

# Akademia Androida



**UNIZETO**  
TECHNOLOGIES

# Android Studio



# Czego potrzebujemy?

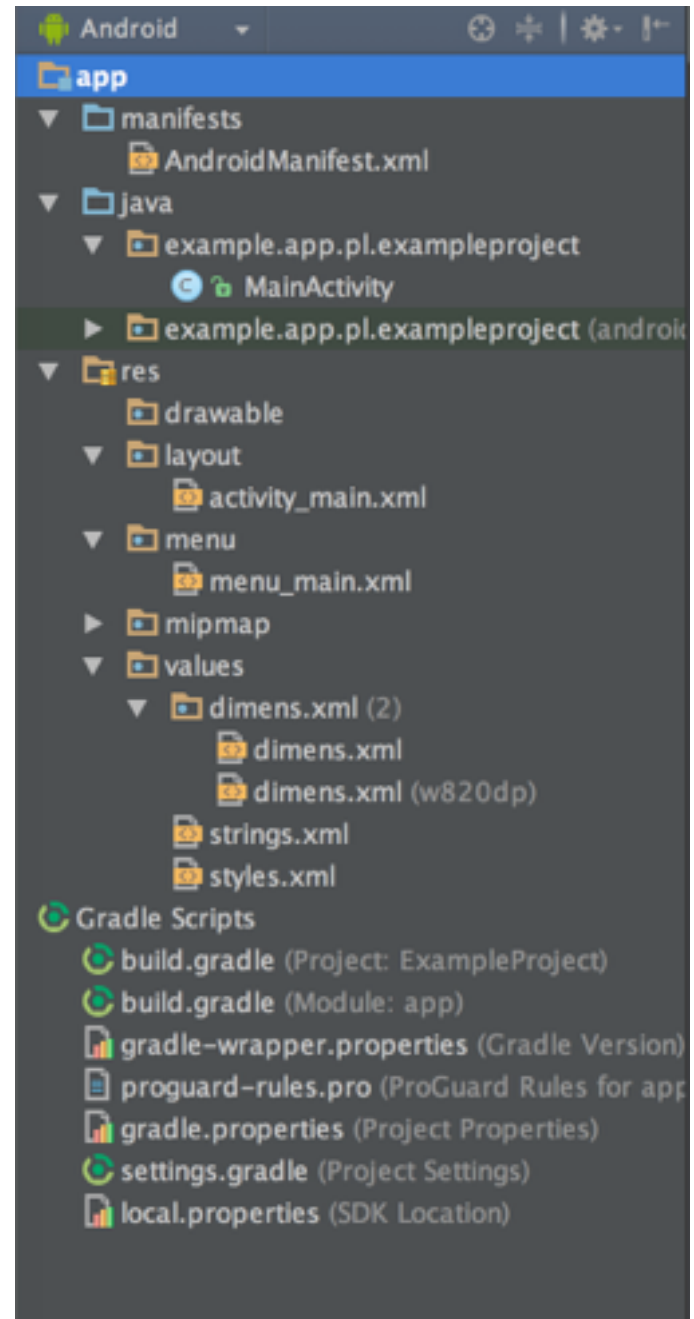
- Środowisko (IDE + SDK):  
<https://developer.android.com/sdk/index.html>
- Java SE (JDK + JRE):  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

# Pierwszy projekt,

# Hello World :)

- <http://developer.android.com/training/basics/firstapp/creating-project.html>

# Struktura projektu



# AndroidManifest.xml

- Każda aplikacja musi zawierać ten plik!
- Podstawowe informacje o aplikacji:
  - nazwa pakietu,
  - komponenty aplikacji (activity, service, broadcast receivers, content provider etc.),
  - uprawnienia aplikacji,
  - API level itp.
- Link: <http://developer.android.com/guide/topics/manifest/manifest-intro.html>

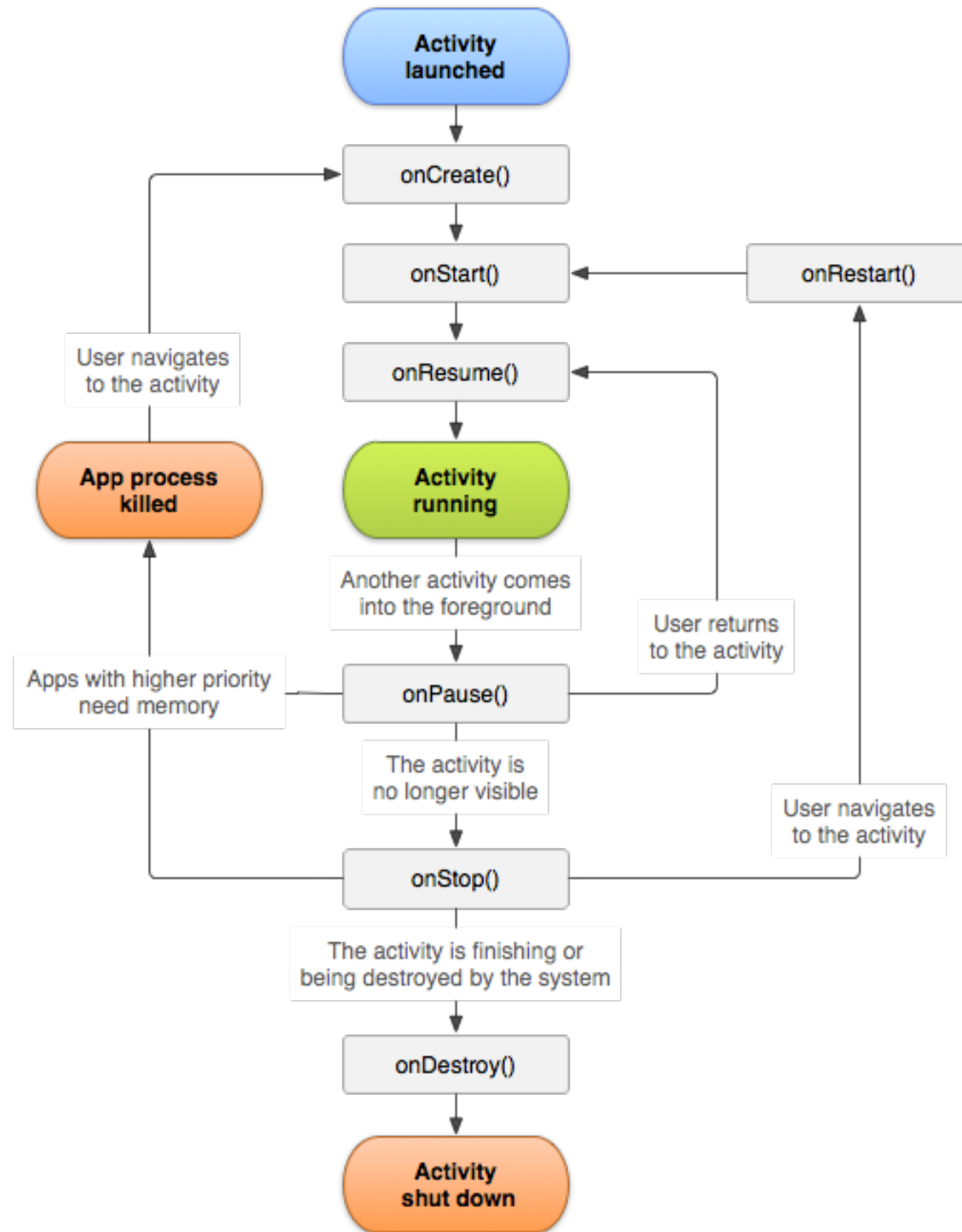
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="example.app.pl.exampleproject" >
4
5     <uses-permission android:name="android.permission.INTERNET" />
6     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
7
8     <application
9         android:allowBackup="true"
10        android:icon="@mipmap/ic_launcher"
11        android:label="@string/app_name"
12        android:theme="@style/AppTheme" >
13         <activity
14             android:name=".MainActivity"
15             android:label="@string/app_name" >
16             <intent-filter>
17                 <action android:name="android.intent.action.MAIN" />
18
19                 <category android:name="android.intent.category.LAUNCHER" />
20             </intent-filter>
21         </activity>
22
23         <service
24             android:name=".ExampleService"
25             android:exported="false" />
26
27     </application>
28
29 </manifest>
30
```

# Cykl życia activity

Method	Description	Killable?	Next
<code>onCreate()</code>	Called when the activity is first created. This is where you should do all of your normal static set up: create views, bind data to lists, etc. This method also provides you with a Bundle containing the activity's previously frozen state, if there was one. Always followed by <code>onStart()</code> .	No	<code>onStart()</code>
<code>onRestart()</code>	Called after your activity has been stopped, prior to it being started again. Always followed by <code>onStart()</code>	No	<code>onStart()</code>
<code>onStart()</code>	Called when the activity is becoming visible to the user. Followed by <code>onResume()</code> if the activity comes to the foreground, or <code>onStop()</code> if it becomes hidden.	No	<code>onResume()</code> or <code>onStop()</code>
<code>onResume()</code>	Called when the activity will start interacting with the user. At this point your activity is at the top of the activity stack, with user input going to it. Always followed by <code>onPause()</code> .	No	<code>onPause()</code>
<code>onPause()</code>	Called when the system is about to start resuming a previous activity. This is typically used to commit unsaved changes to persistent data, stop animations and other things that may be consuming CPU, etc. Implementations of this method must be very quick because the next activity will not be resumed until this method returns. Followed by either <code>onResume()</code> if the activity returns back to the front, or <code>onStop()</code> if it becomes invisible to the user.	Pre-HONEYCOMB	<code>onResume()</code> or <code>onStop()</code>
<code>onStop()</code>	Called when the activity is no longer visible to the user, because another activity has been resumed and is covering this one. This may happen either because a new activity is being started, an existing one is being brought in front of this one, or this one is being destroyed. Followed by either <code>onRestart()</code> if this activity is coming back to interact with the user, or <code>onDestroy()</code> if this activity is going away.	Yes	<code>onRestart()</code> or <code>onDestroy()</code>
<code>onDestroy()</code>	The final call you receive before your activity is destroyed. This can happen either because the activity is finishing (someone called <code>finish()</code> on it, or because the system is temporarily destroying this instance of the activity to save space. You can distinguish between these two scenarios with the <code>isFinishing()</code> method.	Yes	<i>nothing</i>

- Link: <http://developer.android.com/reference/android/app/Activity.html>

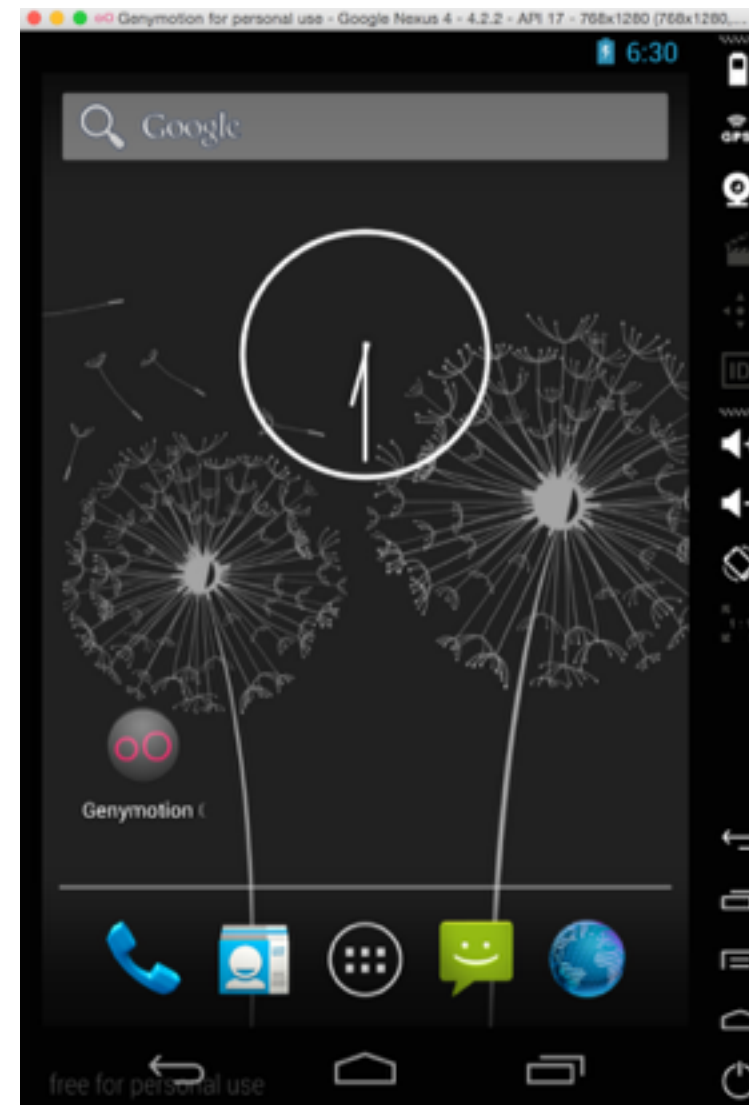




# Emulator



Android Studio



Genymotion

# Dziękuję za uwagę!

- Kontakt: [madwizut@gmail.com](mailto:madwizut@gmail.com)