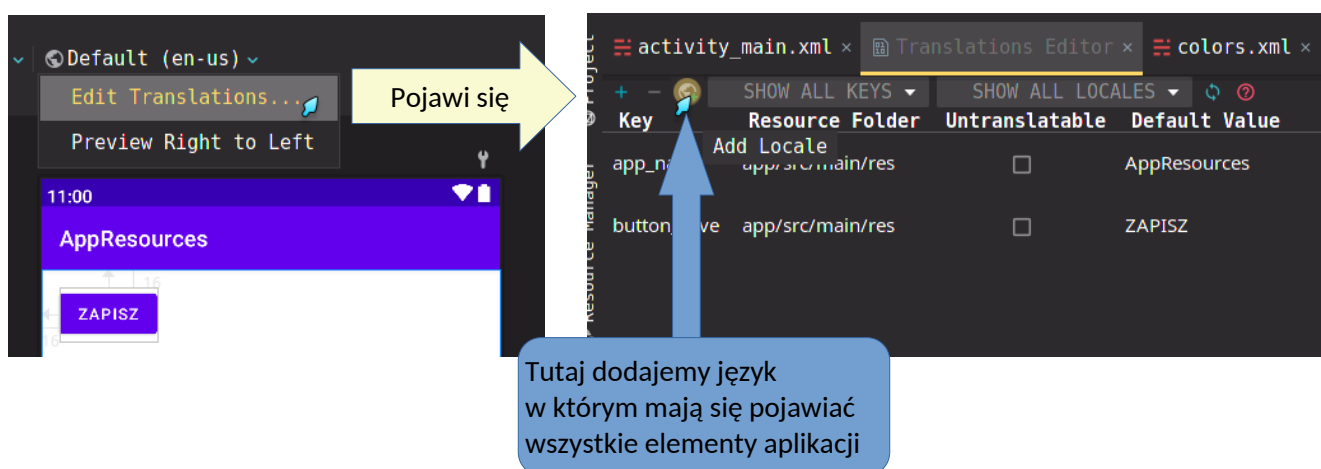
Następnie w naszej aplikacji tworzymy button i używamy naszego wpisu aby dać mu tekst, który ma się wyświetlać.



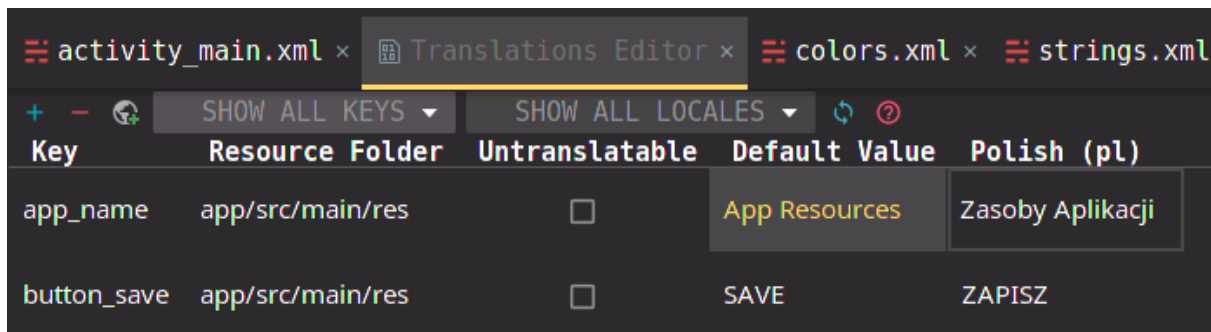
Po wybraniu naszego wpisu tekst w buttonie zmienia się:



Tylko po co się tak męczyć? Ano jeśli piszemy aplikację z dostosowaniem pod język danego kraju to takie zapisy pozwolą na tworzenie tłumaczenia:



Po dodaniu języka polskiego i dodaniu tłumaczenia:



Key	Resource Folder	Untranslatable	Default Value	Polish (pl)
app_name	app/src/main/res	<input type="checkbox"/>	App Resources	Zasoby Aplikacji
button_save	app/src/main/res	<input type="checkbox"/>	SAVE	ZAPISZ

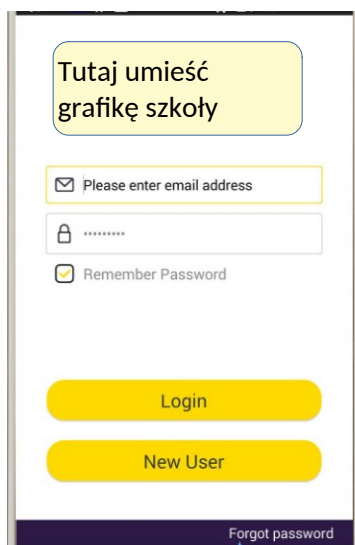
Teraz nasza aplikacja będzie tłumaczona na inne języki w zależności od tego co jej ustawimy. Jeśli danego języka nie dodaliśmy do aplikacji to będzie się wyświetlała wartość domyślna (**Default Value**).

Zadania

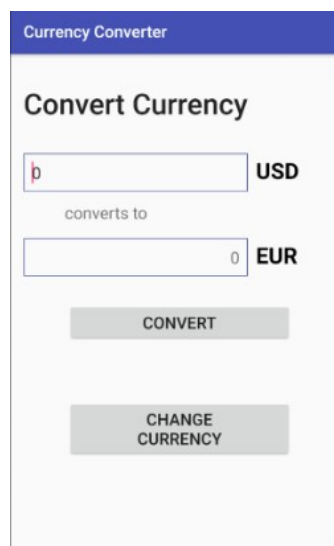
1. Stwórz wygląd aplikacji (dwie aplikacje) jak poniżej, zastosuj tłumaczenie na język polski (default value – english), użyj drawable do zmiany koloru i kształtu przycisków:

<https://stackoverflow.com/questions/18507351/how-to-create-custom-button-in-android-using-xml-styles>

<https://www.codebrainer.com/blog/13-designs-for-buttons-every-android-beginner-should-know>



Przycisk na całą szerokość ekranu



Zmień kolory i kształt przycisków, zmień tło strony (kolor) oraz kolory napisów. Ustaw tłumaczenie dla języka Polskiego.