



Programowanie mobilne

Wstęp do programowania w systemie Android

dr inż. Andrzej Wilczyński - www.awilczynski.me

Android (1)

- Platforma/System operacyjny dla urządzeń mobilnych (nie tylko telefonów/tabletów)
- Kompletny stos elementów od programu rozruchowego i bibliotek po interfejsy API, wbudowane aplikacje i pakiet SDK
- Opensource: <https://source.android.com/>
- Duże możliwości modyfikacji UI

Dokumentacja dla deweloperów jest dostępna tutaj:

<https://developer.android.com/>

Android (2)

- Używany również w:
 - zegarkach - <https://wearos.google.com/> (Wear OS)
 - TV, <https://www.android.com/tv/>
 - w samochodach <https://www.android.com/auto/>



Android (3)

Platformę Android można podzielić na pięć części. Są to:

- Aplikacje
- Framework aplikacji
- Biblioteki warstwy pośredniej
- System operacyjny
- Pakiet SDK i narzędzia programistyczne



Android – aplikacje

Wszystkie aplikacje zainstalowane na urządzeniu, niektóre oferowane jako część samego Androida (Camera, Gallery, Browser). Istnieją również aplikacje Google'a wbudowane w większość oficjalnych komplikacji (Gmail, Maps).



Android – framework aplikacji

Zapewnia obsługę aplikacji, służy do ich uruchamiania. Obejmuje część pakietu SDK i interfejsy API. Kiedy program potrzebuje dostępu na przykład do czujników sprzętowych może pobrać je poprzez framework aplikacji.



Android – biblioteki warstwy pośredniej

Komponenty oprogramowania znajdujące się pomiędzy systemem operacyjnym i aplikacjami lub frameworkiem aplikacji. Obejmują między innym funkcję przechowywania danych oraz środowisko uruchomieniowe Dalvik.



Android - system operacyjny

System operacyjny Androida jest oparty na Linuxie. Zapewnienie on interfejs sprzętowego za pomocą zbioru sterowników urządzeń (na przykład sterownik audio czy wideo), przetwarza dane wyjściowe od użytkowników itd.



Android – pakiet SDK i narzędzia programistyczne

Pakiet SDK obejmuje biblioteki zapewniające dostęp do wszystkich funkcji urządzenia, a także framework aplikacji. Udostępnia między innymi narzędzia do kompilowania kodu źródłowego aplikacji do zrozumiałych dla maszyny wirtualnej Dalvik plików klas *.dex*.



Dalvik

Dalvik to opracowana przez firmę Google maszyna wirtualna Javy. Odpowiada ona za wykonywanie urchamianych na Androdzie aplikacji. Zaprojektowano go z myślą o trzech podstawowych wymaganiach:

- musi działać szybko nawet na słabych procesorach
- musi działać w systemach o małej ilości pamięci
- musi być energooszczędny

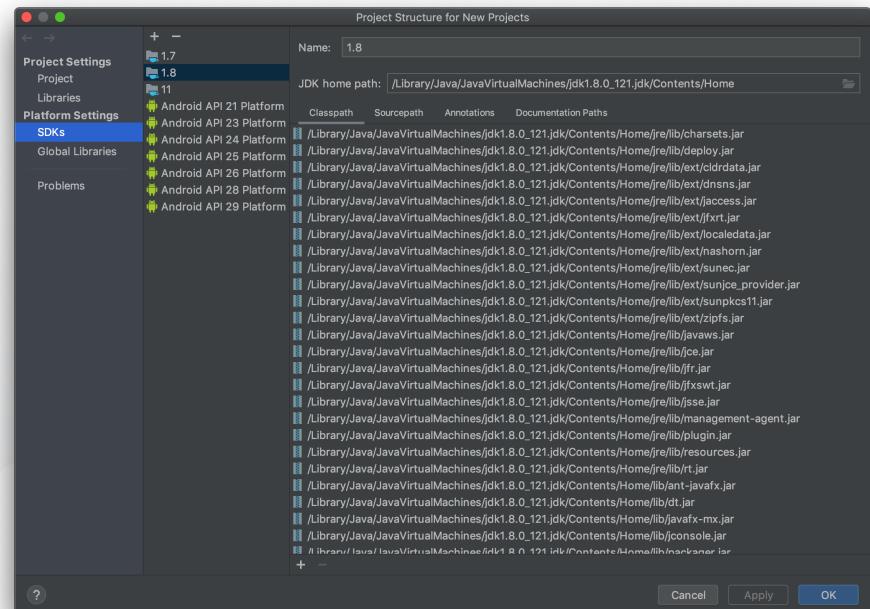


Instalacja i konfiguracja środowiska

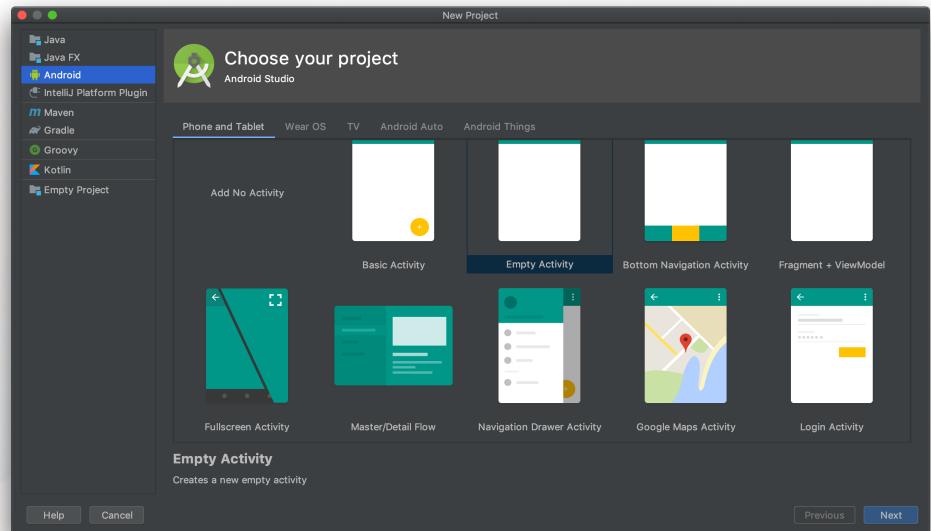
- Android SDK (Android Software Development Kit)
- JDK (Java Development Kit)
- IntelliJ IDEA
- AVD (Android Virtual Device)



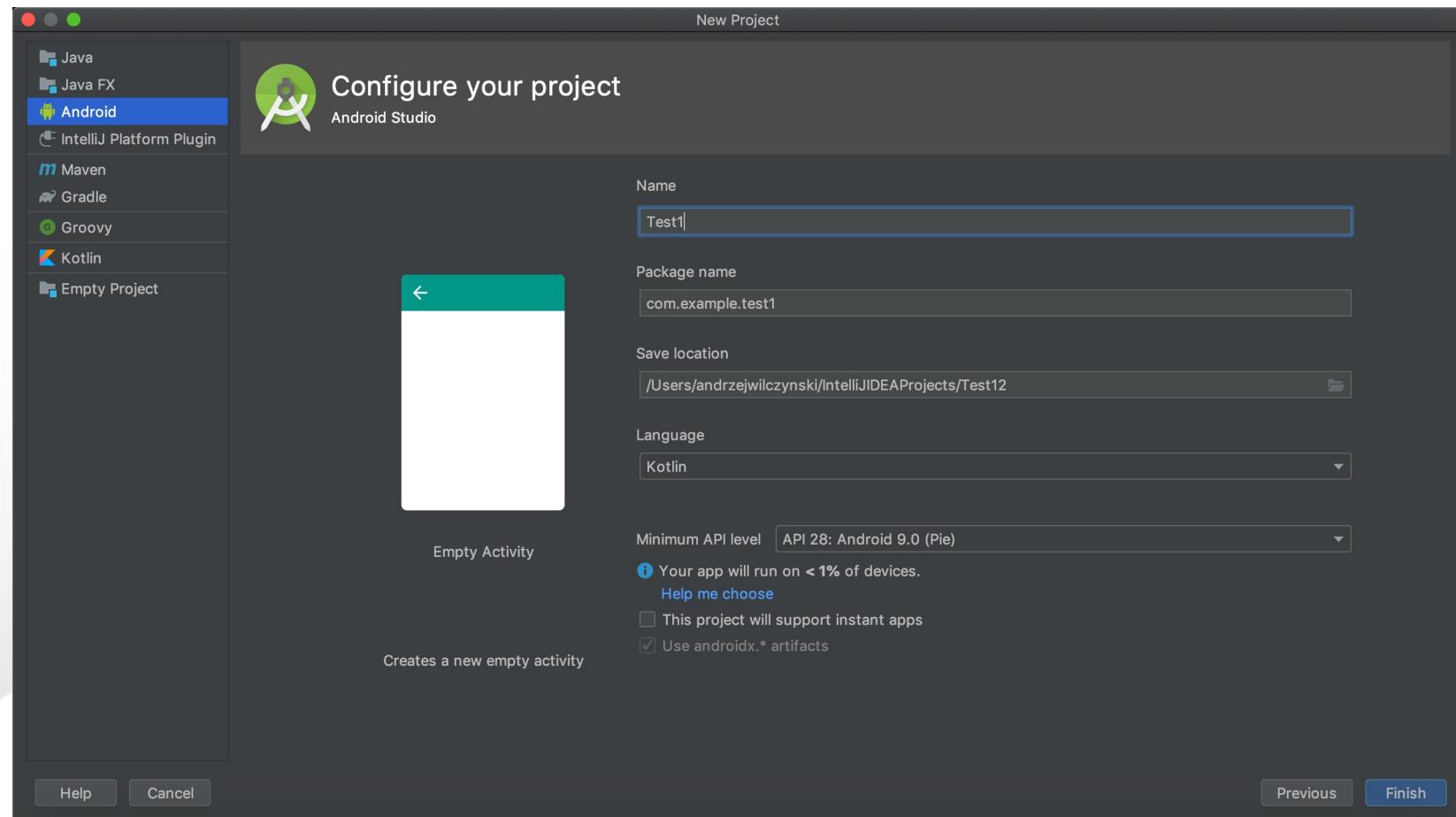
Dodawanie JDK i SDK



Tworzenie projektu (1)



Tworzenie projektu (2)



Struktura plików

The screenshot shows the IntelliJ IDEA IDE interface with the following details:

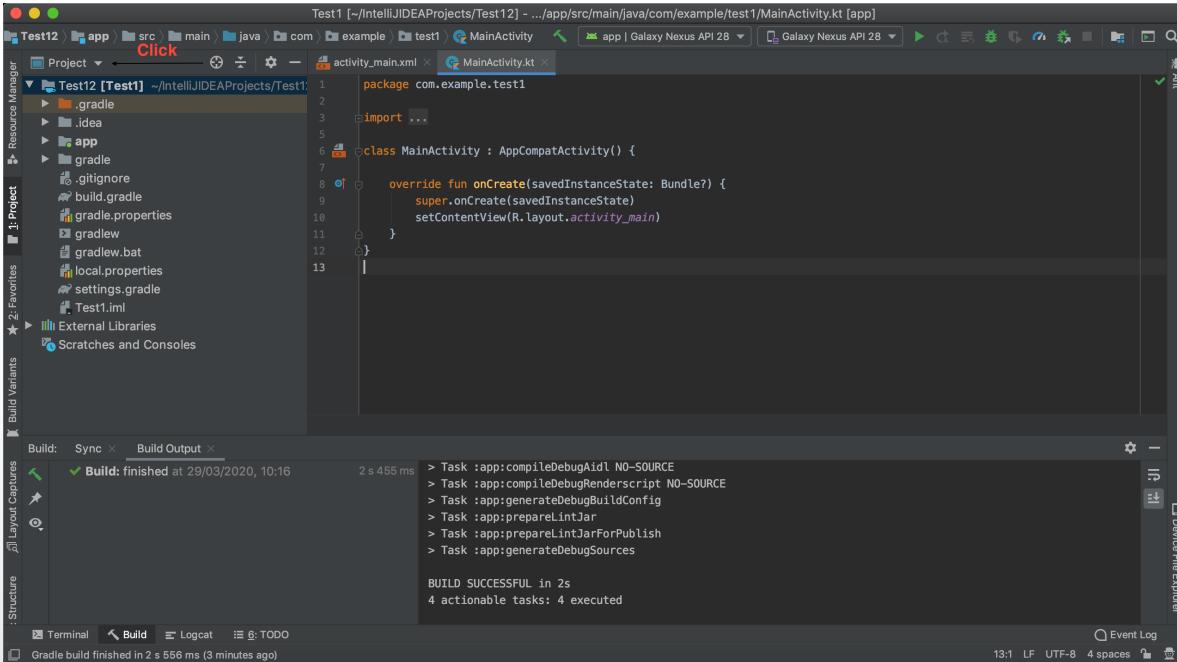
- Title Bar:** Test1 [~/IntelliJIDEAProjects/Test1] - .../app/src/main/java/com/example/test1/MainActivity.kt [app]
- Toolbars:** Standard Java development tools like Run, Stop, Refresh, and others.
- Project View:** Shows the project structure under "Test1 [Test1]".
 - Root: .gradle, .idea, app, gradle, .gitignore, build.gradle, gradle.properties, gradlew, gradlew.bat, local.properties, settings.gradle, Test1.iml
 - External Libraries
 - Scratches and Consoles
- Code Editor:** Displays the file `MainActivity.kt` with the following code:

```
package com.example.test1
import ...
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```
- Build Output:** Shows the build log:

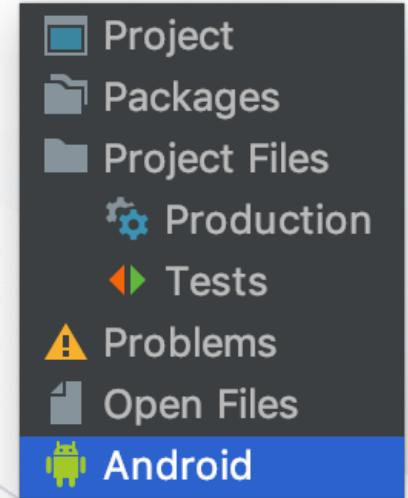
```
Build: Sync x Build Output x
Build: finished at 29/03/2020, 10:16
2 s 455 ms
> Task :app:compileDebugAidl NO-SOURCE
> Task :app:compileDebugRenderscript NO-SOURCE
> Task :app:generateDebugBuildConfig
> Task :app:prepareLintJar
> Task :app:prepareLintJarForPublish
> Task :app:generateDebugSources

BUILD SUCCESSFUL in 2s
4 actionable tasks: 4 executed
```
- Bottom Status Bar:** Shows "Gradle build finished in 2 s 556 ms", "13:1 LF UTF-8 4 spaces", and icons for Terminal, Build, Logcat, TODO, Event Log, and Device File Explorer.

Zmiana widoku (1)



```
Test12 [Test1] ~/IntelliJIDEAProjects/Test12 - .../app/src/main/java/com/example/test1/MainActivity.kt [app]
Click
Project Manager
Resource Manager
1: Project Test12 [Test1] ~/IntelliJIDEAProjects/Test12
2: Favorites
External Libraries
Scratches and Consoles
Build Variants
Build: Sync x Build Output x
Build: finished at 29/03/2020, 10:16 2 s 455 ms
> Task :app:compileDebugAidl NO-SOURCE
> Task :app:compileDebugRenderscript NO-SOURCE
> Task :app:generateDebugBuildConfig
> Task :app:prepareLintJar
> Task :app:prepareLintJarForPublish
> Task :app:generateDebugSources
BUILD SUCCESSFUL in 2s
4 actionable tasks: 4 executed
Event Log
13:1 LF UTF-8 4 spaces
Terminal Build Logcat TODO
Gradle build finished in 2 s 556 ms (3 minutes ago)
```



Zmiana widoku (2)

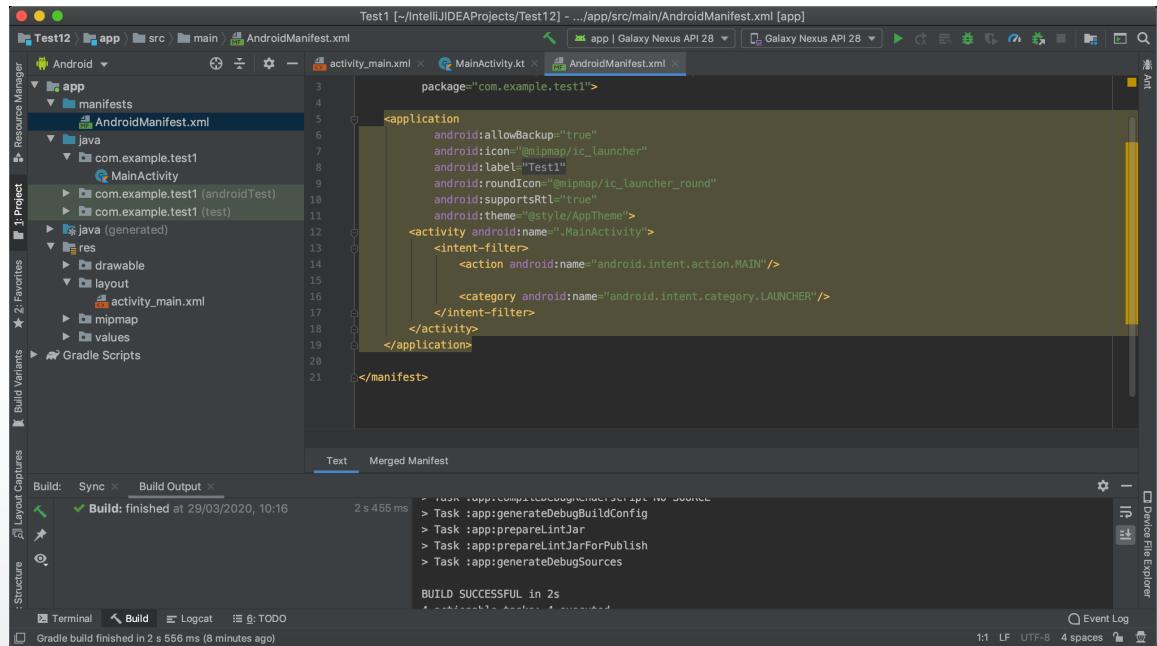
The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The project is named "Test1". The "app" module contains "src/main/java/com/example/test1/MainActivity.kt" and "src/main/res/layout/activity_main.xml".
- MainActivity.kt:** The code defines a class `MainActivity` that extends `AppCompatActivity`. It overrides the `onCreate` method to call `super.onCreate(savedInstanceState)` and set the content view to `R.layout.activity_main`.
- activity_main.xml:** The XML layout file defines a single `FrameLayout` element with the id `activity_main`.
- Build Output:** The build log shows the following tasks completed:
 - > Task :app:compileDebugAidl NO-SOURCE
 - > Task :app:compileDebugRenderscript NO-SOURCE
 - > Task :app:generateDebugBuildConfig
 - > Task :app:prepareLintJar
 - > Task :app:prepareLintJarForPublish
 - > Task :app:generateDebugSources

BUILD SUCCESSFUL in 2s
4 actionable tasks: 4 executed
- Bottom Status Bar:** Shows "Gradle build finished in 2 s 556 ms (7 minutes ago)".
- Bottom Right:** Includes a watermark for "dr inż. Andrzej Wilczyński - www.awilczynski.me" and the page number "17".

Manifest aplikacji

Punkt wyjścia do tworzenia każdej aplikacji na system Android. Pierwszą rzeczą jaką wykonuje system operacyjny Androida po uruchomieniu aplikacji jest wczytanie jej manifestu. W pliku tym między innymi należy wskazać, która aktywność jest punktem wejścia do aplikacji.



The screenshot shows the Android Studio interface with the project 'Test1' open. The left sidebar displays the project structure, including the 'app' folder containing 'manifests', 'java', and 'res' directories. The 'AndroidManifest.xml' file is selected in the manifest folder. The main editor window shows the XML code for the manifest:

```
Test1 [~/IntelliJIDEAProjects/Test12] - .../app/src/main/AndroidManifest.xml [app]
app | Galaxy Nexus API 28   Galaxy Nexus API 28
AndroidManifest.xml

<manifest>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="test1"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

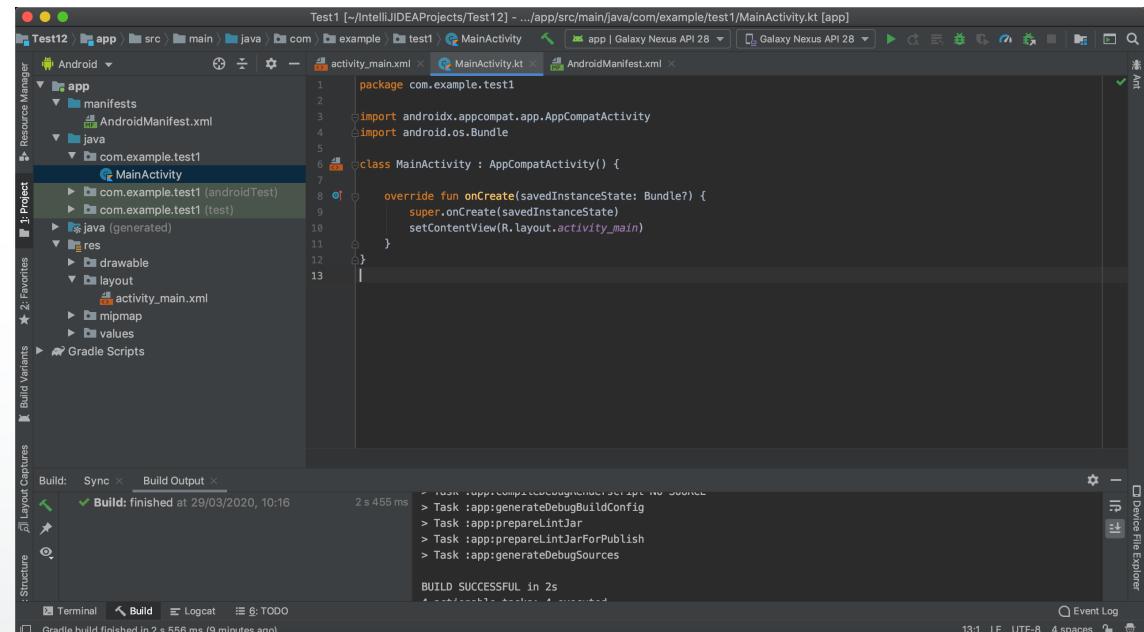
Below the editor, the 'Build Output' tab is active, showing the build log:

```
Build: Sync x Build Output x
Build: finished at 29/03/2020, 10:16
> Task :app:compileDebugSources NO_SOURCE
> Task :app:generateDebugBuildConfig
> Task :app:prepareLintJar
> Task :app:prepareBinJarForPublish
> Task :app:generateDebugSources
BUILD SUCCESSFUL in 2s
Gradle build finished in 2 s 556 ms (8 minutes ago)
```



Kod źródłowy

Kod źródłowy umieszczany jest w folderze *src*, a na wiodku Android jest przedstawiony w elemencie struktury o nazwie *java*. Znajdują się tutaj wszystkie klasy zdefiniowane przez programistę.



```
Test1 [-~/IntelliJIDEAProjects/Test12] - .../app/src/main/java/com/example/test1/MainActivity.kt [app]
Test12 app src main java com example test1 MainActivity activity_main.xml MainActivity.kt AndroidManifest.xml

Resource Manager
  Android
    app
      manifests
        AndroidManifest.xml
      java
        com.example.test1
          MainActivity
          MainActivity.java (selected)
          com.example.test1 (androidTest)
          com.example.test1 (test)
        java (generated)
        res
        drawable
        layout
          activity_main.xml
        mipmap
        values
      Gradle Scripts

Project
  1: Project
  2: Favorites
  Build variants

Build: Sync Build Output
  Build: finished at 29/03/2020, 10:16 2 s 455 ms
  > Task :app:compileDebugSources NO SOURCE
  > Task :app:generateDebugBuildConfig
  > Task :app:prepareLintJar
  > Task :app:prepareLintJarForPublish
  > Task :app:generateDebugSources
  BUILD SUCCESSFUL in 2s
  13:1 LF UTF-8 4 spaces
  Event Log

Structure
  Terminal Build Logcat TODO
  Gradle build finished in 2 s 556 ms (9 minutes ago)

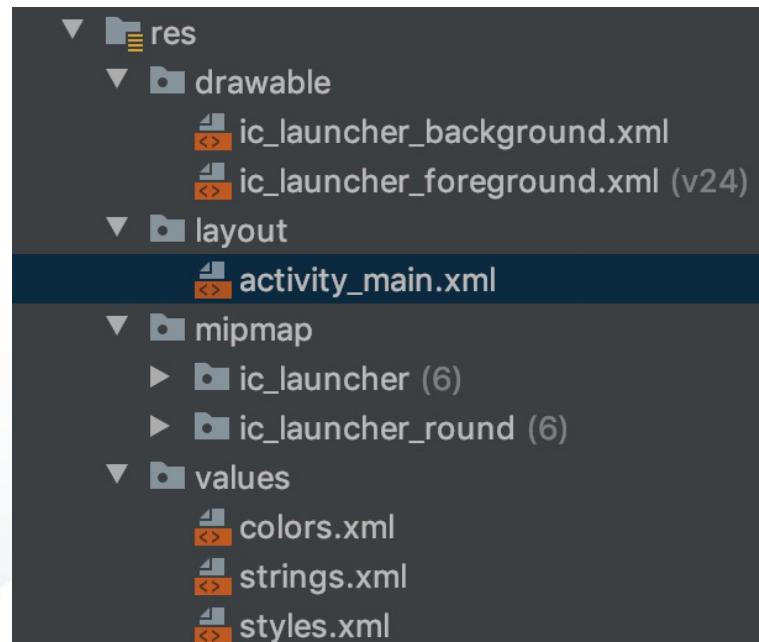
Layout Captures
  Device Emulator

File Explorer
```

The screenshot shows the IntelliJ IDEA interface with the project 'Test1' open. The 'app' module is selected. In the 'Resource Manager' panel, the 'Android' tab is chosen, showing the project structure. The 'app' folder contains 'manifests', 'java', 'res', 'drawable', and 'layout' subfolders. Inside 'java', there's a package 'com.example.test1' containing 'MainActivity' and other test-related classes. The 'activity_main.xml' layout file is shown under 'layout'. The 'Build Output' tab at the bottom indicates a successful build. A hand holding a smartphone is visible at the bottom of the slide, displaying a dashboard with various charts and data.

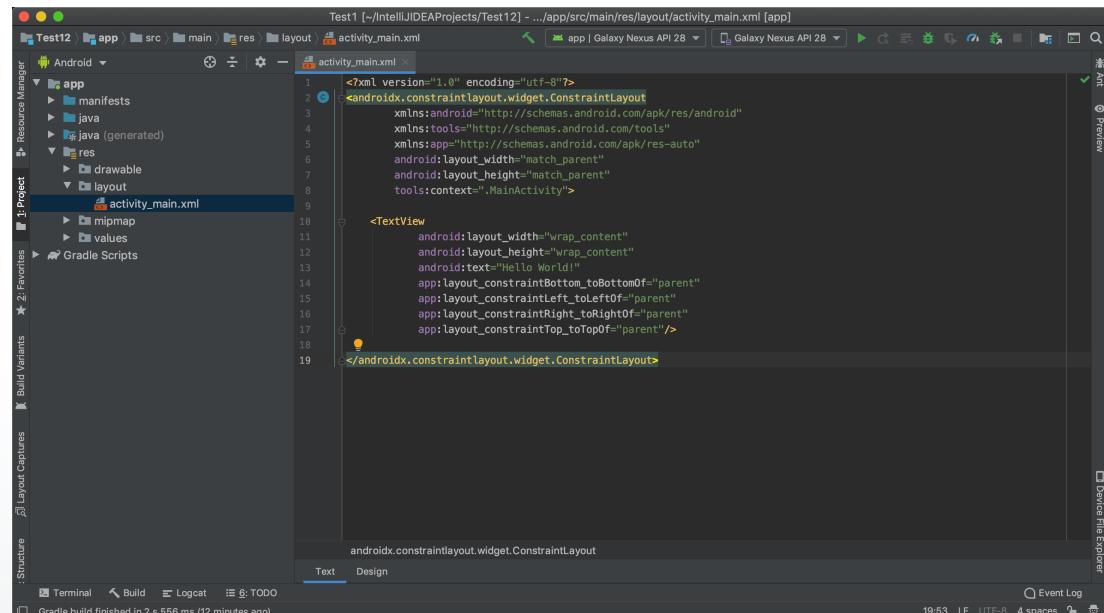
Zasoby

Zasoby to między innymi obrazki używane w aplikacji, różne wersje językowe tekstu lub statyczne wartości dowolnego typu zapisane poza kodem aplikacji.



Układ

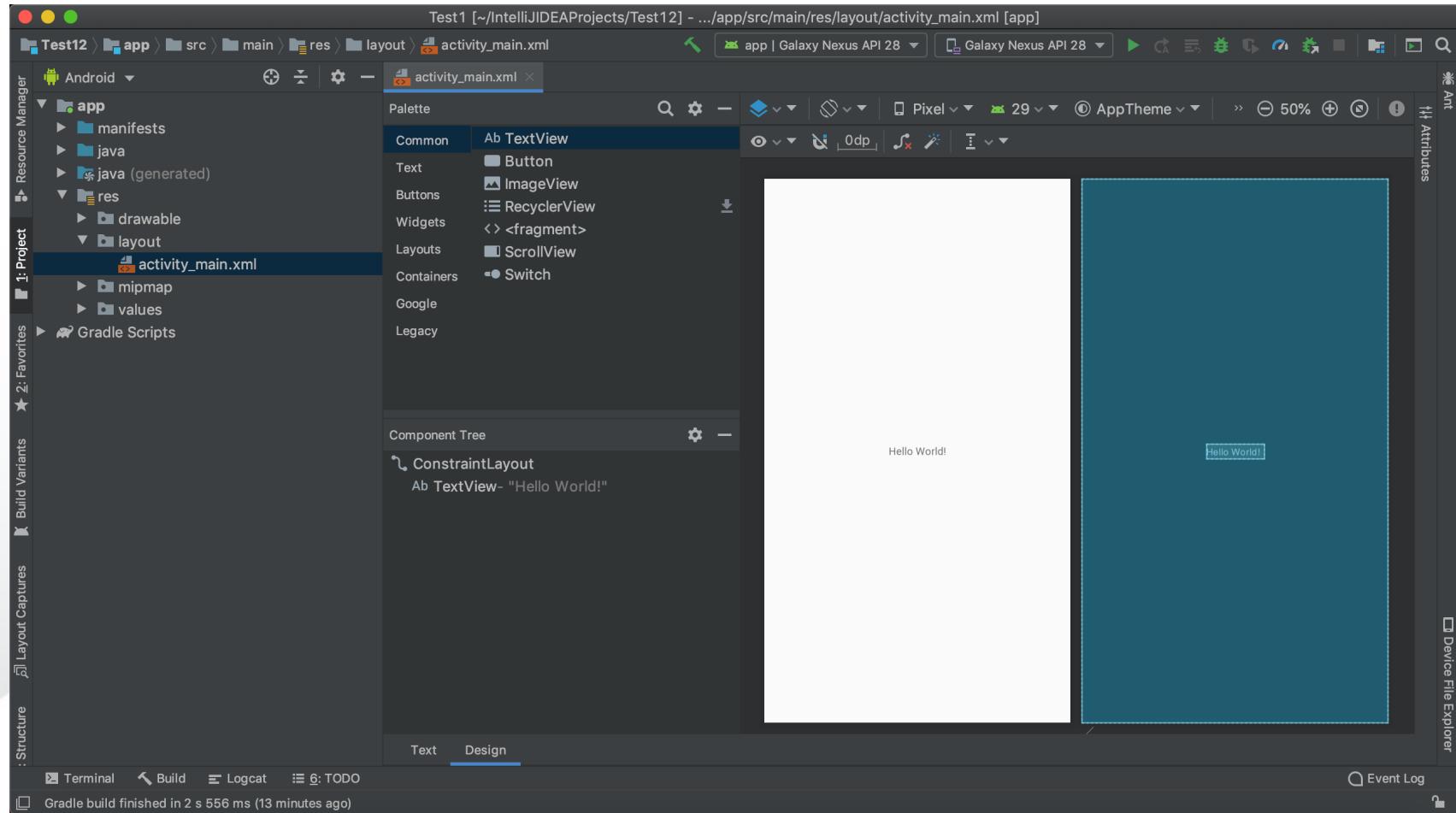
Zasób, który służy do projektowania ekranów i tworzenia różnych elementów interfejsu użytkownika w Androidzie. Zasób układu definiowany jest jako plik XML.



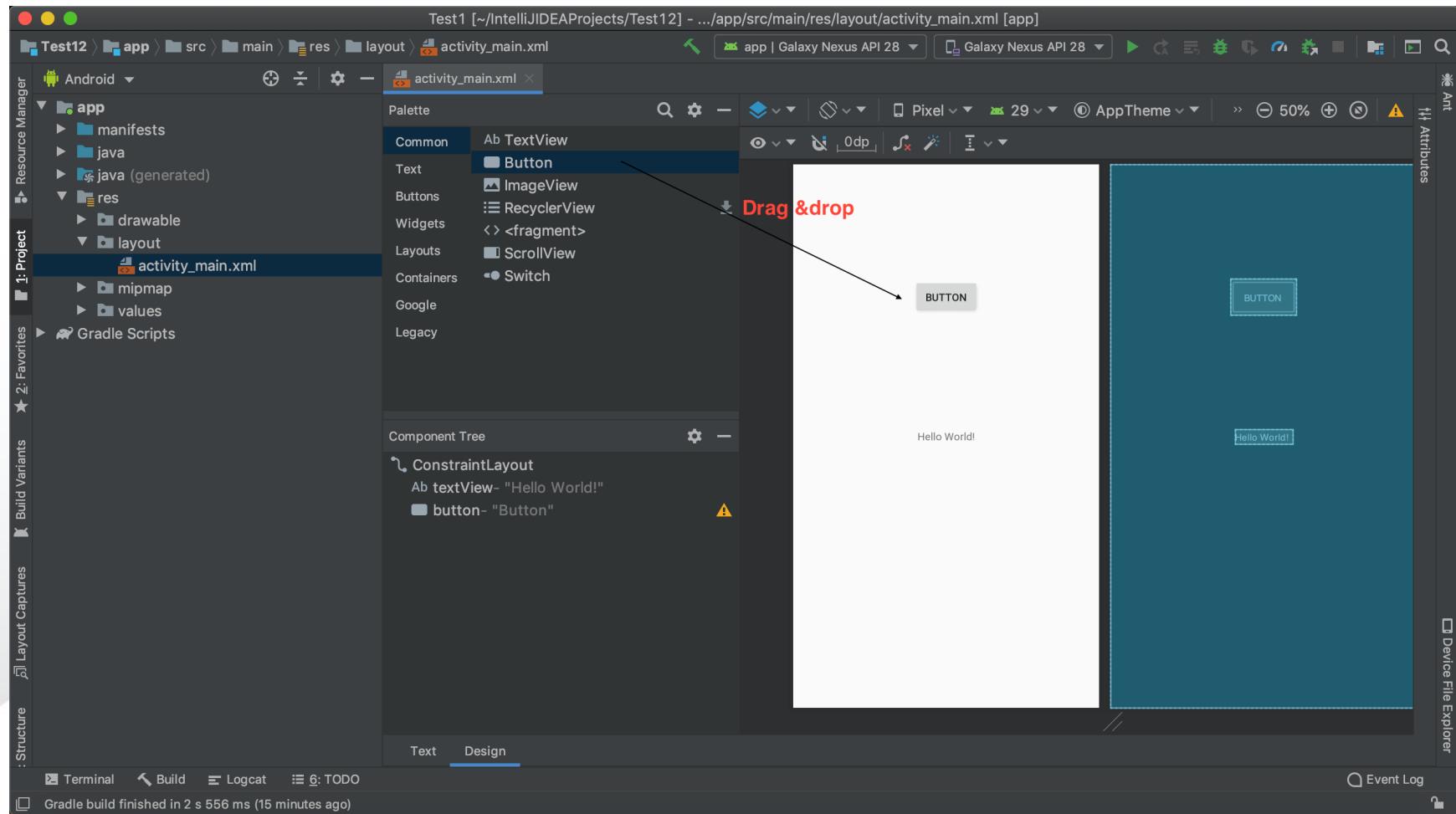
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

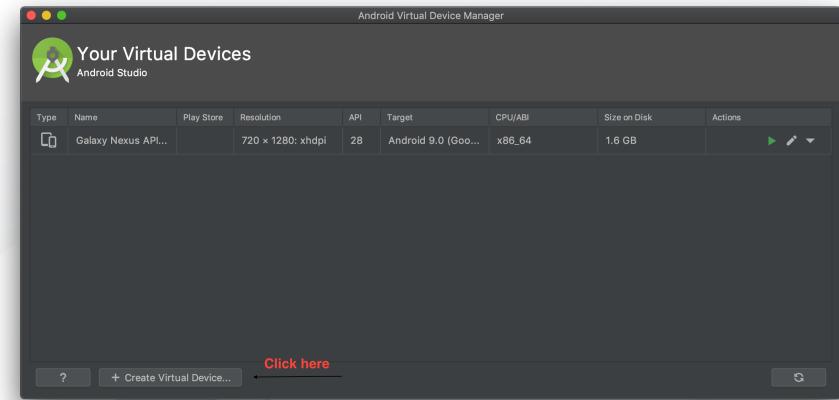
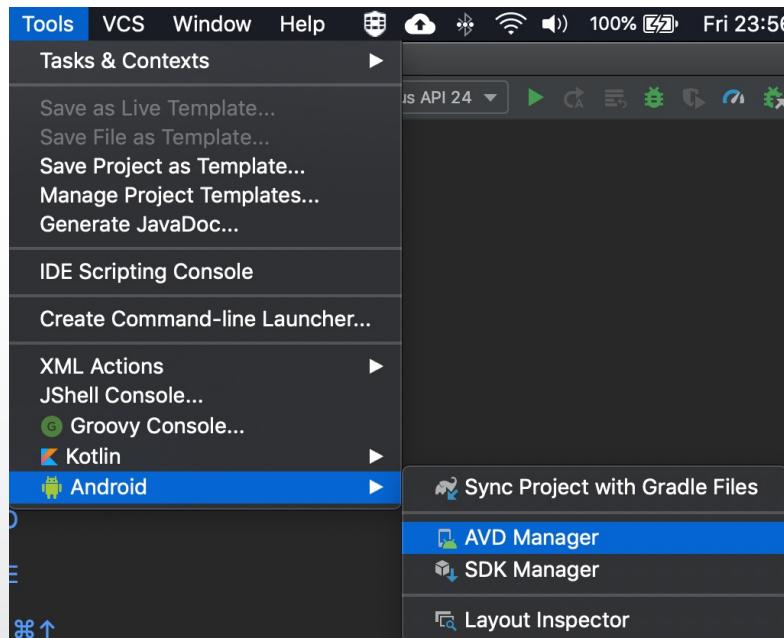
Zmiana UI z Text na Design



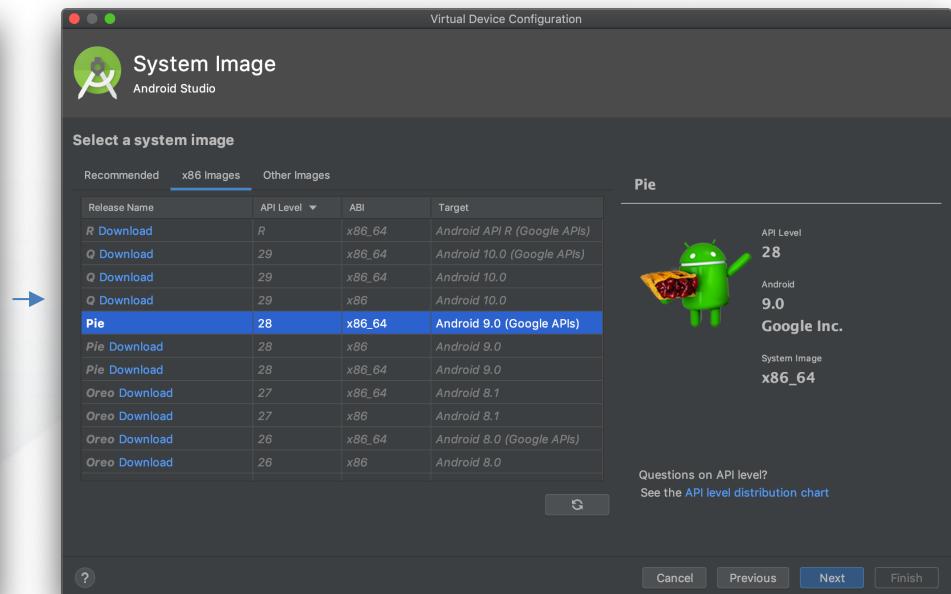
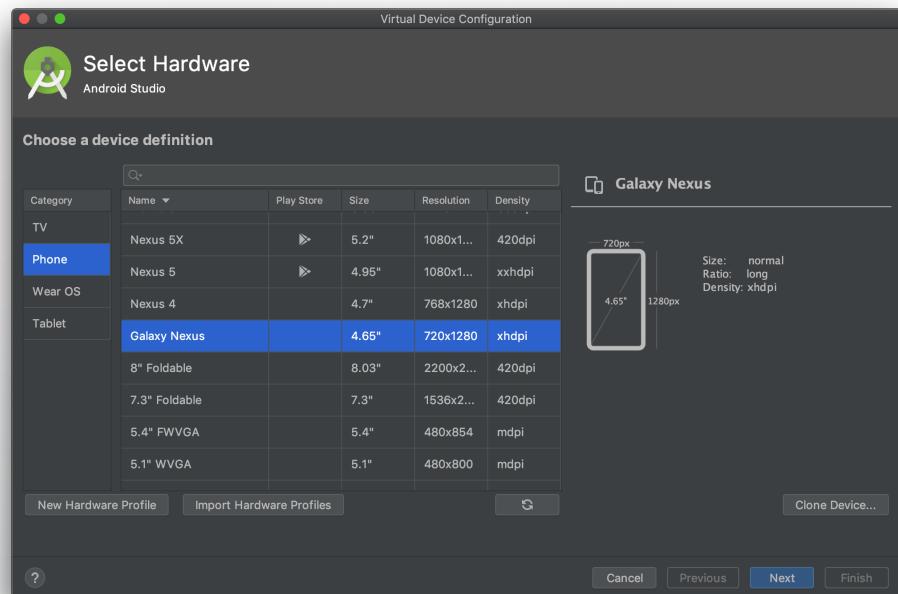
Korzystanie z palety



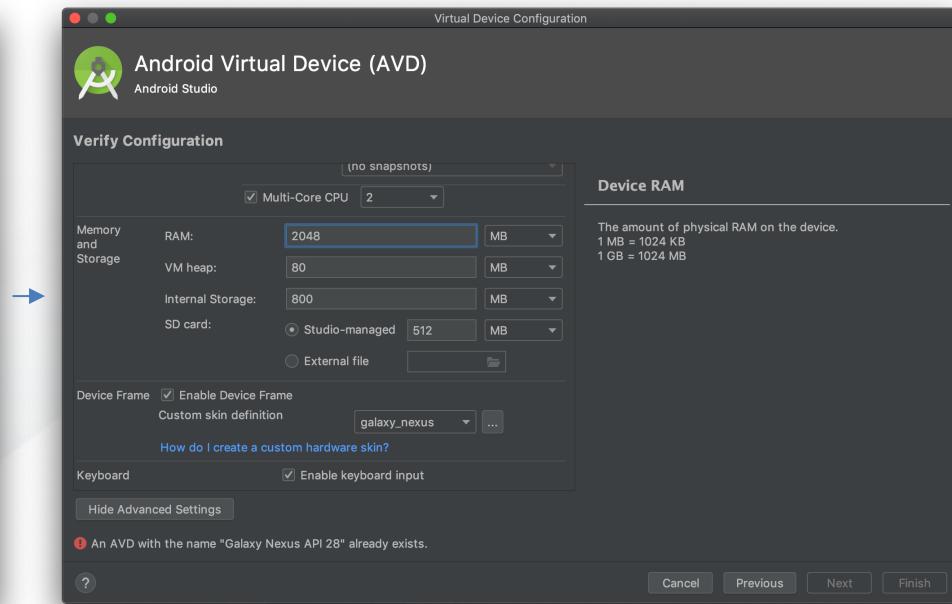
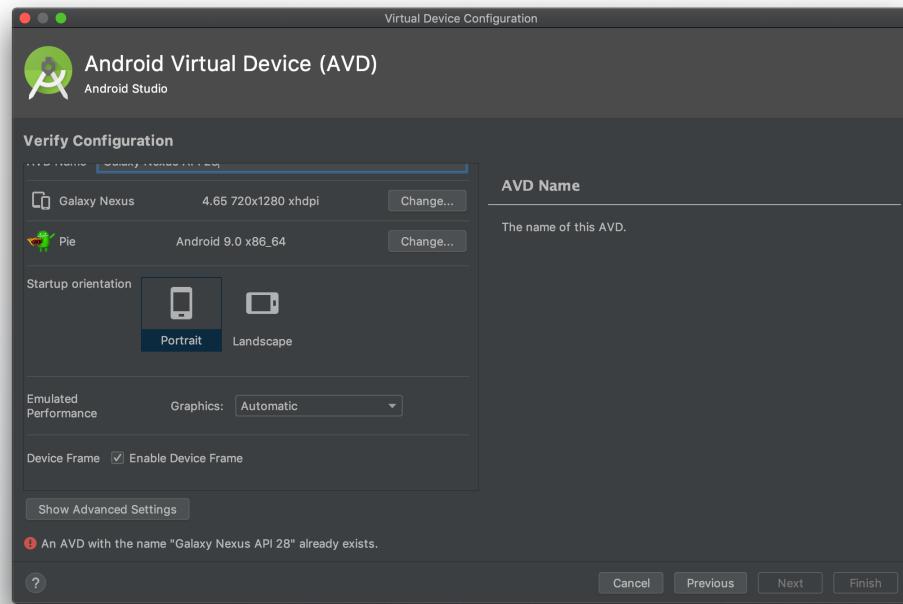
Konfigurowanie AVD (1)



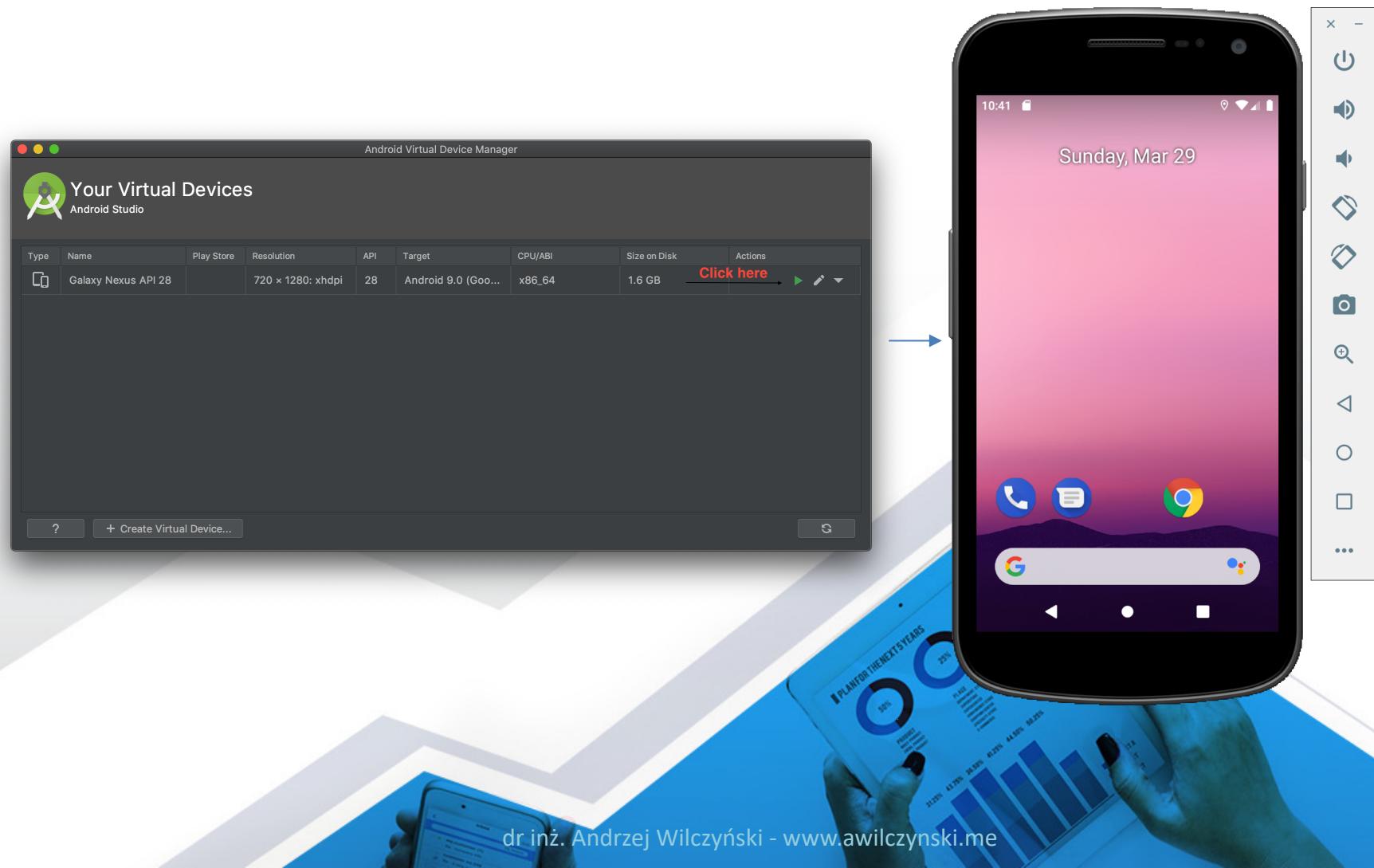
Konfigurowanie AVD (2)



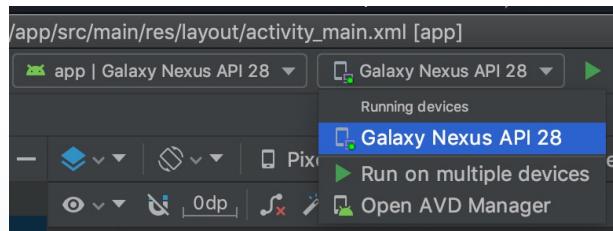
Konfigurowanie AVD (3)



Uruchamianie emulatora



Uruchamianie aplikacji na emulatorze (1)



The screenshot shows the IntelliJ IDEA interface. The code editor displays the file `MainActivity.kt` with the following code:

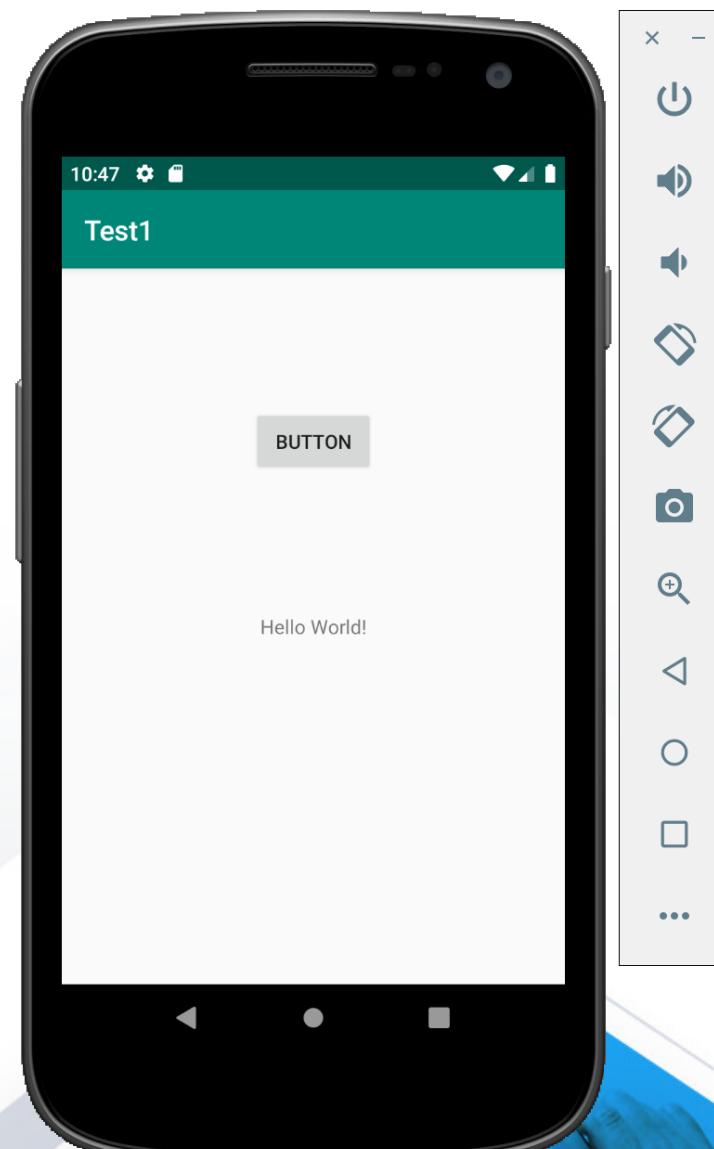
```
Test1 [~/IntelliJIDEAProjects/Test12] - .../app/src/main/java/com/example/test1/MainActivity.kt [app]
Test12 app src main java com example test1 MainActivity.kt
Android app manifests java com.example.test1 MainActivity
MainActivity.java
MainActivity.kt
MainActivity.xml
activity_main.xml
MainActivit...
MainActivity
MainActivity : AppCompatActivity()
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
}
```

The Device File Explorer panel on the right shows a tree structure with nodes like `Device`, `File Explorer`, `File System`, and `File Watchers`. A red arrow points from the 'Running devices' dropdown in the first screenshot to the 'Device' node in the Device File Explorer panel here.

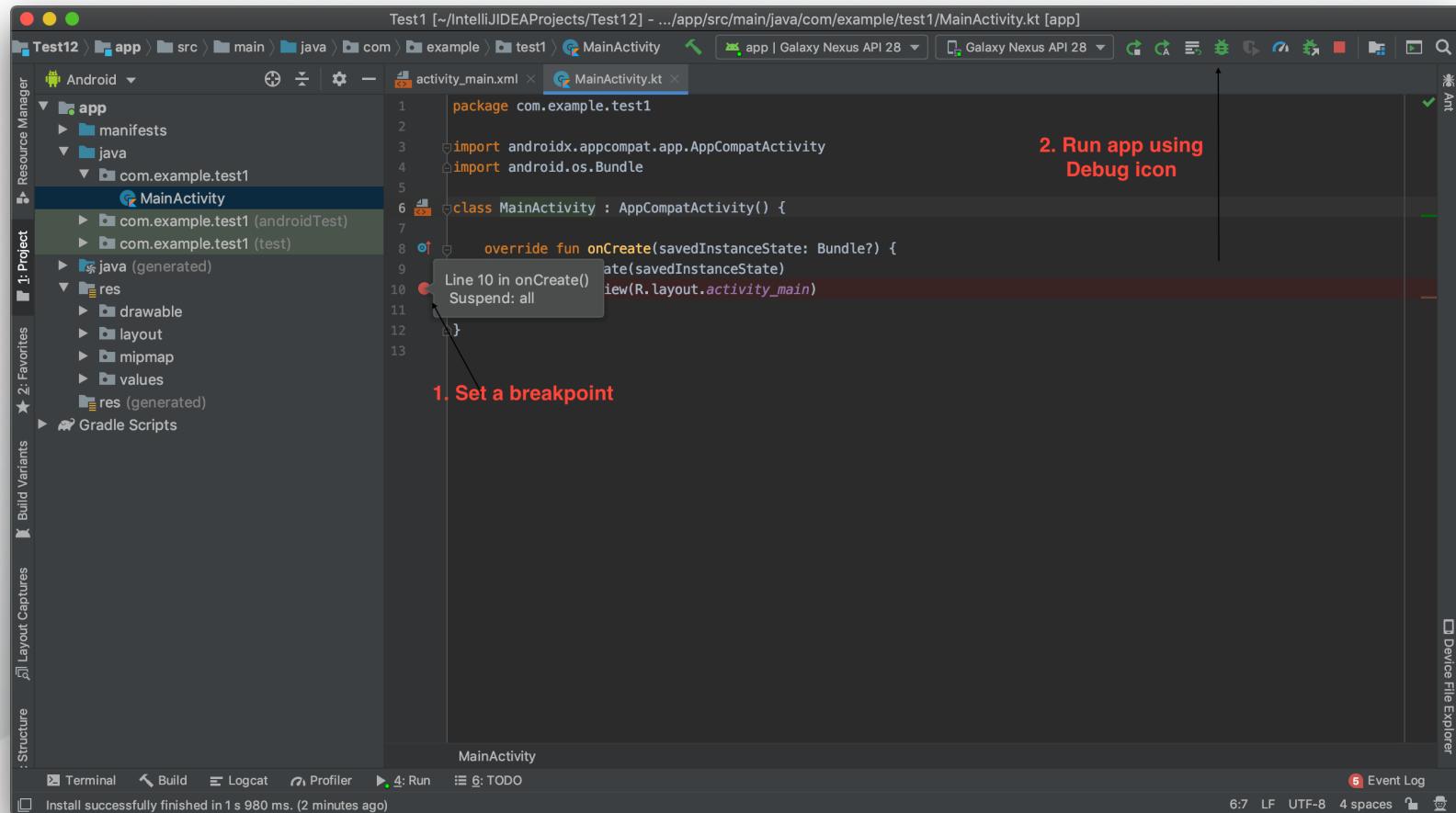
Uruchamianie aplikacji na emulatorze (2)

Po zmianach w kodzie nie musisz uruchamiać na nowo emulatora!

Wystarczy, że na nowo wykonasz deploy aplikacji.



Debugowanie aplikacji (1)



Debugowanie aplikacji (2)

Test1 [~/IntelliJIDEAProjects/Test12] - .../app/src/main/java/com/example/test1/MainActivity.kt [app]

Resource Manager

1: Project

2: Favorites

Android Manager

Activity Manager

Gradle Scripts

activity_main.xml

MainActivity.kt

Activity.java

```
1 package com.example.test1
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5
6 class MainActivity : AppCompatActivity() {
7
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_main)
11    }
12}
```

MainActivity > onCreate()

Debug: app

Debugger

Frames

Variables

Collecting data...

"main"@10,636 in group "main": RUNNING

onCreate:10, MainActivity (com.example.test1)

performCreate:7136, Activity (android.app)

performCreate:7127, Activity (android.app)

callActivityOnCreate:1271, Instrumentation (android.app)

performLaunchActivity:2893, ActivityThread (android.app)

handleLaunchActivity:3048, ActivityThread (android.app)

execute:78, LaunchActivityItem (android.app.servertransaction)

executeCallbacks:108, TransactionExecutor (android.app.servertransaction)

execute:68, TransactionExecutor (android.app.servertransaction)

handleMessage:1808, ActivityThread\$H (android.app)

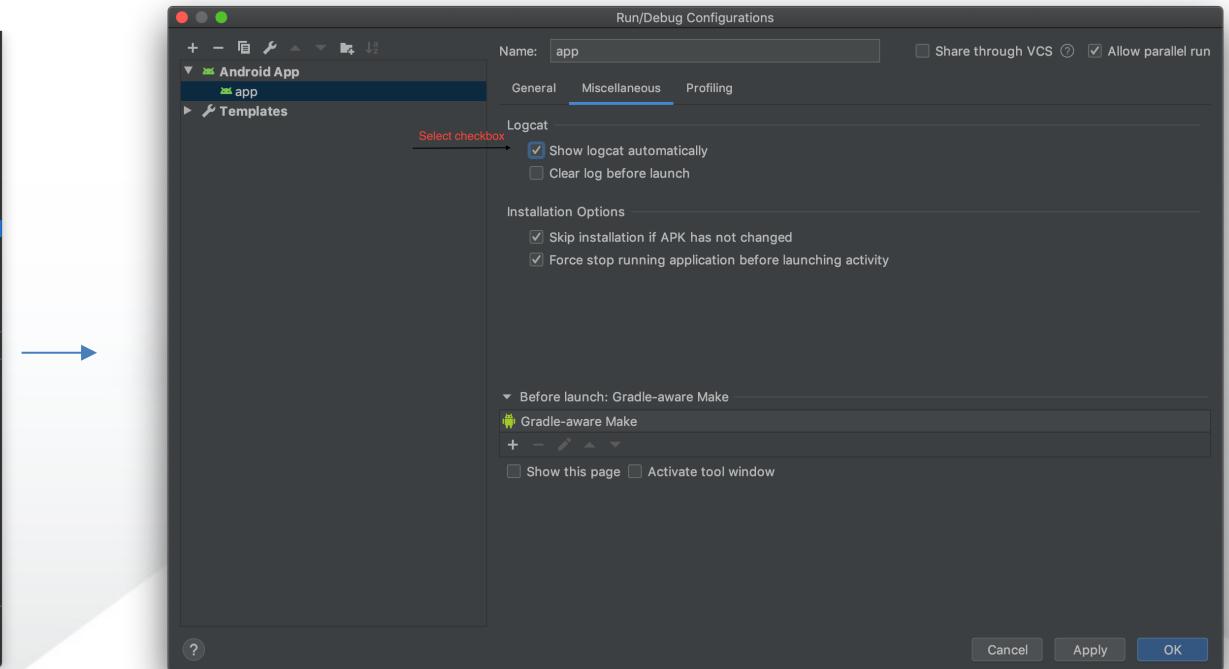
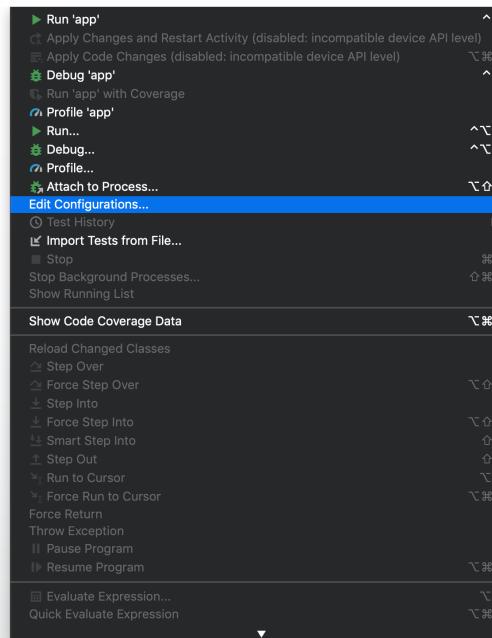
dispatchMessage:106, Handler (android.os)

Terminal Build Logcat Profiler Run Debug TODO

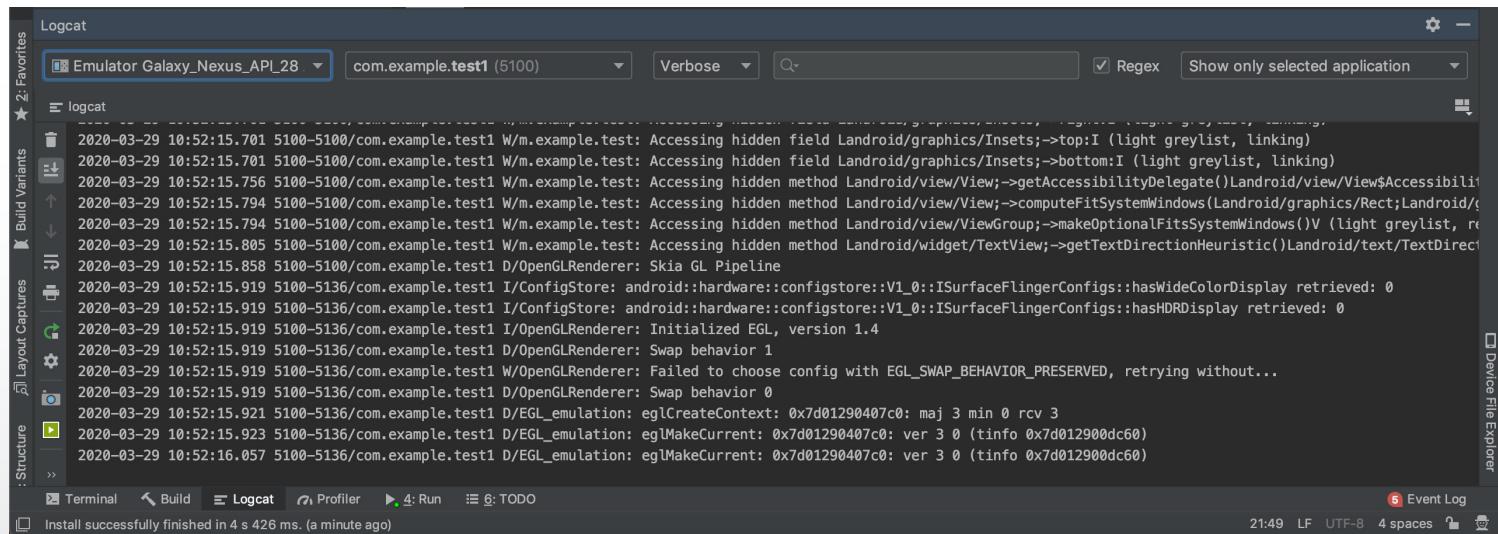
Event Log

Install successfully finished in 2 s 234 ms. (a minute ago)

Automatyczne uruchamianie Logcata w przypadku crashu aplikacji



Korzystanie z Logcata



Dziękuję! ☺



Bibliografia

1. Charlie Collins, Michael Galpin, Matthias Kaepler — *Android w praktyce*, Helion, 2012.

