

Exploring First Android App Using Kotlin



Sriyank Siddhartha

AUTHOR

linkedin.com/in/sriyank

sriyank.siddhartha@gmail.com



Overview



Objective

- Get started with Java to Kotlin code conversion

Concepts to implement

- Kotlin Android Extension plugin
 - View binding
 - View caching
- Extension property
- Synthetic property
- Companion object
- Kotlin reflection



Ways to Convert Java to Kotlin

Whole Project

Risky approach

Not recommended

One File at a Time

Safe

Recommended



Initial Project Setup



View Binding



Every developer knows well about `findViewById()`



It results in `nasty` and `redundant` code



Several libraries offers solution to this problem but still we need to write some code such as annotating `Views` to expose them



`Kotlin Android Extension` plugin offers a seamless solution for this. It allows us to get rid of `findViewById()`



Kotlin Android Extentions Plugin: View Binding



Within module's gradle file enable the plugin

apply plugin: 'kotlin-android-extensions'



In your activity, import synthetic properties for a layout

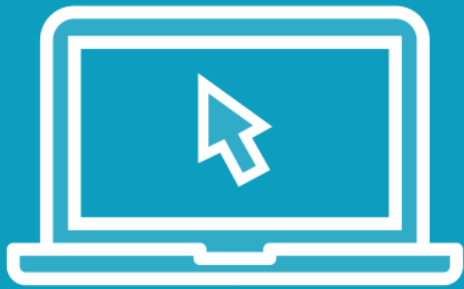
import kotlinx.android.synthetic.main.<layout>.*



Start using a **View** in your activity with the help of its **id**



Demo



Kotlin Android Extensions plugin

Rebuild MainActivity.kt

- Validation layer



Kotlin Android Extension: Behind The Scenes



How the plugin works?

- View caching
 - Calls `findViewById` once for each View
 - Stores the reference to View in cache

Why Views are cached?

- Calling `findViewById` multiple times can be slow
- So the plugin adds a hidden cache function for each of `Activity`, `Fragment` or `View`



Miscellaneous Concepts



Kotlin Reflection

- The most basic reflection: Class References

→ `MyClass :: class`

- Kotlin reference to class of type `KClass<T>`

→ `MyClass :: class.java`

- Java reference to class of type `Class<T>`

Companion object

- Behaves static as in Java



Summary



Successfully converted Java file to Kotlin

Kotlin Android Extension plugin

- View binding
- View caching
- Extension property
- Synthetic property

Implementation of Kotlin References

- Kotlin Reflection

Companion object

- One companion object per Kotlin class

