

# Implementing Android APIs Using Kotlin

---



**Sriyank Siddhartha**

AUTHOR

[linkedin.com/in/sriyank](https://linkedin.com/in/sriyank)

[sriyank.siddhartha@gmail.com](mailto:sriyank.siddhartha@gmail.com)



# FAQs When Switching to Kotlin



Can we use existing Android APIs written in Java?



Can we use third party Java libraries in Kotlin project?



Yes, you can. Kotlin supports Java **Interoperability**.

Existing Java code can be called from Kotlin and vice-versa.



# Interoperability



# Initial Project Setup

---



# Demo



## Use existing Android APIs

- SharedPreferences
- NavigationView

## Use third party libraries

- Retrofit
- Realm



# Kotlin Standards

**Naming Convention**

**Null Safety**



# Null Safety

lateinit

?.

Safe call operator

?:

Elvis

lazy

?.let { ... }

Safe call with let

!!

not-null assertion



# What to Use?

## lateinit vs. lazy

### lateinit

- If you need mutable variable [ var ]
- If the value is being set by some other variable

**Note:** If you try to access lateinit variable without initializing it then it throws `UninitializedPropertyAccessException`

### lazy

- If you need immutable variable [ val ] which needs to be initialized once and the instance is shared among all





# Summary



## Interoperability

- Java and Kotlin file in same project
- Any existing Android APIs
- Any external libraries

## Null safety

- lateinit
- lazy
- Various operators
  - Safe calls [ ?. ]
  - Elvis [ ?: ]
  - Safe call with let [ ?.let{ } ]
  - Not null assertion [ !! ]



# References



**SharedPreferences**

**Realm Database**

**Retrofit**

**Kotlin Fundamentals and Basics**

