

Top 25 Mobile Testing Interview Questions (with Kotlin Code Samples)

Author: Lamhot Siagian [LinkedIn](#)

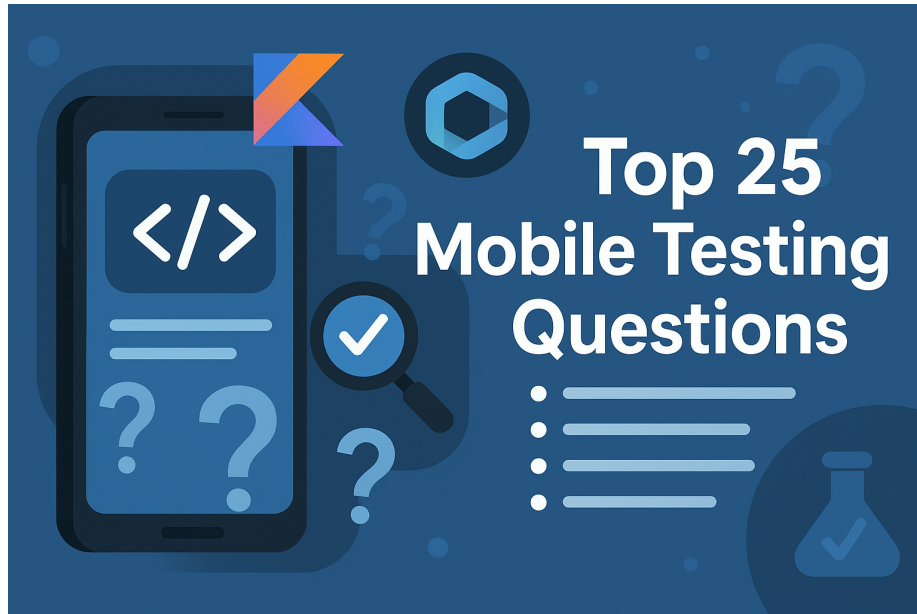


Figure 1: Top 25 Mobile Testing Interview Questions

Below are 25 common mobile-testing questions, each illustrated with a concise Kotlin snippet and a brief explanation.

1. What is Espresso and how do you write a simple Espresso test in Kotlin?

Answer: Espresso is a UI-testing framework from Google for writing concise, reliable Android UI tests. “`kotlin @RunWith(AndroidJUnit4::class) class LoginScreenTest { @get:Rule val activity = ActivityScenarioRule(LoginActivity::class.java)`

```
@Test fun successfulLogin_showsHome() {  
    onView(withId(R.id.username)).perform(typeText("user"))  
    onView(withId(R.id.password)).perform(typeText("pass"), closeSoftKeyboard())  
    onView(withId(R.id.loginButton)).perform(click())  
    onView(withText("Welcome")).check(matches(isDisplayed()))  
}  
} ““
```

We use `onView()` to find views, `perform()` to act, and `check()` to assert.

2. How do you wait for asynchronous operations in Espresso?

Answer: Use an `IdlingResource` to let Espresso know when your app is idle.

```
class SimpleIdlingResource : IdlingResource {
    @Volatile private var callback: IdlingResource.ResourceCallback? = null
    private var isIdle = false
    override fun getName() = this::class.java.name
    override fun isIdleNow() = isIdle.also { if (it) callback?.onTransitionToIdle() }
    override fun registerIdleTransitionCallback(cb: IdlingResource.ResourceCallback) {
        callback = cb
    }
    fun setIdleState(idle: Boolean) { isIdle = idle }
}
```

3. How do you scroll a RecyclerView to a specific item?

```
onView(withId(R.id.recyclerView))
    .perform(
        RecyclerViewActions.scrollTo<RecyclerView.ViewHolder>(
            hasDescendant(withText("Item 42"))
        )
    )
```

4. Show a Compose UI test example.

```
@get:Rule val composeTestRule = createAndroidComposeRule<MainActivity>()

@Test fun counterIncrements() {
    composeTestRule.onNodeWithTag("IncrementBtn").performClick()
    composeTestRule.onNodeWithTag("CounterText")
        .assertTextEquals("Count: 1")
}
```

5. How do you test a Room database in memory?

```
@Before fun setup() {
    val context = ApplicationProvider.getApplicationContext<Context>()
    db = Room.inMemoryDatabaseBuilder(context, AppDatabase::class.java).build()
    dao = db.noteDao()
}
```

```
@After fun tearDown() = db.close()
```

6. How do you mock a network call using Retrofit and Mockito?

```
@Mock lateinit var service: ApiService
@Before fun init() = MockitoAnnotations.openMocks(this)

@Test fun fetchNotes_returnsList() = runBlocking {
    `when`(service.getNotes()).thenReturn(listOf(Note(1, "A", "B")))
    val result = service.getNotes()
    assertEquals(1, result.size)
}
```

7. How do you test a ViewModel in isolation?

```
@get:Rule val coroutineRule = MainCoroutineRule()

@Test fun loadNotes_updatesState() = coroutineRule.runBlockingTest {
    val fakeRepo = FakeNotesRepository()
    val vm = NotesViewModel(fakeRepo)
    vm.loadNotes()
    assertTrue(vm.state.value.notes.isNotEmpty())
}
```

8. How do you test an Activity lifecycle?

```
@Test fun rotateScreen_recreatesActivity() {
    val scenario = ActivityScenario.launch(MainActivity::class.java)
    scenario.onActivity { it.requestedOrientation = ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE }
    scenario.recreate()
    // Assertions after rotation...
}
```

9. How to take a screenshot during a test?

```
val device = UiDevice.getInstance(InstrumentationRegistry.getInstrumentation())
device.takeScreenshot(File("/sdcard/test_screenshot.png"))
```

10. How do you test deep links?

```
@Test fun deepLink_toDetails() {
    val intent = Intent(Intent.ACTION_VIEW).setData("myapp://note/5".toUri())
    ActivityScenario.launch<MainActivity>(intent)
    onView(withText("Note ID: 5")).check(matches(isDisplayed()))
}
```

11. How do you test Android WorkManager tasks?

```
@Test fun workEnqueues_success() {  
    val request = OneTimeWorkRequestBuilder<SyncWorker>().build()  
    WorkManager.getInstance(context).enqueue(request).result.get()  
    val statuses = WorkManager.getInstance(context).getWorkInfosByTag("sync").get()  
    assertEquals(WorkInfo.State.ENQUEUED, statuses.first().state)  
}
```

12. How to test runtime permissions?

```
@Test fun cameraPermission_granted() {  
    grantPermission(Manifest.permission.CAMERA)  
    onView(withId(R.id.openCamera)).perform(click())  
    onView(withText("Camera opened")).check(matches(isDisplayed()))  
}
```

13. How do you test location updates?

```
val fusedLocationClient = mock<FusedLocationProviderClient>()  
whenever(fusedLocationClient.lastLocation).thenReturn(Tasks.forResult(location))
```

14. How to test notifications?

```
val manager = NotificationManagerCompat.from(context)  
val notifications = manager.activeNotifications  
assertTrue(notifications.any { it.id == MY_NOTIFICATION_ID })
```

15. How do you test dynamic feature modules?

```
val splitInstallManager = SplitInstallManagerFactory.create(context)  
assertTrue(splitInstallManager.installedModules.contains("feature_notes"))
```

16. How to test BLE connections?

```
val gatt = mock<BluetoothGatt>()  
`when`(gatt.connect()).thenReturn(true)
```

17. How to test multi-language support?

```
@Test fun spanishLocale_showsHola() {  
    setLocale("es")  
    onView(withId(R.id.greeting)).check(matches(withText(";Hola!")))  
}
```

18. How to test dark mode?

```
@Test fun darkMode_switchesTheme() {
    composeTestRule.setContent {
        MaterialTheme(colors = darkColors()) { MyApp() }
    }
    // UI assertions in dark theme...
}
```

19. How to mock a ViewModel in Compose previews?

```
@Preview @Composable
fun PreviewNotes() {
    NoteAppContent(vm = FakeViewModel())
}
```

20. How to test accessibility labels?

```
onView(withId(R.id.saveButton))
    .check(matches(withContentDescription("Save note")))
```

21. How to test rotation with Compose?

```
composeTestRule.activityRule.scenario.onActivity {
    it.requestedOrientation = ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE
}
```

22. How to stub network failures?

```
`when`(service.getNotes()).thenReturn(IOException("Network error"))
```

23. How do you test database migrations?

```
val helper = MigrationTestHelper(
    InstrumentationRegistry.getInstrumentation(),
    AppDatabase::class.java.canonicalName
)
helper.createDatabase("test-db", 1).apply { close() }
helper.runMigrationsAndValidate("test-db", 2, true, MIGRATION_1_2)
```

24. How to integrate tests into a CI pipeline?

```
# in your CI config
./gradlew connectedAndroidTest --stacktrace
```

25. How do you retry flaky tests automatically?

```
class RetryRule : TestRule {  
    override fun apply(base: Statement, desc: Description) = object : Statement() {  
        override fun evaluate() {  
            var caught: Throwable? = null  
            repeat(3) {  
                try { base.evaluate(); return } catch (e: Throwable) { caught = e }  
            }  
            throw caught!!  
        }  
    }  
}
```

Each snippet shows a focused strategy for testing a specific area of Android mobile apps with Kotlin. Happy interviewing (and coding)!