

Updated on 11/10/2024

Sign up

# Android Studio training with Android 12

5 days (35 hours)

## Presentation

Android Studio is a development environment for professionally developing native Android mobile applications. It is originally based on IntelliJ IDEA and uses Gradle, a production engine coded in Java. This IDE/EDI is compatible with Windows, macOS and Linux operating systems. Since the release of Android Studio, the Project Marble initiative has spanned several versions, focusing on improving three main areas of the IDE: system stability, improved and more uniform functionality and, of course, bug fixes.

This course is aimed at all developers wishing to launch a professional-quality mobile project using Android. In this hands-on training course, you'll learn how to exploit the latest features developed by Google for tablets, smartphones and other compatible embedded systems. Create Android applications using the Android Studio tool suite, an integrated development environment (IDE) for Android developers using Java APIs. Using best development practices, we'll create a native mobile application, from scratch right through to production deployment and publication in the Play Store marketplace.

With this Android training course, you'll learn the latest and most productive tools in the Android ecosystem, ensuring rapid mobile application development with minimal effort on your part. We'll cover the main Android topics such as notifications and toasts, intents & broadcast receivers, and related services.

Then you'll learn how to publish your applications and sell them online in the Google Play Store.

Thanks to our expert's feedback, you'll learn about the main tools, best practices and essential resources you need to continue becoming an expert!

As with all our training courses, this one will introduce you to the latest version of the Android SDK (at the time of writing: [Android 12 API level 31](#)).

## Objectives

- Install and configure your development environment
- Create modern mobile graphic interfaces natively
- Discover the new features of the Android Studio IDE
- Harnessing the power of Android 12's Level 31 API
- Deploy your application on the Google Play Store marketplace
- Use an offline database and connect your mobile App to Web Services

## Target audience

- Developers, Lead Developers.

## Prerequisites

- Experience and prior knowledge of Java recommended.

## Further information

- Discover our new [training course on the new Kotlin language](#)
- Our [advanced Android Embedded training course](#) or our [Android training course](#) will take the mystery out of your life.
- If you want to switch to hybrid mobile development, we recommend :
  - Google [Flutter](#)
  - [Ionic training](#)
  - Microsoft [Xamarin training](#)

## Our Android Studio with Android 12 training program

### Installing Android Studio

- Installation
- Installing an SDK
- App Designer
- App Preview
- Workspace Structure
- Editor & Tool windows
- The Android API level/NDK
- Virtual devices
- Instant Run
- Importing a project into Android Studio

### Android SDK

- Introduction
- Overview of the different versions
- Android application life cycle
- Graphical interface editor
- The various graphics components
- Component/code interaction
- Using dynamic assets/static resources
- Permissions
- Layouts
- Navigation between different screens
- Listings
- Custom views
- Menu
- Preference

## Choose your language: Language Support

- Kotlin support
  - Benefits of Kotlin
  - Extending Kotlin
- C/C++ support
  - The NDK
- Development kits
- Creating a Things project

## First steps in Android

- Creating and organizing a project
- Introduction to view development
- Styles and resources
- Launch the application on your phone (or emulator)
- Mobile vs Tablet
- Internationalizing an application

## Accessing hardware APIs

- Geolocation
- Accelerometer
- File system
- Camera
- Media
- Bluetooth

## Device Development

- Screen orientation
- Virtual devices
  - Layout and image qualification
  - Scale and platform
  - Alternative emulators
- Android Wear
  - Connecting to a wearable AVD
  - Wearable layouts
  - The WatchViewStub class
  - Shape - aware layouts
- Accessing sensors
  - Sensor emulation
- Device monitoring
  - Project templates
  - Monitoring and profiling

## Resources

- Assets & Resources
- Asset Studio
  - Image Asset Studio
  - Layered icons
  - Launcher icon tools
  - Vector Asset Studio
  - Vector drawables
- Icon animation
  - Other animations
- General drawables
- Palette library

## UI Design & Development

- Lifecycle, ViewModel, LiveData, Room
- The Layout Editor
- Linear & relative layout classes
- The constraint layout
- ConstraintLayout creation
  - Applying constraints
- Graphic Properties tool
- Alignment : Baseline alignment
- Controlling position
- Toolbar
- Multiple screen previewing
  - Hardware profiles
  - Virtual storage
- Material design

- Android styles
  - Material themes
  - The Theme Editor
  - XML fonts
- The design library
  - Coordinator layout
  - Snackbars & Floating Action Buttons
- Collapsing app bars
  - Raw text resources
  - The percent library

## Data access

- JSON data parser
- Backing up data locally / offline
- Synchronization
- Remote data access
  - Detecting and using the network
  - Creating an HTTP connection
  - Strategy for caching remote data

## Data access

- Templates and Plugins
  - Project templates
  - Navigation Drawer template
  - Explorer structure
  - Class inspection plugins
- The Master/Detail Flow template
- Customization: Custom templates
- Third-party templates
- Project samples
- Example of third-party plugins
  - ADB Wi-Fi
  - Codota

## Best practices

- Use of external libraries/components (Dagger, Glide, Rx....)
- Design
- MVP Architecture
- Avoid memory leaks

## Tools

- Genymotion
- Leak canary
- Fabric
- PostMan

## Debugging & testing

- Debug Android code using Studio tools
- Debugging Mobile Web apps, and via the Android API
- Monitor performance
- Logcat filters
- JUnit testing
  - Local unit tests
- Testing a UI
  - Testing views
  - Testing lists and data
  - Recording tests
  - Remote testing
  - Stress testing
- Performance monitoring
  - CPU profiling
  - Memory profiler
  - Network profiler

## Publication

- Deploy and publish on the Google Play Store marketplace
- Gradle build configurations
- Command-line options
- Product flavors
- APK analysis
- Publishing applications
- Sign your App
- Managing keystores
- Google Play app signing
- Managing automation: Automatic signing

## Companies concerned

This training course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

## Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as enrolment is finalized, the learner receives a self-assessment questionnaire enabling us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and objectives.

This questionnaire also enables us to anticipate any connection or internal security problems (intra-company or virtual classroom) that could be problematic for the follow-up and smooth running of the training session. This questionnaire also enables us to anticipate any connection or internal security difficulties within the company (intra-company or virtual classroom) that could be problematic for the follow-up and smooth running of the training session.

## Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

## Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

## Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

## Sanction

A certificate will be issued to each trainee who completes the course.