KamiLimu

12th June 2024

Android Development

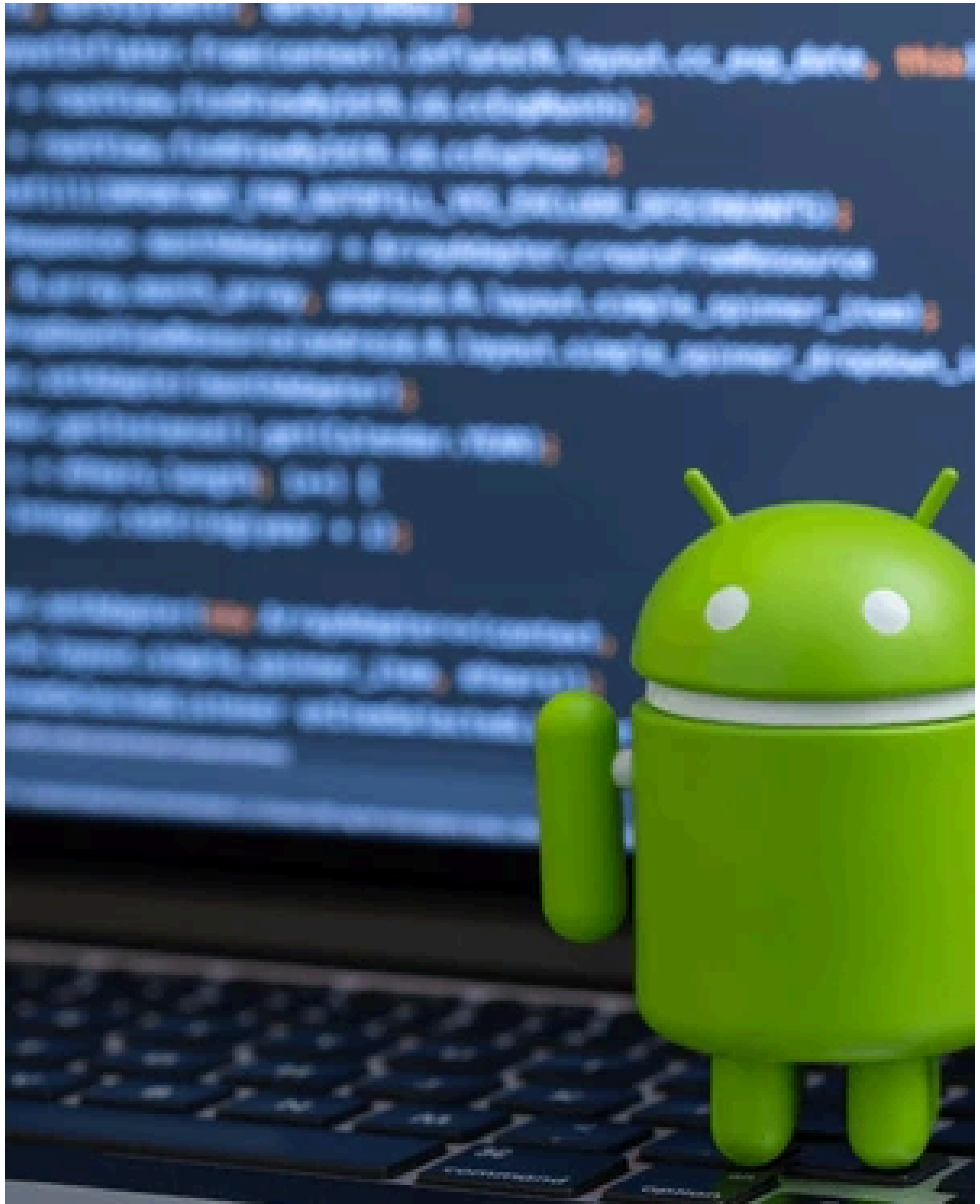
Introduction to Android app development
with Android Studio

KamiLimu

Android Development

What we shall cover

- Environment Setup
- Project structure
- Developer options on an Android device
- Developing on an emulator and Android device
- Connecting to an Android device via ADB
- Debugging



Android Studio

Installation & Configuration

Installation

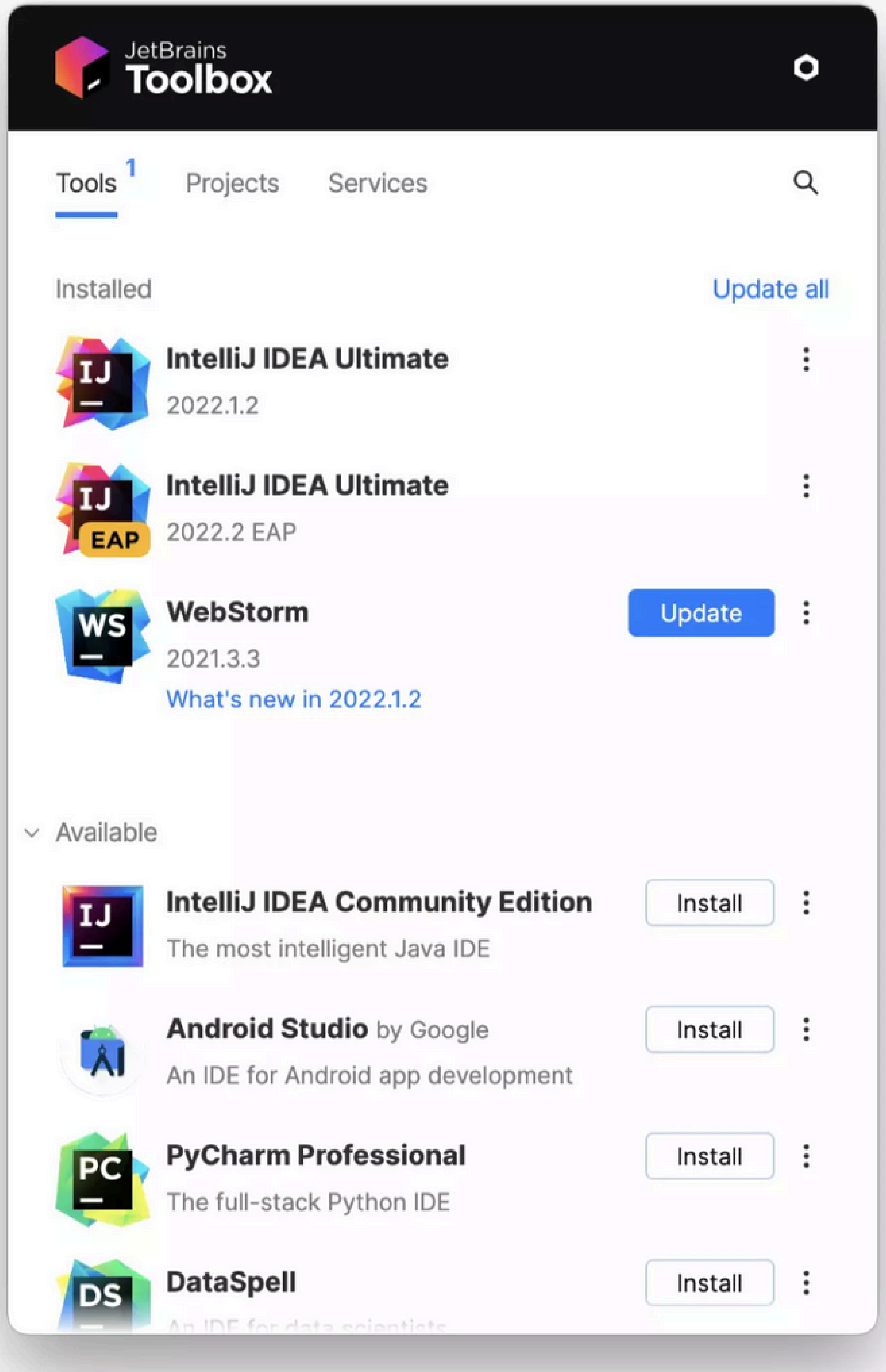
Install JetBrains Toolbox

Install Android Studio within JetBrains Toolbox

Configuration

Launch Android Studio

Update SDK components from the popup



IntelliJ IDEA Ultimate
2022.1.2

IntelliJ IDEA Ultimate
2022.2 EAP

WebStorm
2021.3.3
What's new in 2022.1.2

IntelliJ IDEA Community Edition
The most intelligent Java IDE

Android Studio by Google
An IDE for Android app development

PyCharm Professional
The full-stack Python IDE

DataSpell
An IDE for data scientists

Project Structure

Different display modes on the left panel

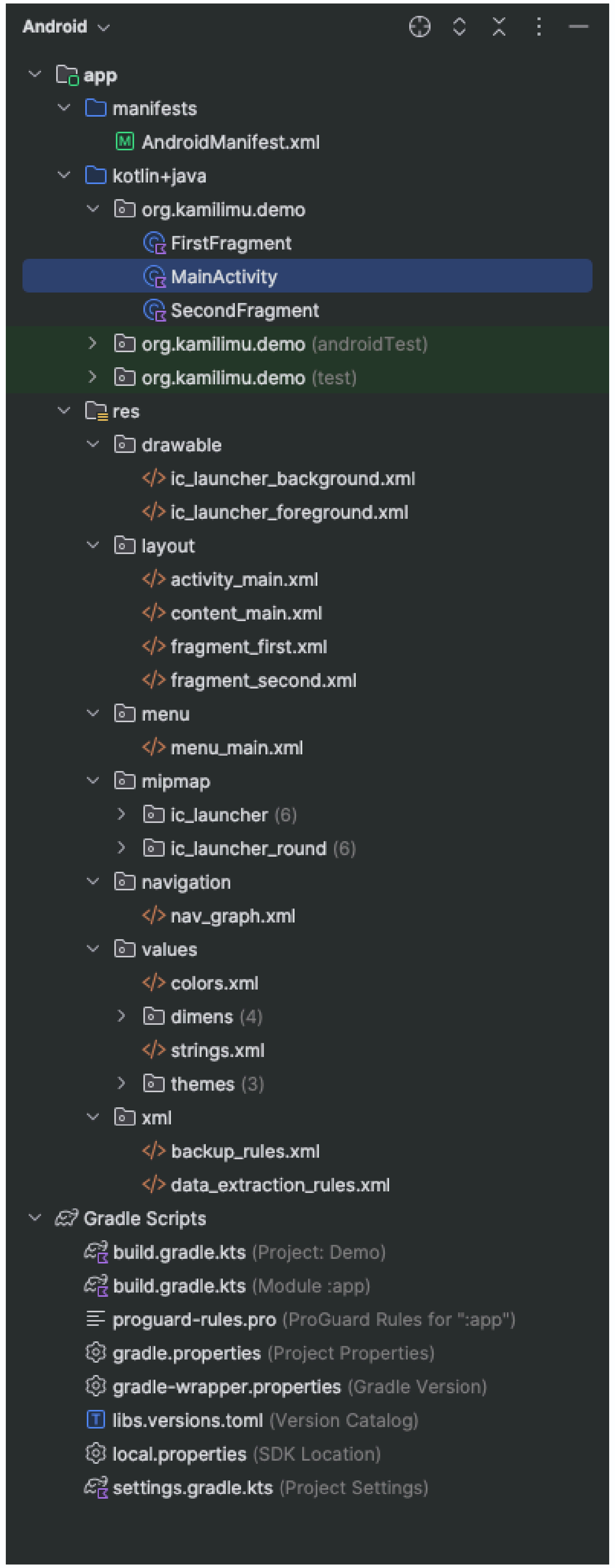
Android view

Not actual file hierarchy

Organized by modules

Directories

1. Manifest
2. Kotlin+Java directory
3. Res directory
4. Gradle scripts



```
app
  manifests
    AndroidManifest.xml
  kotlin+java
    org.kamilimu.demo
      FirstFragment
      MainActivity
      SecondFragment
    org.kamilimu.demo (androidTest)
    org.kamilimu.demo (test)
  res
    drawable
      ic_launcher_background.xml
      ic_launcher_foreground.xml
    layout
      activity_main.xml
      content_main.xml
      fragment_first.xml
      fragment_second.xml
    menu
      menu_main.xml
    mipmap
      ic_launcher (6)
      ic_launcher_round (6)
    navigation
      nav_graph.xml
    values
      colors.xml
      dimens (4)
      strings.xml
      themes (3)
    xml
      backup_rules.xml
      data_extraction_rules.xml
Gradle Scripts
  build.gradle.kts (Project: Demo)
  build.gradle.kts (Module :app)
  proguard-rules.pro (ProGuard Rules for ":app")
  gradle.properties (Project Properties)
  gradle-wrapper.properties (Gradle Version)
  libs.versions.toml (Version Catalog)
  local.properties (SDK Location)
  settings.gradle.kts (Project Settings)
```

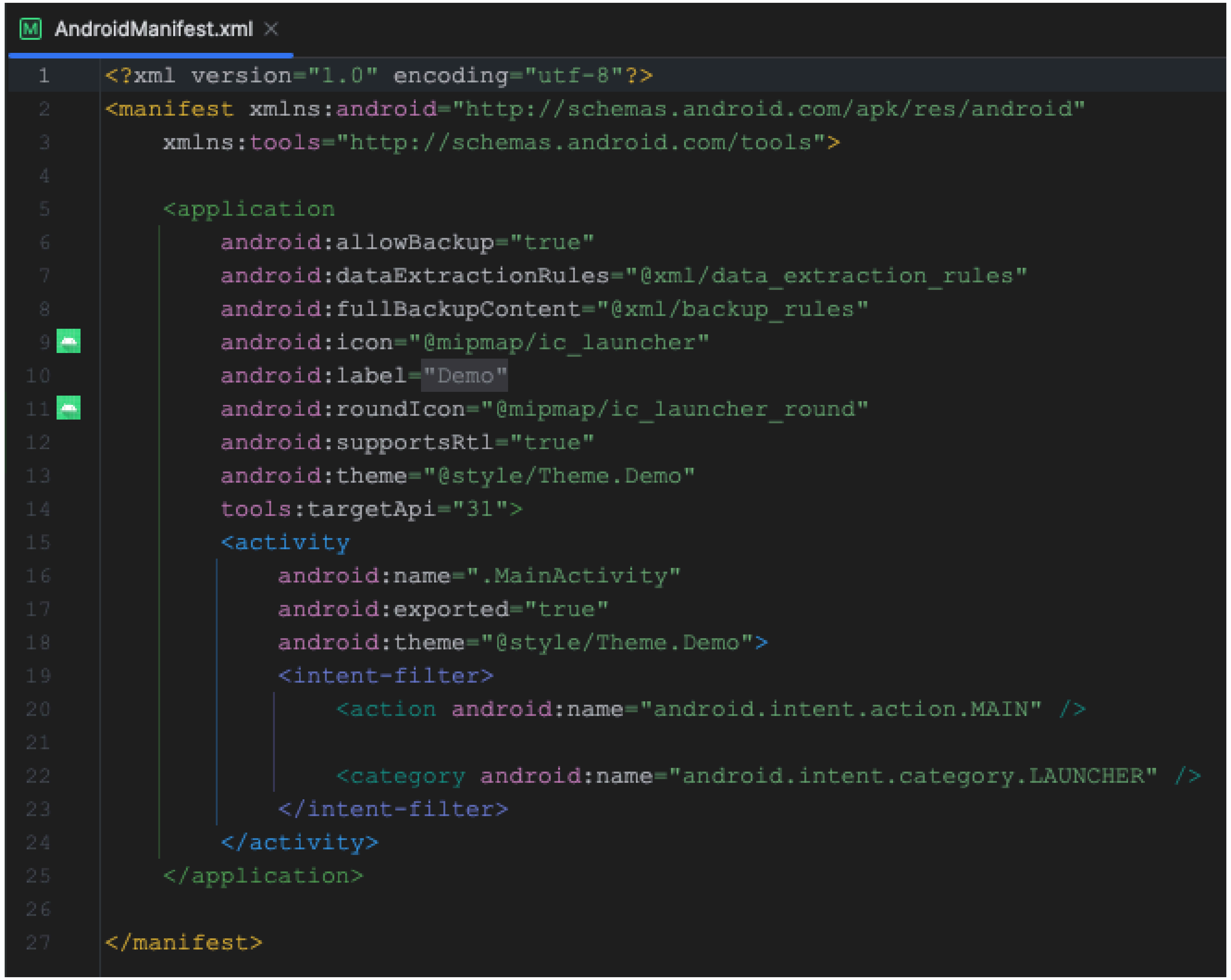
Project Structure

Manifest

Contains app information - Required by

- Android OS
- Android Build Tools
- Google Play

Describes app components



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Demo"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Demo"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:theme="@style/Theme.Demo">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

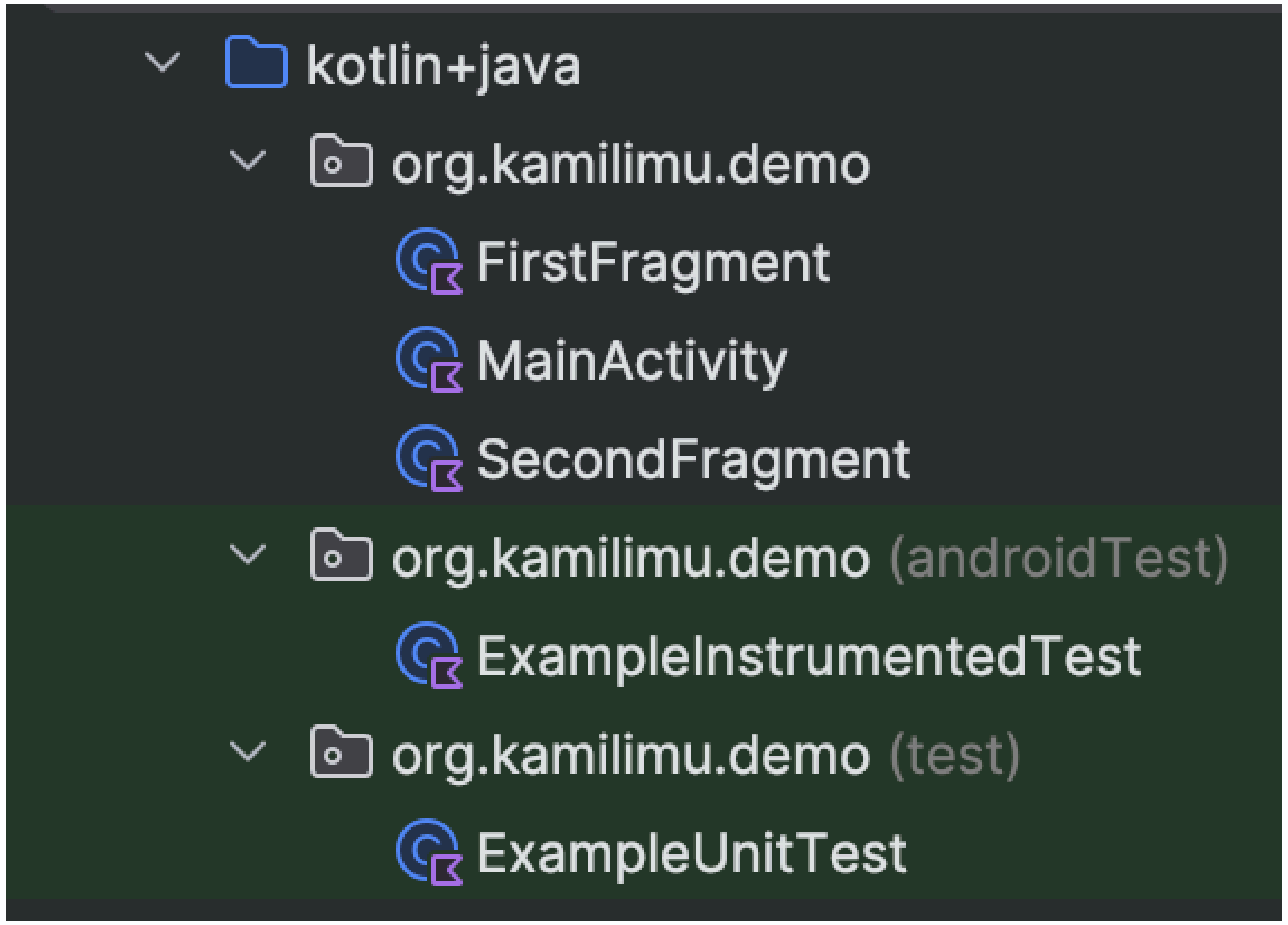
Project Structure

Kotlin+Java Directory

Contains the source code

Divided into packages

Includes test packages (Created by default)



```
kotlin+java
  org.kamilimu.demo
    FirstFragment
    MainActivity
    SecondFragment
  org.kamilimu.demo (androidTest)
    ExampleInstrumentedTest
  org.kamilimu.demo (test)
    ExampleUnitTest
```

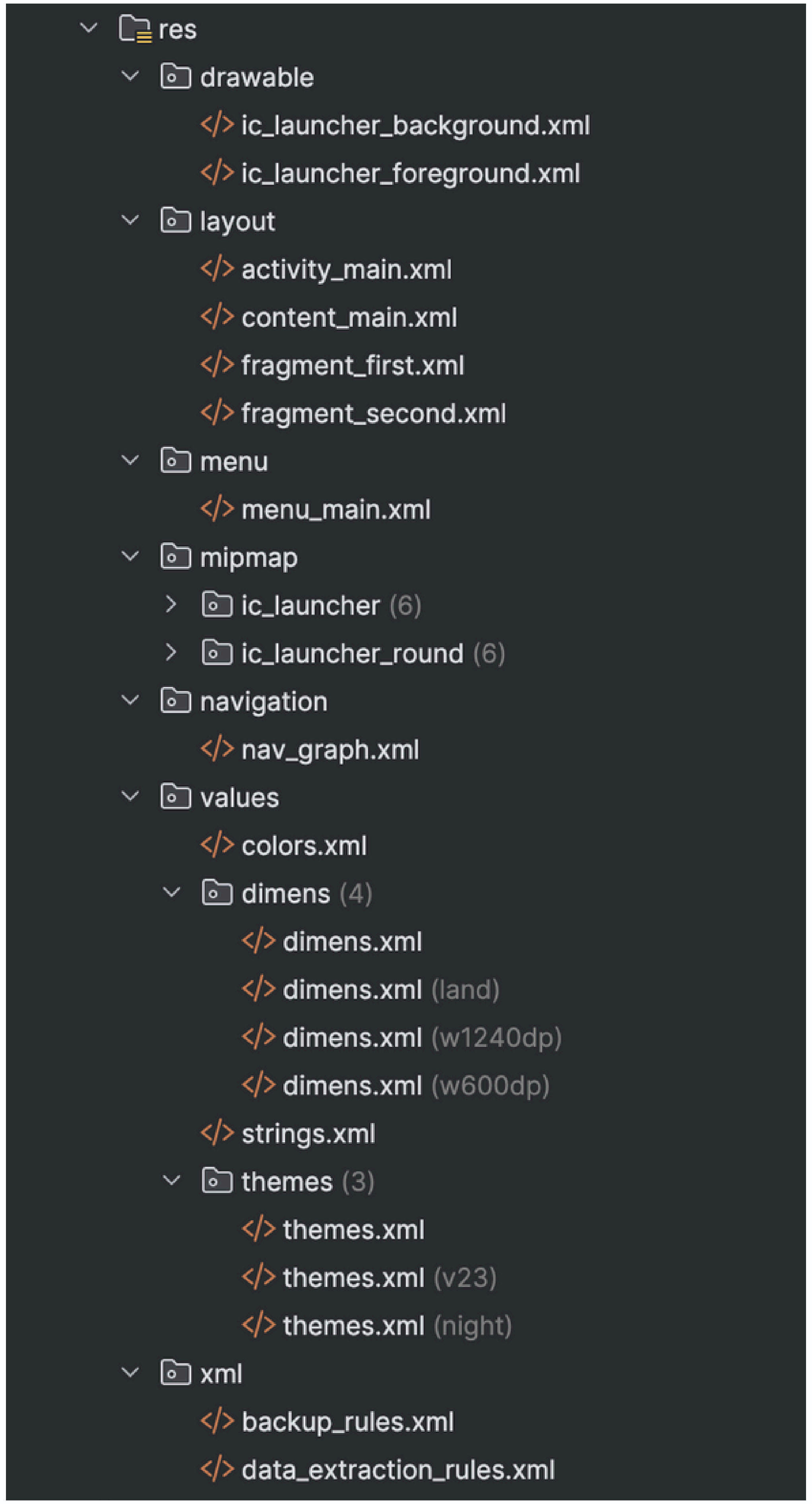
Project Structure

Res Directory

Contains resources needed by the app

The resources include

- Mipmap - Holds launcher icons (Appear on the home screen)
- Drawables - Holds images (png, jpg, etc.)
- Values - Colors, strings, themes
- XML - Holds XML configuration files
- Raw - Holds raw resource files that the app needs in the original form. Examples: .mp3, .mp4, .txt, .json, .html



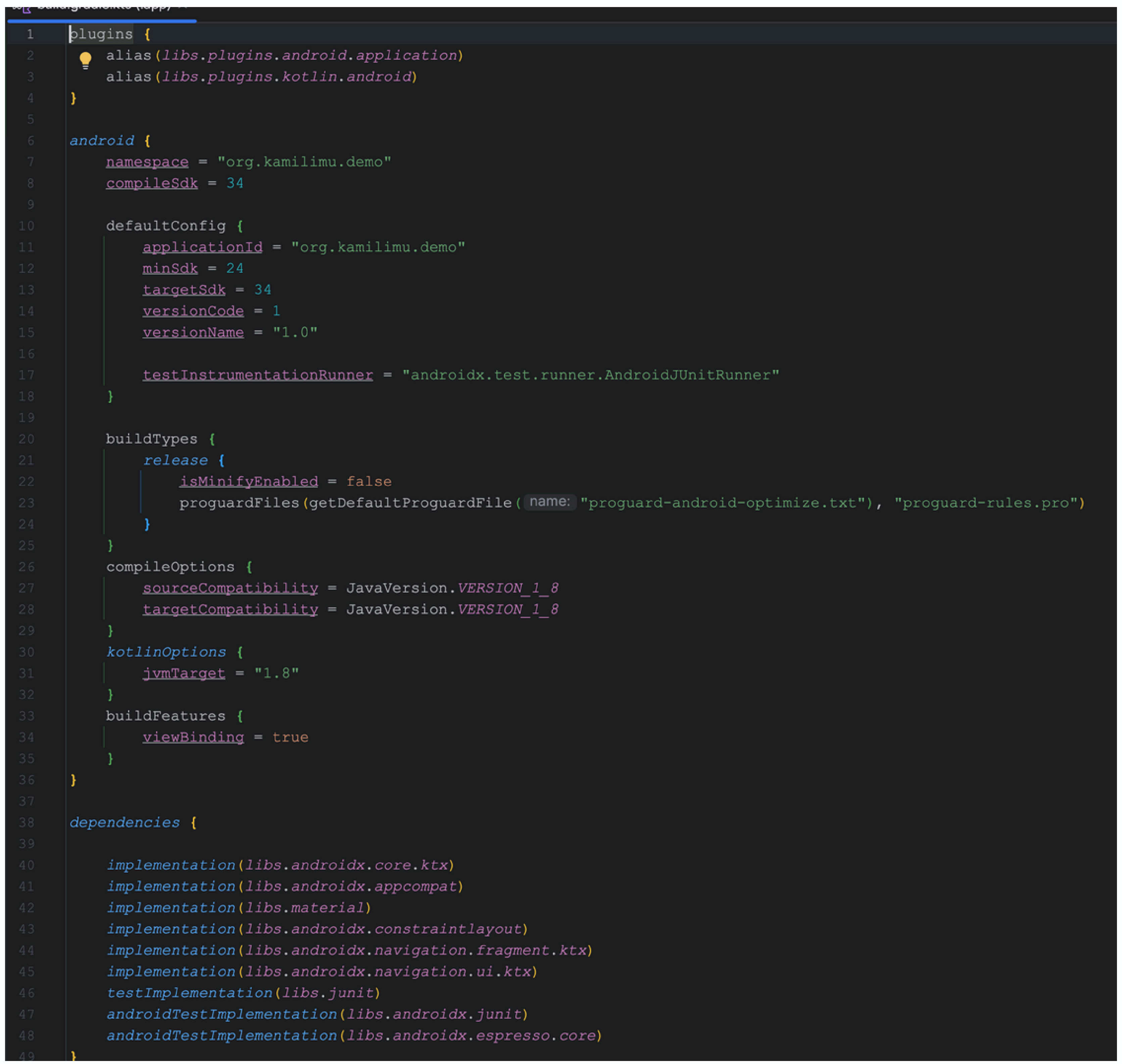
```
res
  drawable
    ic_launcher_background.xml
    ic_launcher_foreground.xml
  layout
    activity_main.xml
    content_main.xml
    fragment_first.xml
    fragment_second.xml
  menu
    menu_main.xml
  mipmap
    ic_launcher (6)
    ic_launcher_round (6)
  navigation
    nav_graph.xml
  values
    colors.xml
    dimens (4)
      dimens.xml
      dimens.xml (land)
      dimens.xml (w1240dp)
      dimens.xml (w600dp)
    strings.xml
    themes (3)
      themes.xml
      themes.xml (v23)
      themes.xml (night)
  xml
    backup_rules.xml
    data_extraction_rules.xml
```

Project Structure

Gradle Scripts

Android uses the Gradle build system to manage the building, packaging, and distribution of applications

The result is an Android App Bundle (aab)



```
plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
}

android {
    namespace = "org.kamilimu.demo"
    compileSdk = 34

    defaultConfig {
        applicationId = "org.kamilimu.demo"
        minSdk = 24
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(getDefaultProguardFile( name: "proguard-android-optimize.txt"), "proguard-rules.pro")
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = "1.8"
    }
    buildFeatures {
        viewBinding = true
    }
}

dependencies {

    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.appcompat)
    implementation(libs.material)
    implementation(libs.androidx.constraintlayout)
    implementation(libs.androidx.navigation.fragment.ktx)
    implementation(libs.androidx.navigation.ui.ktx)
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
}
```

Project Structure

Gradle Scripts

Key Features

- Modular architecture - Organize the project into multiple modules
- Flexible build configurations - Different build variants (debug, release). Different product flavors (demo, production)
- Dependency management - Automatically handles resolution of external libraries
- Incremental builds - Optimizes the build process by only recompiling parts of the project that change between builds
- Task automation - Ability to automate code compiling, APK packaging, running tests and more

Project Structure

Gradle Scripts

Core Components

- Project-level `build.gradle` - Located in the root project directory. Defines configuration options common to all modules
- Module-level `build.gradle` - Located in each module’s directory `app/build.gradle`. Defines configurations specific to the module. Such as dependencies, build types, product flavors
- Gradle Wrapper - A set of scripts that ensure a specific version of Gradle is used to build the project

Project Structure

Gradle Scripts

Key Parts

- Plugins
    - libs.plugins.android.application - Indicates that it is an Android module
    - libs.plugins.kotlin.android - Enables Kotlin support
- Android block
    - compileSdkVersion - Specifies the Android API level used to compile the app
    - defaultConfig - contains default settings for the app. Such as application Id, minimum and target SDK versions, version information
    - buildTypes - defines different build configurations such as `debug` and `release`
- Dependencies block - Lists libraries and modules that the module depends on. The dependencies can be local libraries, remote libraries from repositories or other modules within the project

Developer options

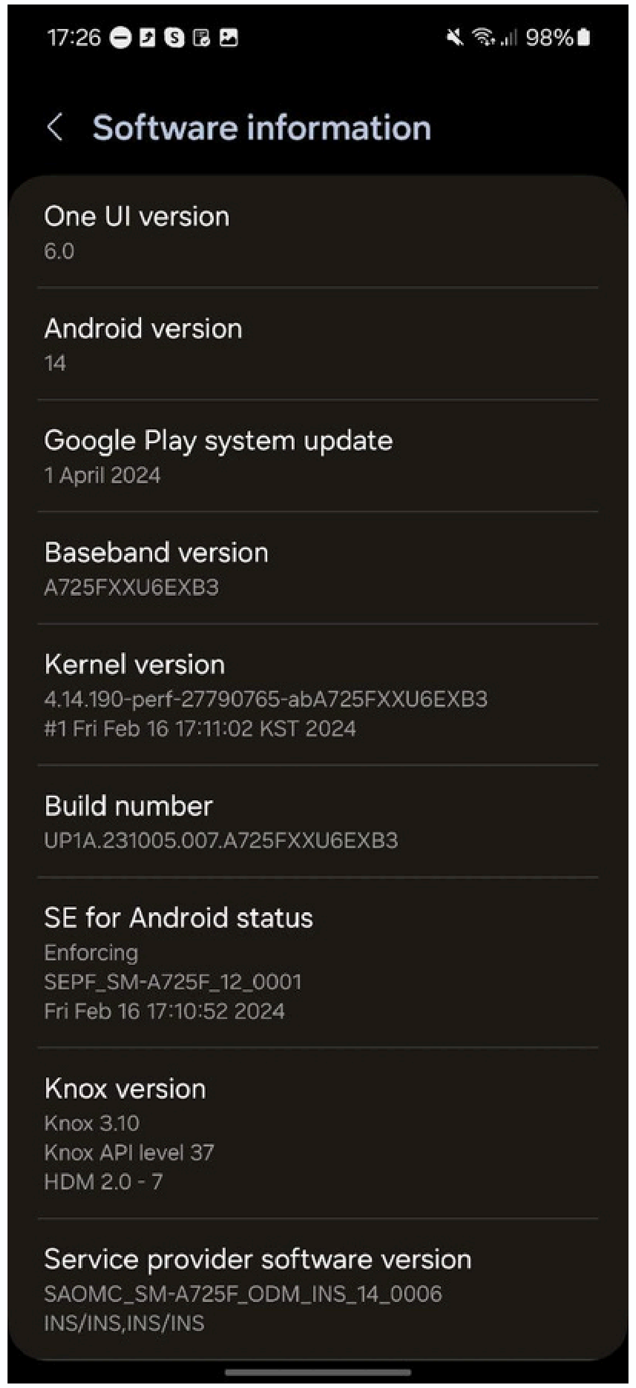
To develop and test on a physical Android device, enable developer options in settings

Enabling developer options

Settings -> About Phone -> Software Information -> Build Number

Tap `Build Number` number 7 times, then enter the phone’s password

Locate developer options in the settings



17:26 98%

Software information

One UI version
6.0

Android version
14

Google Play system update
1 April 2024

Baseband version
A725FXXU6EXB3

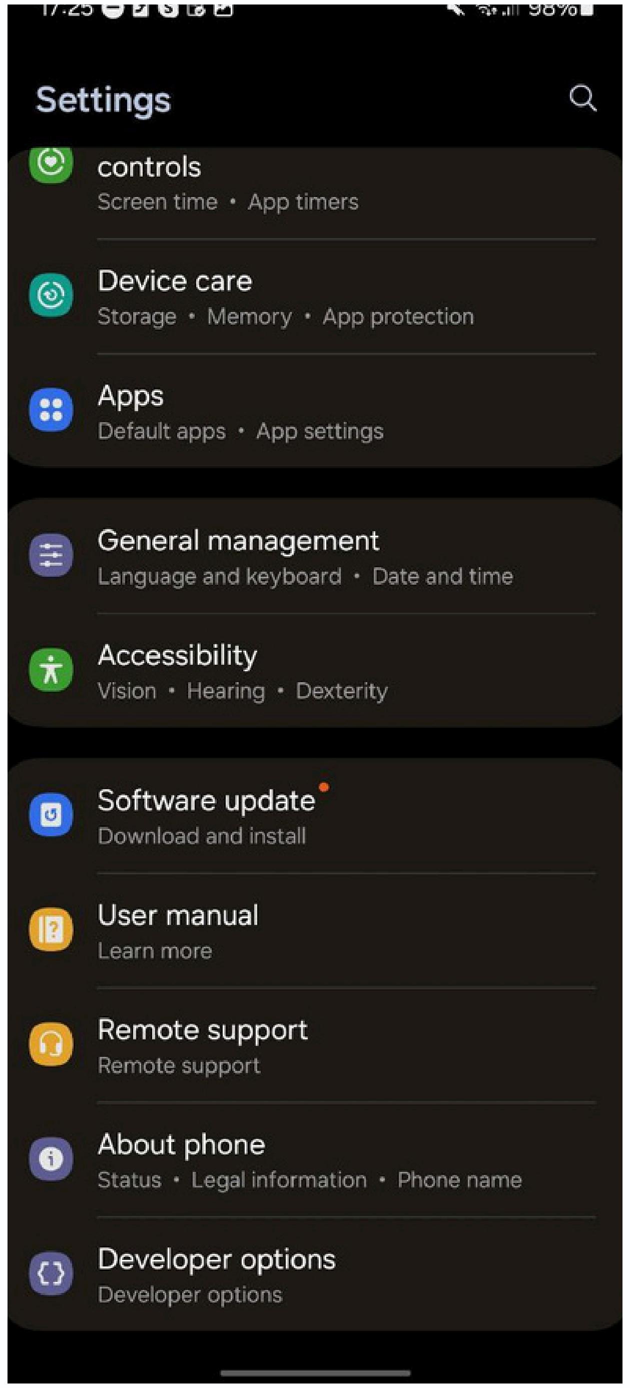
Kernel version
4.14.190-perf-27790765-abA725FXXU6EXB3
#1 Fri Feb 16 17:11:02 KST 2024

Build number
UP1A.231005.007.A725FXXU6EXB3

SE for Android status
Enforcing
SEPF_SM-A725F_12_0001
Fri Feb 16 17:10:52 2024

Knox version
Knox 3.10
Knox API level 37
HDM 2.0 - 7

Service provider software version
SAOMC_SM-A725F_ODM_INS_14_0006
INS/INS,INS/INS



17:25 98%

Settings

controls
Screen time • App timers

Device care
Storage • Memory • App protection

Apps
Default apps • App settings

General management
Language and keyboard • Date and time

Accessibility
Vision • Hearing • Dexterity

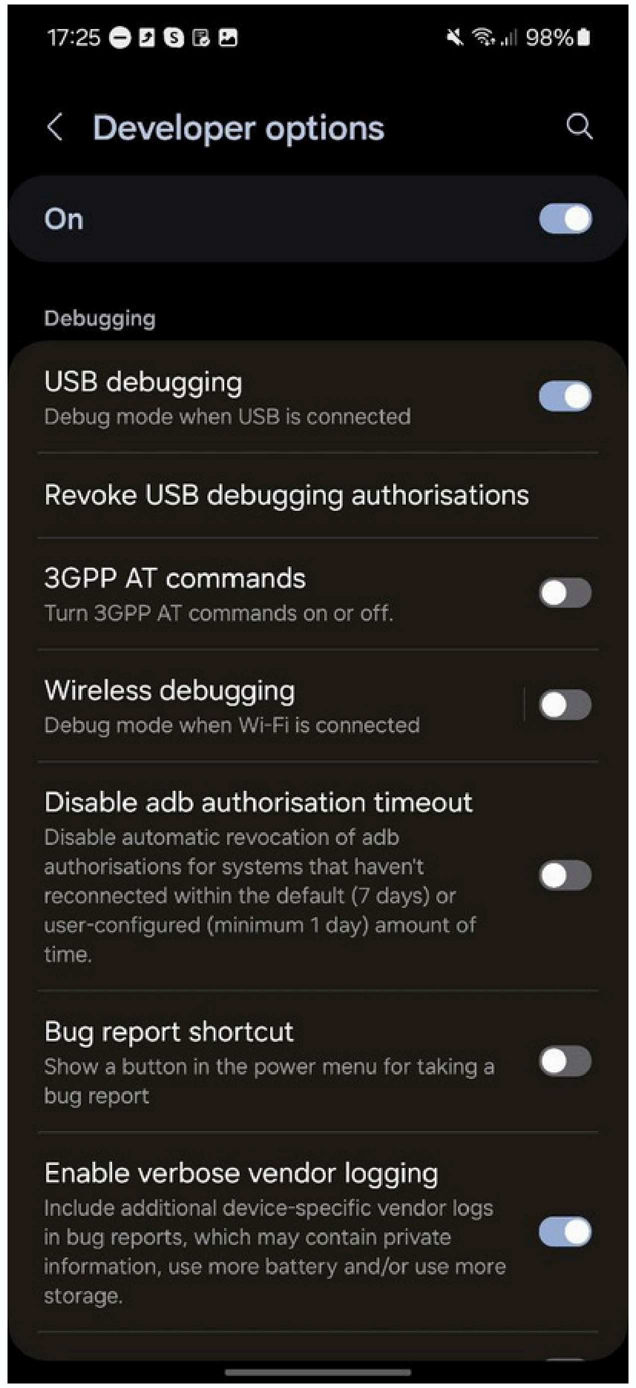
Software update
Download and install

User manual
Learn more

Remote support
Remote support

About phone
Status • Legal information • Phone name

Developer options
Developer options



17:25 98%

Developer options

On ●

Debugging

USB debugging ●
Debug mode when USB is connected

Revoke USB debugging authorisations

3GPP AT commands ○
Turn 3GPP AT commands on or off.

Wireless debugging ○
Debug mode when Wi-Fi is connected

Disable adb authorisation timeout ○
Disable automatic revocation of adb authorisations for systems that haven't reconnected within the default (7 days) or user-configured (minimum 1 day) amount of time.

Bug report shortcut ○
Show a button in the power menu for taking a bug report

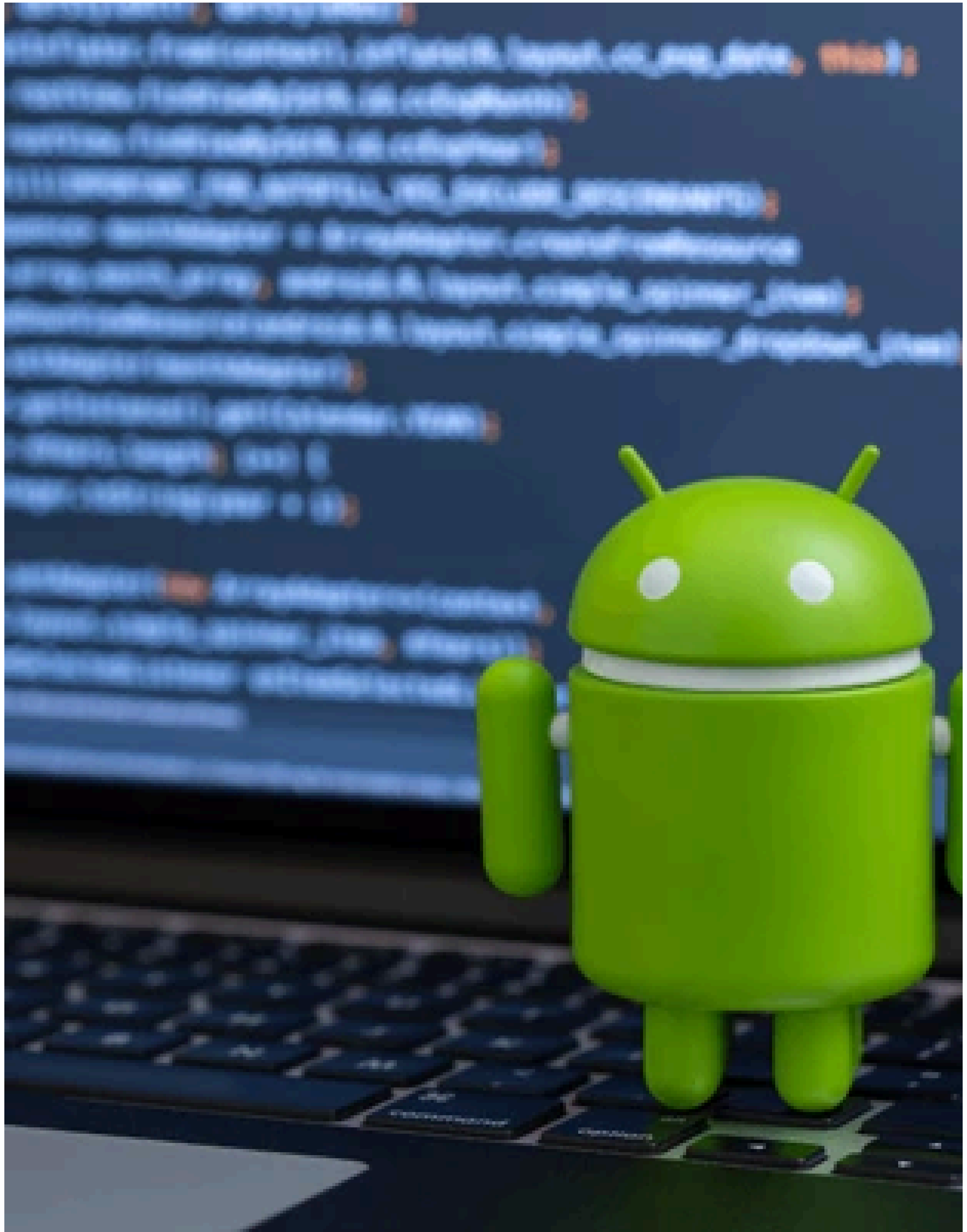
Enable verbose vendor logging ●
Include additional device-specific vendor logs in bug reports, which may contain private information, use more battery and/or use more storage.

Debugging in Android

Debugging - process of **isolating** and **removing** defects in software code.

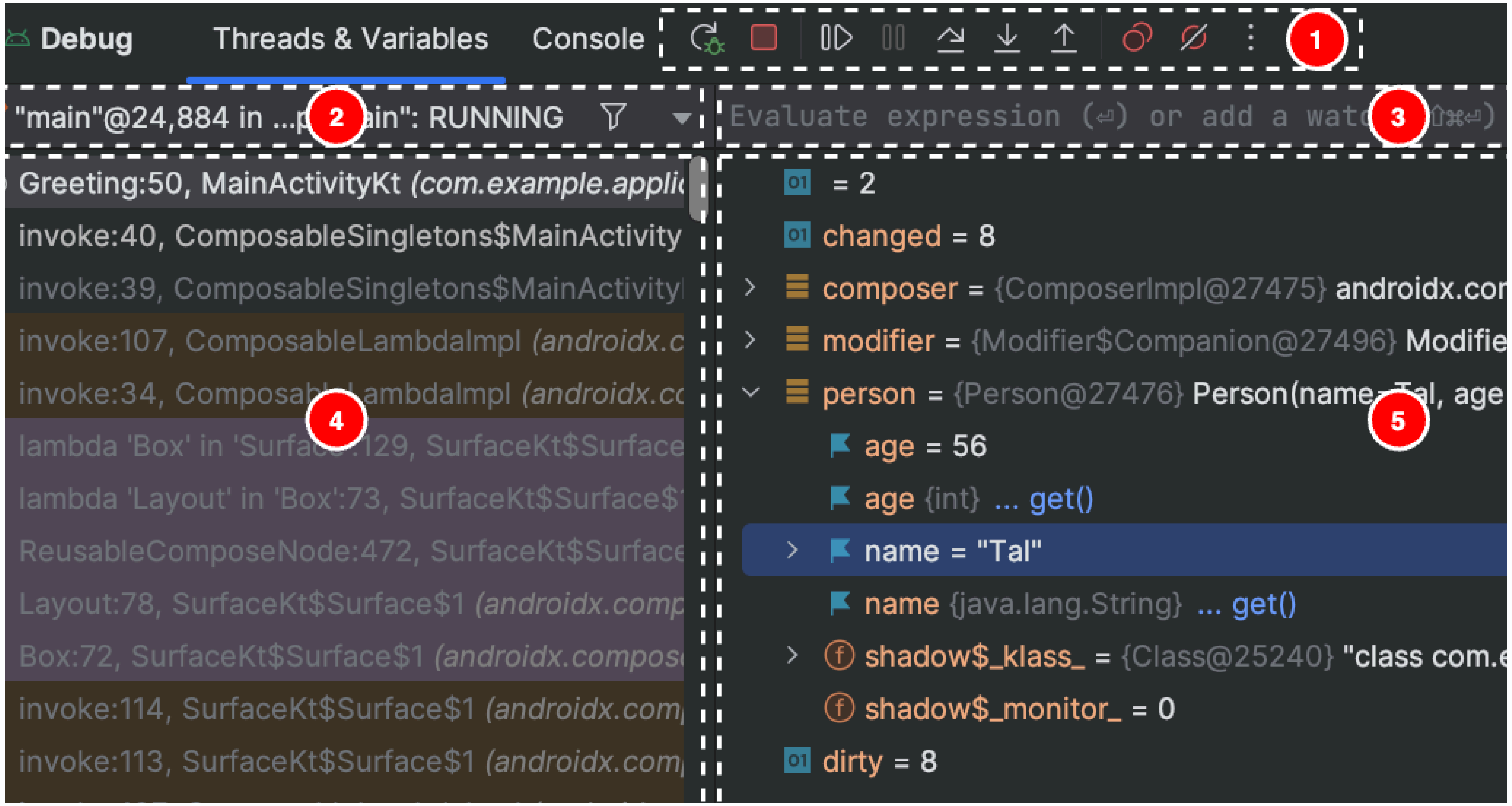
Process of Debugging in Android development.

- Select a device to debug your app on.
- Set **breakpoints** in your coding language e.g Kotlin.
- Examine **variables** and **evaluate** expressions at runtime.



KAMI LIMU

Debugging Window



```
"main"@24,884 in ...p    ain": RUNNING
Evaluate expression (⏎) or add a watc    ⇧⌘⏎)
```

```
Greeting:50, MainActivityKt (com.example.applic
invoke:40, ComposableSingletons$MainActivity
invoke:39, ComposableSingletons$MainActivity
invoke:107, ComposableLambdaImpl (androidx.c
invoke:34, ComposableLambdaImpl (androidx.co
lambda 'Box' in 'Surface':129, SurfaceKt$Surface
lambda 'Layout' in 'Box':73, SurfaceKt$Surface$
ReusableComposeNode:472, SurfaceKt$Surface
Layout:78, SurfaceKt$Surface$1 (androidx.comp
Box:72, SurfaceKt$Surface$1 (androidx.compos
invoke:114, SurfaceKt$Surface$1 (androidx.com
invoke:113, SurfaceKt$Surface$1 (androidx.com
```

```
   = 2
   changed = 8
>  composer = {ComposerImpl@27475} androidx.cor
>  modifier = {Modifier$Companion@27496} Modifie
v  person = {Person@27476} Person(name=Tal, age
      age = 56
      age {int} ... get()
   >  name = "Tal"
      name {java.lang.String} ... get()
   >  shadow$_klass_ = {Class@25240} "class com.e
      shadow$_monitor_ = 0
   dirty = 8
```

1. Execution and navigation toolbar. Work with breakpoints
2. Thread selector
3. Evaluation and watch expression entry. Inspect variables.
4. Stack display
5. Variables pane. Inspect variables.

Thank you.
Any Question or Feedback?