



Aplikacje typu single Activity

Rafał Zientara



Rotacja ekranu

- Przy obrocie ekranu cały widok jest tworzony na nowo
- Funkcja `onSaveInstanceState()` jest w stanie przechowywać informacje zawarte w aplikacji w obiekcie `Bundle`
- Funkcja `onRestoreInstanceState()` pozwala na wczytanie wcześniej zapisanych danych()
- Ten sam obiekt `Bundle` dla odzyskania stanu znajduje się również w `onCreate()`, jednak należy sprawdzić czy nie jest on `null`em (pierwsze otwarcie aplikacji)
- Bo obiektu `Bundle` możemy wczytywać i zapisywać zmienne pod odpowiednimi kluczami w postaci łańcuchów znaków np. „`moje_dane`”



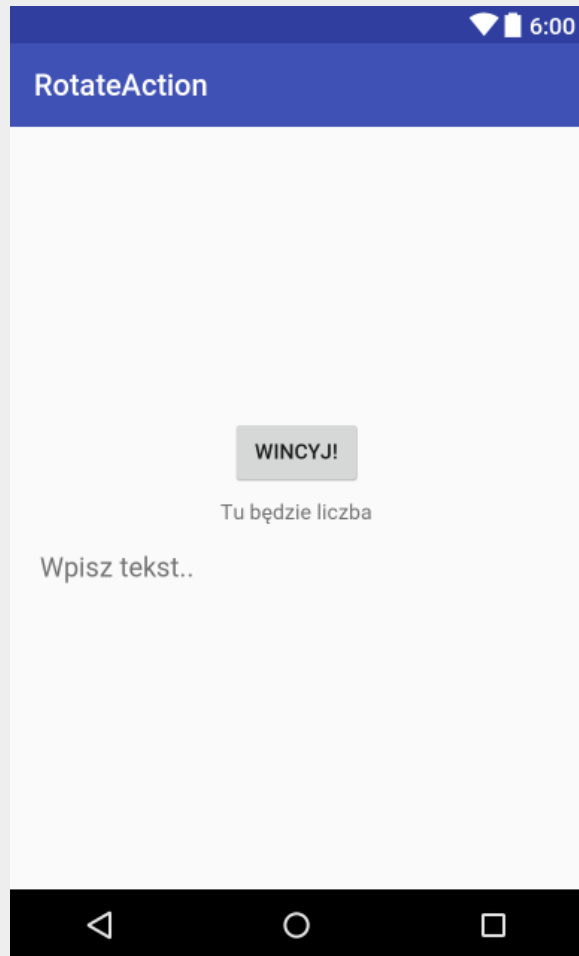
- Przykład

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putInt(SAVED_CLICK_COUNT, count);
}

@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    count = savedInstanceState.getInt(SAVED_CLICK_COUNT);
    countText.setText(String.valueOf(count));
}
```



Zadanie 1



- Przycisk „Wincyj” ma zwiększać licznik w miejscu gdzie mamy „Tu będzie liczba”
- Po obrocie ekranu nadal zachowane będą dane na temat wpisanego tekstu oraz stan licznika



Przedstawianie kolekcji danych - ListView

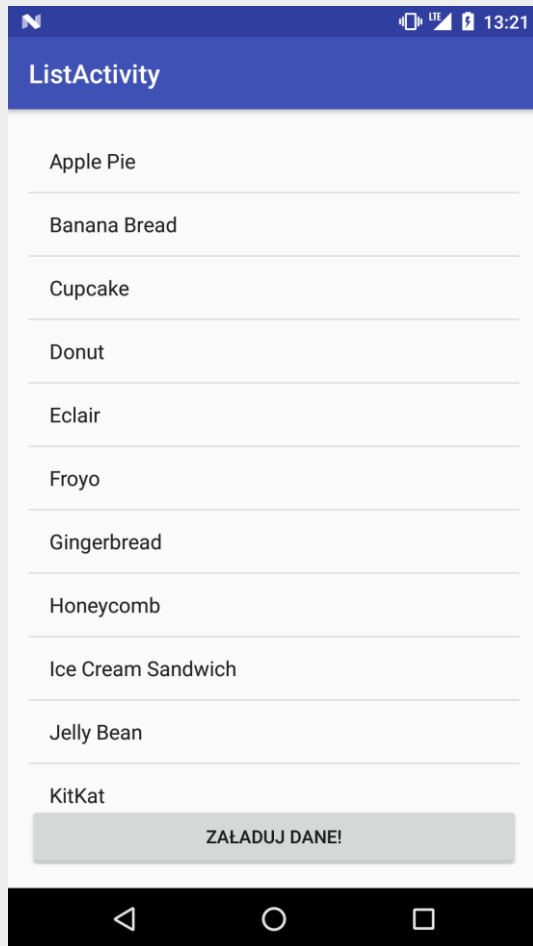
- **ListView** umożliwia przewijanie wyników
- By wyświetlić dane potrzebuje **Adaptora**, oraz danych
- Podstawowy **ArrayAdapter**:

```
ListAdapter adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, versionsList);
```

- Zmienna **versionsList** może być jako tablica lub lista danych



Zadanie 2



- Uwórz layout z na podstawie konceptu
- Przycisk „Załaduj dane” ma wylosować z zasobów jedną z wersji Androida i dodać ją do listy
- Lista wersji Androida dostępna na [Dropbox](#)
- Po obrocie ekranu nadal zachowana będzie lista wylosowanych elementów



- Style można tworzyć w zasobach aplikacji w folderze values
- Format XML
- Pozwala na utworzenie zasad jak mają wyglądać obiekty z ustawionym stylem
- Gdy wprowadzimy zmiany w stylu to wprowadzimy zmiany we wszystkich miejscach gdzie był ustawiony styl

```
<TextView  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:textColor="#00FF00"  
    android:typeface="monospace"  
    android:text="@string/hello" />
```



```
<TextView  
    style="@style/CodeFont"  
    android:text="@string/hello" />
```



- Style można też tworzyć na podstawie innych stylów
- W stylu określamy parametry jakie atrybuty widoku mają zostać ustawione

```
<style name="CodeFont" parent="@android:style/TextAppearance.Medium">  
    <item name="android:layout_width">fill_parent</item>  
    <item name="android:layout_height">wrap_content</item>  
    <item name="android:textColor">#00FF00</item>  
    <item name="android:typeface">monospace</item>  
</style>
```

```
<style name="CodeFont.Red">  
    <item name="android:textColor">#FF0000</item>  
</style>
```




- Do stylu ogólnego aplikacji można ustawiać style do poszczególnych elementów

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
    <item name="android:textColor">@color/colorPrimary</item>
    <item name="android:buttonStyle">@style/ButtonStyle</item>"
</style>
<style name="ButtonStyle" >
    <item name="android:textColor">@android:color/white</item>
    <item name="android:background">@color/colorPrimary</item>
    <item name="android:padding">8dp</item>
</style>
```



- Theme można ustawić przez manifest dla aplikacji...

```
<application android:theme="@style/CustomTheme">
```

- ... lub dla Activity

```
<activity android:theme="@android:style/Theme.Translucent">
```

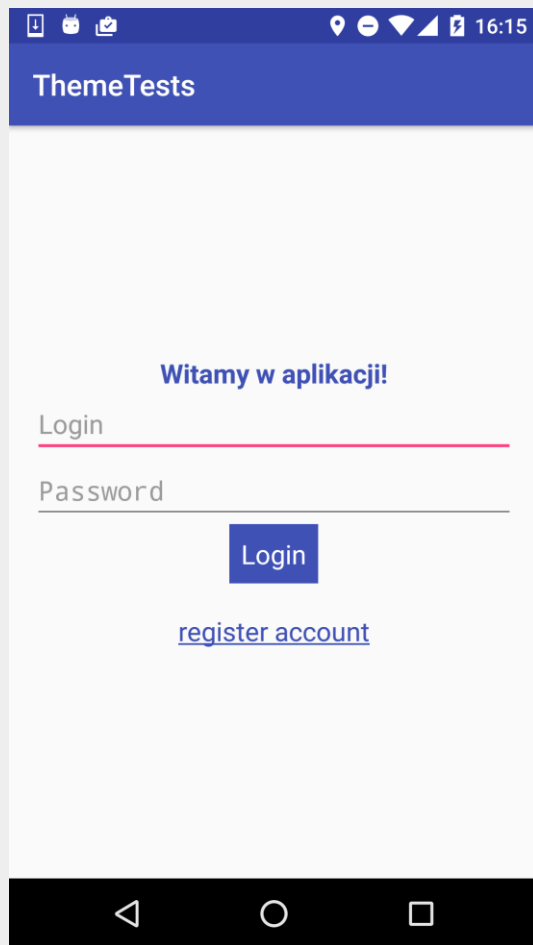
- Android Support Library posiada również gotowe style i tematy
- Spis wszystkich stylów i tematów

<https://android.googlesource.com/platform/frameworks/base/+refs/heads/master/core/res/res/values/styles.xml>

<https://android.googlesource.com/platform/frameworks/base/+refs/heads/master/core/res/res/values/themes.xml>



Zadanie 3



Logowanie i Rejestracja

- Stwórz Theme na całą aplikację gdzie ustawisz style dla EditText i Button (wygląd elementów zależy od Was)
- Można zamiast Theme użyć Stylu i używać go na poszczególnych elementach
- Stwórz ekran logowania według konceptu po lewej



Zadanie 4

- Uwórz layout z na podstawie konceptu
- Ikony Bluetooth znajdują się w LinearLayout o orientacji poziomej, a ten w HorizontalScrollView
- Cała kartka jest jako RelativeLayout

RotateAction



Napisz co u Ciebie!



WYŚLIJ