Top 50 Android Interview Questions & Answers – SDE I to SDE III

Last Updated : 19 Jun, 2024

In this COVID-19 pandemic, the world is moving from the "work from office" culture to the "work from home" culture. Also, people across the world are now depending on apps whether it's for daily tasks like shopping, commuting, paying bills, or it's getting information and staying connected with one another. Because of this reason, companies are not shying away from extending increased Android Developer salaries in India. According to PayScale, **the average Android Software developer salary in India is ₹399,594 / year.**

Additionally, the better the experience, the higher the salary. Similarly, **the average experienced Android developer's salary in India is ₹1,316,973**. These are the numbers in India. If we talk about the Android Developer Salary in **the US and the UK** then the numbers are **$113,900 and £35,554** respectively. And one more interesting thing is that more than 6000 apps are released on Google Play Store every day!!

But is it easy to build your career in "Android Software Development"? What are the processes or steps to get into the Android Development

World? What are the questions asked in the Interview Round? So let's discuss the process or steps to get into the Android Development World in brief.

- **Step 1: Resume Shortlisting**
- **Step 2: Take-Home Assignment**
    - They will provide you with an assignment, and you have to complete it within a stipulated time period
- **Step 3: First Round (Discussion about the Take-Home Assignment + Basic Concepts)**
    - Discussion about the Take-Home Assignment means why you have used this "x" thing and why not this "y",
    - How can it be better,
    - Basics of Kotlin/Java & Android, etc.
- **Step 4: Second Round (In-Depth Concept in Kotlin/Java & Android + Programming)**
    - In-depth means Kotlin Coroutine,
    - Memory Leaks,
    - If your Activity is opening slowly how to investigate that,
    - Data Structure and Algorithm, etc.
- **Step 5: Third Round (Depend on Company to Company)**
    - It may be with Team Lead,
    - HR Round, etc.

In general, the company hires for the following 3 roles:

Roles	Experience Required	Questions to be Expected
SDE – I (Android)	0 – 2 Years of Experience or Fresher	Kotlin/Java Basics, Android Basic Concepts (Activity, Service, Fragment, etc)
SDE – II (Android)	2 – 5 Years of Experience	Architecture (MVVM, MVP), How Testing is Done, etc.
SDE – III (Android)	4 – 8 Years of Experience	Dependency Injection (Dagger, Kotlin Coroutine, Memory Leaks, etc)

From tech giants like **Google, Amazon, Facebook**, etc to the rising startups like **Dream11, Zomato, Paytm, CRED**, etc are hiring lots of Android Developers for their companies. So we have prepared a list of **Top 50 Android Interview Questions along with their answers** and the questions are designed from fresher to advanced level serially.

1. What is Android and the latest version of Android?

Android is an operating system that is built basically for Mobile phones. It is based on the Linux Kernel and other open-source software and is developed by Google. It is used for touchscreen mobile devices such as smartphones and tablets. But nowadays these are used in Android Auto cars, TV, watches, cameras, etc. It has been one of the best-selling OS for smartphones. Android OS was developed by Android Inc.

The latest version of Android is **Android 14 (API level 34)** and the initial stable release date is October **4, 2023**.

To read more, refer* to *the article: Introduction to Android

2. What’s Activity in Android?

Activity class is one of the very important parts of the Android component. Any app, no matter how small it is (in terms of code and scalability), has at least one Activity class. Unlike most programming languages, in which the **main()** method is the entry point for that program or application to start its execution, the android operating system initiates the code in an Activity instance by invoking specific callback methods that correspond to specific stages of its lifecycle. So it can be said that An activity is the entry point for interacting with the user.

To read more, refer* to *the article: Introduction to Activities in Android

3. Why is XML used for frontend development in Android?

XML stands for **Extensible Markup Language**. XML is a markup language much like HTML used to describe data. XML itself is well readable both by humans and machines. Also, it is scalable and simple to develop. In Android, we use XML for designing our layouts because XML is a lightweight language, so it doesn’t make our layout heavy.

4. What are the components of the Android Application?

There are some necessary building blocks that an Android application consists of. These loosely coupled components are bound by the application manifest file which contains the description of each component and how they interact. The four main components of android applications are:

- Activities
- Services
- Content Providers
- Broadcast Receiver
- Intents

To read more, refer **to** ***the article:*** Components of an Android Application

5. What is the Dalvik Virtual Machine?

DVM is a virtual machine to execute Android applications. The Java bytecode**(.class file)** generated by the javac compiler is converted into **Dalvik bytecode** to make the application source files executable on the DVM. Since Android devices have a definite processing capacity, memory, and battery life, the DVM design principle aims to optimize itself so that it can load fastly and run smoothly even on low memory/powered devices. This virtual machine is very efficient in running multiple instances on the same device.

To read more, refer **to** ***the article:*** DVM

6. What are the differences between Dalvik and ART?

Dalvik Virtual Machine	Android Run Time
Faster Booting time	Rebooting is significantly longer
Cache builds up overtime	The cache is built during the first boot
Occupies less space due to JIT	Consumes a lot of storage space internally due to AOT
Works best for small storage devices	Works best for Large storage devices
Stable and tested virtual machine	Experimental and new – not much app support comparatively
Longer app loading time	Extremely Faster and smoother Faster and app loading time and

Dalvik Virtual Machine	Android Run Time
	lower processor usage
Uses JIT compiler (JIT: Just-In-Time) Thereby resulting in lower storage space consumption	Uses AOT compiler(Ahead-Of-Time) thereby compiling apps when installed
Application lagging due to garbage collector pauses and JIT	Reduced application lagging and better user experience
App installation time is comparatively lower as the compilation is performed later	App installation time is longer as compilation is done during installation
DVM converts bytecode every time you launch a specific app.	ART converts it just once at the time of app installation. That makes CPU execution easier. Improved battery life due to faster execution.

To read more, refer **to** ***the article:*** Difference Between Dalvik and ART in Android

7. How does an Android App Work?

Developing an android application involves several processes that happen sequentially. After writing the source code files, when developers click the Run button on the Android studio, plenty of operations and processes start at the backend.

- **Building the APK File**
  - Code Compilation
  - Conversion into Dalvik bytecodes
  - Generating .apk file

    - App Distribution
- **Deploy the Application**
    - Establish the ADB Server
    - Transfer .apk file to the Device
- **Run the Application**
    - App launch request
    - Conversion of the .dex code to native OAT format

To read more, refer to the article: [How Does Android App Work?]

8. What is Toast in Android?

A **Toast** is a short alert message shown on the Android screen for a short interval of time. Android **Toast** is a short popup notification that is used to display information when we perform any operation in our app. It disappears automatically. If the user wants a permanently visible message, then a **notification** can be used.

To read more, refer to the article: Toast

9. What’s Service in Android?

Services in Android are a special component that facilitates an application to run in the background in order to perform long-running operation tasks. The prime aim of a service is to ensure that the application remains active in the background so that the user can operate multiple applications at the same time. A user interface is not desirable for android services as it is designed to operate long-running processes without any user intervention. A service can run continuously in the background even if the application is closed or the user switches to another application.

To read more, refer to the article: Services in Android

10. What’s Content Provider in Android?

In Android, Content Providers are a very important component that serves the purpose of a relational database to store the data of applications. The role of the content provider in the android system is like a central repository in which data of the applications are stored, and it facilitates other applications to securely access and modify that data based on the user requirements. The Android system allows the content provider to store the application data in several ways. Users can manage to store the application data like images, audio, videos, and personal contact information by storing them in SQLite Database, in files, or even on a network.

To read more, refer* to *the article: Content Providers in Android

11. What's Broadcast Receiver in android?

Broadcast in android is the system-wide event that can occur when the device starts, when a message is received on the device, when incoming calls are received, or when a device goes to airplane mode, etc. Broadcast Receivers are used to respond to these system-wide events. Broadcast Receivers allow us to register for the system and application events, and when that event happens, then the registered receivers get notified. There are mainly two types of Broadcast Receivers:

- **Static Broadcast Receivers:** These types of Receivers are declared in the manifest file and work even if the app is closed.
- **Dynamic Broadcast Receivers:** These types of receivers work only if the app is active or minimized.

To read more, refer* to *the article: Broadcast Receiver in Android

12. What's Gradle and write down its usage in Android?

Gradle is a build system (open source) that is used to automate building, testing, deployment, etc. "Build.gradle" are scripts where one can automate the tasks. For example, the simple task to copy some files from one directory to another can be performed by Gradle build script before the actual build process happens.

Usage: Every Android project needs a Gradle for generating an apk from the *.java* and *.xml* files in the project. Simply put, a Gradle takes all the source files (java and XML) and applies appropriate tools, e.g., converts the java files into dex files and compresses all of them into a single file known as apk that is actually used.

To read more, refer* to *the articles: build.gradle file, Introduction to Gradle

13. What's Fragment in android?

In Android, the fragment is the part of Activity that represents a portion of the User Interface(UI) on the screen. It is the modular section of the android activity that is very helpful in creating UI designs that are flexible in nature and auto-adjustable based on the device screen size. There are mainly 3 types of fragments:

- Single Fragment
- List Fragment
- Fragment Transaction

To read more, refer* to *the article: Introduction to Fragments

14. What's RecyclerView in Android & How it works?

RecyclerView is a ViewGroup added to the Android Studio as a successor of the GridView and ListView. It is an improvement on both of them. It has been created to make possible construction of any lists with **XML** layouts as an item that can be customized vastly while **improving the efficiency of ListViews and GridViews**. This improvement is achieved by recycling the views which are out of the visibility of the user. For example, if a user scrolled down to a position where items 4 and 5 are visible; items 1, 2, and 3 would be cleared from the memory to reduce memory consumption.

To read more, refer to the article: RecyclerView in Android with Example

15. What's the Difference Between Intent and Intent filters?

An Intent is an object passed to **Context.startActivity(), Context.startService() or Activity.startActivityForResult()** etc. to launch an activity or get an existing activity to do something new. On the other hand, an Intent filter describes the capability of the component(like activities, services, and broadcast receivers).

To read more, refer to the article: Difference Between Implicit Intent and Explicit Intent in Android

16. What is the AndroidManifest.xml?

Every project in Android includes a manifest file, which is AndroidManifest.xml, stored in the root directory of its project hierarchy. The manifest file is an important part of our app because it defines the structure and metadata of our application, its components, and its requirements. This file includes nodes for each of the Activities, Services, Content Providers, and Broadcast Receivers that make the application and using Intent Filters and Permissions determines how they coordinate with each other and other applications. The manifest file also specifies the application metadata, which includes its icon, version number, themes, etc., and additional top-level nodes can specify any required permissions, unit tests, and define hardware, screen, or platform requirements.

To read more, refer to the article: The Application Manifest File

17. What's Android SDK and its Components?

Android SDK stands for **Android Software Development Kit** which is developed by Google for Android Platform. With the help of Android SDK, we can create Android Apps easily. Android SDK is a collection of libraries and Software Development tools that are essential for

Developing Android Applications. Whenever Google releases a new version or update of Android Software, a corresponding SDK also releases with it. In the updated or new version of SDK, some more features are included which are not present in the previous version. Android SDK consists of some tools which are very essential for the development of Android Application. These tools provide a smooth flow of the development process from developing and debugging. Android SDK is compatible with all operating systems such as Windows, Linux, macOS, etc.

Components:

- Android SDK Build tool.
- Android Emulator.
- Android SDK Platform-tools.
- Android SDK Tools.

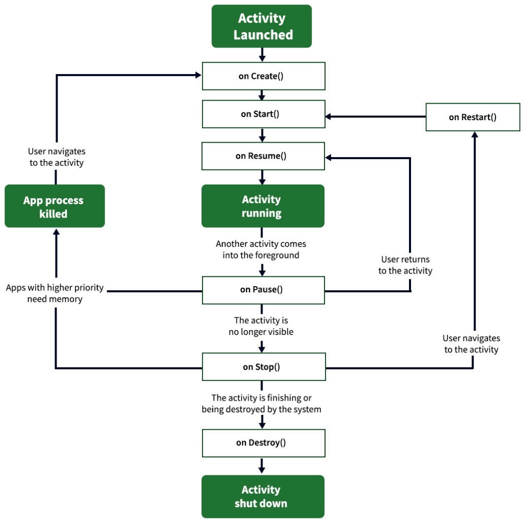
To read more, refer* to *the article: Android SDK and its Components

18. Disadvantages of Android

- Fragmentation provides a very intuitive approach for user experience, but it has some drawbacks, where the development team needs time to adjust with the various screen sizes of mobile smartphones that are now available in the market and invoke the particular features in the application.
- The Android devices might vary broadly. So the testing of the application becomes more difficult.
- As the development and testing consume more time, the cost of the application may increase, depending on the application's complexity and features.

19. Explain the Activity Lifecycle in brief.

These are the different stages of the Activity Lifecycle:


Activity Launched
on Create()
on Start()
on Restart()
User navigates to the activity
on Resume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
on Pause()
The activity is no longer visible
User navigates to the activity
on Stop()
The activity is finishing or being destroyed by the system
on Destroy()
Activity shut down

Activity Lifecycle in Android

- **onCreate()**: It is called when the activity is first created. This is where all the static work is done like creating views, binding data to lists, etc.
- **onStart()**: It is invoked when the activity is visible to the user. It is followed by onResume() if the activity is invoked from the background.
- **onRestart()**: It is invoked after the activity has been stopped and prior to its starting stage and thus is always followed by onStart() when any activity is revived from background to on the screen.
- **onResume()**: It is invoked when the activity starts interacting with the user. At this point, the activity is at the top of the activity stack, with a user interacting with it.
- **onPause()**: It is invoked when an activity is going into the background but has not yet been killed. It is a counterpart to onResume()
- **onStop()**: It is invoked when the activity is not visible to the user. It is followed by onRestart() when the activity is revoked from the background, followed by onDestroy() when the activity is closed or finished, and nothing when the activity remains on the background only.
- **onDestroy()**: The final call received before the activity is destroyed. This can happen either because the activity is finished (when finish() is invoked) or because the system is temporarily destroying this instance of the activity to save space.

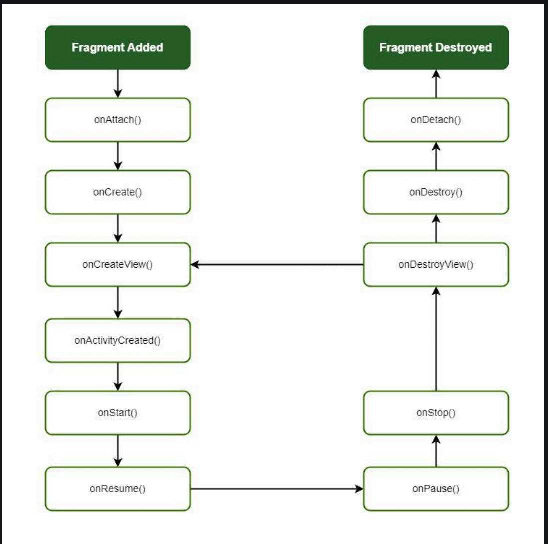
To read more, refer to the article: Activity Lifecycle in Android with Demo App

20. Why do we need to call setContentView() in onCreate() of Activity class?

The reason for doingso is that the activity life cycle onCreate() method is called only once. And this is the big reason we need to call the setContentView() in onCreate(). And it will be inefficient to call this function in onResume(), onStart(), and somewhere else because those methods are called more than once.

21. Explain the Fragment Lifecycle in Brief.

These are the different stages of the Activity Lifecycle:



Android Fragment Lifecycle

Methods	Description
onAttach()	The very first method to be called when the fragment has been associated with the activity. This method executes only once during the lifetime of a fragment.
onCreate()	This method initializes the fragment by adding all the required attributes and components.

Methods	Description
onCreateView()	System calls this method to create the user interface of the fragment. The root of the fragment's layout is returned as the View component by this method to draw the UI.
onActivityCreated()	It indicates that the activity has been created in which the fragment exists. View hierarchy of the fragment also instantiated before this function call.
onStart()	The system invokes this method to make the fragment visible on the user's device.
onResume()	This method is called to make the visible fragment interactive.
onPause()	It indicates that the user is leaving the fragment. System calls this method to commit the changes made to the fragment.
onStop()	Method to terminate the functioning and visibility of fragments from the user's screen.
onDestroyView()	System calls this method to clean up all kinds of resources as well as view hierarchy associated with the fragment.
onDestroy()	It is called to perform the final clean-up of the fragment's state and its lifecycle.
onDetach()	The system executes this method to disassociate the fragment from its host activity.

To read more, refer* to *the article: Fragment Lifecycle in Android

22. What is the difference between a Fragment and an Activity?

Activity	Fragment
Activity is an application component that gives a user interface where the user can interact.	The fragment is only part of an activity, it basically contributes its UI to that activity.
Activity is not dependent on fragment	Fragment is dependent on activity. It can't exist independently.
we need to mention all activity it in the manifest.xml file	Fragment is not required to mention in the manifest file
We can't create multi-screen UI without using fragment in an activity,	After using multiple fragments in a single activity, we can create a multi-screen UI.
Activity can exist without a Fragment	Fragment cannot be used without an Activity.
Creating a project using only Activity then it's difficult to manage	While Using fragments in the project, the project structure will be good and we can handle it easily.
Lifecycle methods are hosted by the OS. The activity has its own life cycle.	Lifecycle methods in fragments are hosted by hosting the activity.
Activity is not light weight.	The fragment is the lite weight.

To read more, refer **to** ***the article:*** Difference Between a Fragment and an Activity in Android

23. What's Context in Android?

The context in Android can be understood as something which gives us the context of the current state of our application. We can break the context and its use into three major points:

- It allows us to access resources.
- It allows us to interact with other Android components by sending messages.
- It gives you information about your app environment.

There are mainly two types of context available in Android.

1. Application Context and
2. Activity Context

To read more, refer to the article: What is Context in Android?

24. Top Image Loading Libraries in Android.

- Picasso
- Glide
- Fresco
- COIL (Coroutine Image Loader)
- UIL (Universal Image Loader)

To read more, refer to the article: Top Image Loading Libraries in Android

25. What is View in Android?

The view is a class that represents the basic building block for UI components. A View occupies a rectangular area on the screen and is responsible for drawing and event handling. It is a superclass for all the UI components. The most common UI components are:

- TextView
- EditText
- ImageView
- Button

- ProgressBar
- CheckBox, etc.

26. Difference Between View and ViewGroup in Android

View	ViewGroup
View is a simple rectangle box that responds to the user's actions.	ViewGroup is the invisible container. It holds View and ViewGroup
View is the SuperClass of All component like TextView, EditText, ListView, etc	ViewGroup is a collection of Views(TextView, EditText, ListView, etc..), somewhat like a container.
A View object is a component of the user interface (UI) like a button or a text box, and it's also called a widget.	A ViewGroup object is a layout, that is, a container of other ViewGroup objects (layouts) and View objects (widgets)
Examples are EditText, Button, CheckBox, etc.	For example, LinearLayout is the ViewGroup that contains Button(View), and other Layouts also.
View refers to the android.view.View class	ViewGroup refers to the android.view.ViewGroup class
android.view.View which is the base class of all UI classes.	ViewGroup is the base class for Layouts.

To read more, refer* to *the article: Difference Between View and ViewGroup in Android

27. What is Kotlin Coroutine on Android?

The Kotlin team defines coroutines as “**lightweight threads**”. They are sort of tasks that the actual threads can execute. Kotlin coroutines introduce a new style of concurrency that can be used on Android to simplify async code. **The official documentation says that coroutines are lightweight threads. By lightweight, it means that creating coroutines doesn’t allocate new threads. Instead, they use predefined thread pools and smart scheduling for the purpose of which task to execute next and which tasks later**.

To read more, refer* to *the article: Kotlin Coroutines on Android

28. How Garbage Collector Works in Android?

Garbage Collector in Android has no compacting. This means the address of objects in the heap never changed after their creation. So garbage collection can be triggered when an allocation fails when an

- OutOfMemoryError is about to be triggered,
- When the size of the heap hits some soft limit, and
- When a GC was explicitly requested.

29. Describe the architecture of your last app.

When developers work on a real mobile application whose nature is dynamic and will expand its features according to the user’s need, then it is not possible to write core logic in activities or fragments. To structure the project’s code and to give it a modular design(separated code parts), architecture patterns are applied to separate the concerns. The most popular android architectures used by developers are the following:

- MVC (Model — View — Controller)
- MVP (Model — View — Presenter)
- MVVM (Model — View — ViewModel)

So you have to tell the architecture of the last app you have developed during the college time project or in a real industry project.

To read more, refer to *the article:* Android Architecture Patterns

30. MVC vs MVP vs MVVM architecture and which one we should choose?

MVC(MODEL VIEW CONTROLLER)	MVP(MODEL VIEW PRESENTER)	MVVM(MODEL VIEW VIEWMODEL)
One of the oldest software architecture	Developed as the second iteration of software architecture which is advanced from MVC.	Industry-recognized architecture pattern for applications.
UI(**View**) and data-access mechanism(**Model**) are tightly coupled.	It resolves the problem of having a dependent **View** by using **Presenter** as a communication channel between **Model** and **View**.	This architecture pattern is more event-driven as it uses data binding and thus makes easy separation of **core business logic** from the **View**.
Controller and **View** exist with the one-to-many relationship. One Controller can select a different view based upon the required operation.	A one-to-one relationship exists between **Presenter** and **View** as one Presenter class manages one View at a time.	Multiple **View** can be mapped with a single **ViewModel** and thus, the one-to-many relationship exists between View and ViewModel.

Which one should we choose?

- **MVC:** Ideal for small-scale projects only.
- **MVP:** Ideal for simple and complex applications.

- **MVVM:** Ideal for big-scale projects. But not ideal for small-scale projects.

To read more, refer* to *the article: Difference Between MVC, MVP and MVVM Architecture Pattern in Android

31. How to Reduce APK size in android?

- Remove unused sources
- Use of Vector Drawables
- Reuse your code
- Compress PNG and JPEG files
- Use of Lint
- Use images in WebP file format
- Use of proguard
- Use of ShrinkResources
- Limit the usage of external libraries
- Use the Android Size Analyzer tool
- Generate App Bundles instead of APK
- Use of Resconfigs

To read more, refer* to *the article: How to Reduce APK Size in Android?

32. What's the Android jetpack and its Key Benefits?

Jetpack is nothing but a set of software components, libraries, tools, and guidance to help in developing great Android apps. Google launched **Android Jetpack** in 2018. Key Benefits of Android Jetpack

- Forms a recommended way for app architecture through its components
- Eliminate boilerplate code
- Simplify complex task
- Provide backward compatibility as libraries like support are unbundled from Android API and are re-packaged to **androidx.* package**

- Inbuilt productivity feature of the Kotlin Integration

To read more, refer* to *the article: Introduction to Android Jetpack

33. What are the different software components of Android Jetpack?

The software components of Android Jetpack has been divided into 4 categories:

- **Foundation Components:** AppCompat, Android KTX, Test, Multidex
- **Architecture Components:** Room, WorkManager, Lifecycle, ViewModel, Paging, Navigation
- **Behavior Components:** DownloadManager, Permissions, Sharing, Slices
- **UI Components:** Animation & Transition, Auto, Fragment, Palette, Layout

34. What's Jetpack Compose and its Benefits?

Jetpack Compose is a modern UI toolkit recently launched by Google which is used for building native Android UI. It simplifies and accelerates the UI development with less code, Kotlin APIs, and powerful tools.

- Declarative
- Compatible
- Increase development speed
- Concise and Idiomatic Kotlin
- Easy to maintain
- Written in Kotlin

To read more, refer* to *the article: Jetpack Compose in Android

35. What are the Architecture Components of Android?

Architecture Components could be classified as follows:

- Room
- WorkManager
- Lifecycle
- ViewModel
- LiveData
- Navigation
- Paging
- Data Binding

To read more, refer* to *the article: Jetpack Architecture Components in Android

36. How to Improve RecyclerView Scrolling Performance in Android?

- Set a specific width and height to ImageView in RecyclerView items
- Avoid using NestedView
- Use the setHasFixedsize method
- Use the image loading library for loading images
- Do less work in the OnBindViewHolder method
- Use the NotifyItem method for your RecyclerView

To read more, refer* to *the article: How to Improve RecyclerView Scrolling Performance in Android?

37. Volley Library in Android

Volley is an **HTTP library** that makes networking very easy and fast, for Android apps. It was developed by Google and introduced during Google I/O 2013. It was developed because there is an absence in Android SDK, of a networking class capable of working without interfering with the user experience. Although Volley is a part of the Android Open Source Project(AOSP), Google announced in January 2017 that Volley will move to a standalone library. It manages the processing and caching of network requests, and it saves developers valuable time from writing the same network call/cache code again and again.

To read more, refer* to *the article: Volley Library in Android

38. What's Retrofit in Android?

Retrofit is a type-safe REST client built by square for Android and Java which intends to make it simpler to expand RESTful web services. Retrofit uses OkHttp as the system's administration layer and is based on it. Retrofit naturally serializes the JSON reaction utilizing a POJO (PlainOldJavaObject) which must be characterized as cutting edge for the JSON Structure. To serialize JSON we require a converter to change it into Gson first. Retrofit is much simpler than other libraries; we don't have to parse our JSON. It directly returns objects but there is one disadvantage: it doesn't provide support to load images from the server, but we can use Picasso for the same.

To read more, refer* to *the article: Introduction to Retrofit

39. Describe MVVM

Model — View — ViewModel (MVVM) is the industry-recognized software architecture pattern that overcomes all drawbacks of MVP and MVC design patterns. MVVM suggests separating the data presentation logic(Views or UI) from the core business logic part of the application.

The separate code layers of MVVM are:

- **Model:** This layer is responsible for the abstraction of the data sources. Model and ViewModel work together to get and save the data.
- **View:** The purpose of this layer is to inform the ViewModel about the user's action. This layer observes the ViewModel and does not contain any kind of application logic.
- **ViewModel:** It exposes those data streams which are relevant to the View. Moreover, it serves as a link between the Model and the View.

To read more, refer* to *the article: MVVM (Model View ViewModel) Architecture Pattern in Android

40. What are the reasons your android app is legging?

- You are doing too much on the main thread
- Your asset files are huge
- You are using an outdated SDK version
- You are using bad libraries
- The speed of the network
- Chatty conversations
- Your code is inefficient

To read more, refer* to *the article: What are the reasons your android app is legging?

41. What is AIDL?

On Android what happens with the processors is one process can't normally access the memory of another process. So in order to interact they need to decompose their objects into primitives that the OS can understand and gather the objects across that boundary. The code to do that gathering is very complex to write, so Android handles it with AIDL. So generally AIDL is similar to the other IDLs and it allows to define the programming interface that both the client and service agree upon in order to interact with each other using interprocess communication (IPC).

42. What is ANR and How can it be Prevented in Android?

ANR stands for Application Not Responding. An ANR will occur if you're running a process on the UI thread **which takes an extended time**, usually around 5 seconds. During this point, the GUI (Graphical User Interface) will lock up which can end in anything the user presses won't be actioned. After the 5 seconds approx. has occurred, if the thread still hasn't recovered then an ANR dialogue box is shown informing the user that the appliance isn't responding and can give the user the choice to either wait, in the hope that the app will eventually recover, or to force close the app.

Stop doing heavy tasks on the main thread. Instead, use worker threads such as IntentService, AsyncTask Handler, or another Thread simply. **Detecting where ANRs happen is straightforward if it's a permanent block** (deadlock acquiring some locks for instance), but harder if it's just a short-lived delay. First, re-evaluate your code and appearance for vulnerable spots and long-running operations.

To read more, refer to the article: What is ANR and How it Can be Prevented in Android?

43. What is Android NDK and why is it useful?

The NDK (Native Development Kit) is a tool that allows you to program in C/C++ for Android devices. It provides platform libraries one can use to manage native activities and access physical device components, such as sensors and touch input. NDK can be useful for cases in which you need to do one or more of the following:

- Squeeze extra performance out of a device to achieve low latency or run computationally intensive applications, such as games or physics simulations.
- Reuse your own or other developers' C or C++ libraries.

44. Explain the JUnit test in brief.

JUnit is a "**Unit Testing**" framework for Java Applications which is already included by default in android studio. It is an automation framework for Unit as well as UI Testing. It contains annotations such as **@Test, @Before**, **@After**, etc. Here we will be using only **@Test** annotation to keep the article easy to understand.

To read more, refer to the article: Unit Testing in Android using JUnit

45. What's Dagger and When to use Dagger?

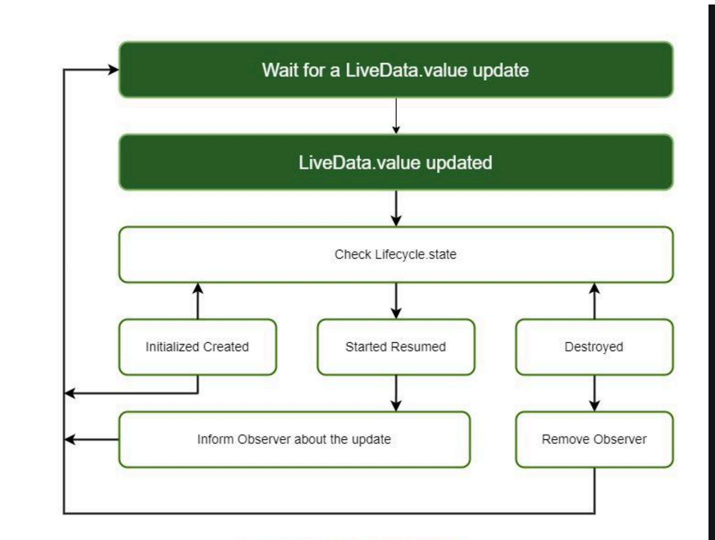
Dagger is a **Dependency Injection** framework that will generate a lot of boilerplate code for you to achieve the goal of **Dependency injection** in Android development. Not that one must have a good understanding of **Dependency Injection** before answering this question.

When to use a dagger?

Project Size	Small	Medium	Large
Tool to use	Manual DI Service Locator Dagger	Dagger	Dagger

46. What’s LiveData in Android Architecture Component and its Advantages?

LiveData component is an observable data holder class i.e, the contained value can be observed. LiveData is a lifecycle-aware component and thus it performs its functions according to the lifecycle state of other application components. Further, if the observer’s lifecycle state is active i.e., either STARTED or RESUMED, only then LiveData updates the app component. LiveData always checks the observer’s state before making any update to ensure that the observer must be active to receive it. If the observer’s lifecycle state is destroyed, LiveData is capable of removing it, and thus it avoids memory leaks. It makes the task of data synchronization easier.



LiveData Component

Advantages of LiveData component:

- UI is updated as per the appropriate change in the data
- It removes the stopped or destroyed activities which reduce the chance of app crash
- No memory leaks as LiveData is a lifecycle-aware component.

To read more, refer* to *the article: [Jetpack Architecture Components in Android]

47. What's Data Binding in Android?

Data Binding library is a support library that provides the feature of binding UI components in an activity/fragment to the data sources of the application. The library carries out this binding task in a declarative format and not in a programmatic way. Below is an example to understand the working of this library accurately:

To find a TextView widget and bind it to the userName property of the ViewModel variable, the **findViewById()** method is called:

```
TextView textView = findViewById(R.id.sample_text);

textView.setText(viewModel.getUserName());
```

After using the Data Binding library, the above code changes by using the assignment expression as follows:

```
<TextView
 android:text=”@{viewmodel.userName}” />
```

Advantages of Data Binding Component:

- Make code simpler and easy to maintain by removing UI frameworks called in the activity.
- Allows classes and methods to observe changes in data
- Allows to make objects and fill which works as collection observables.

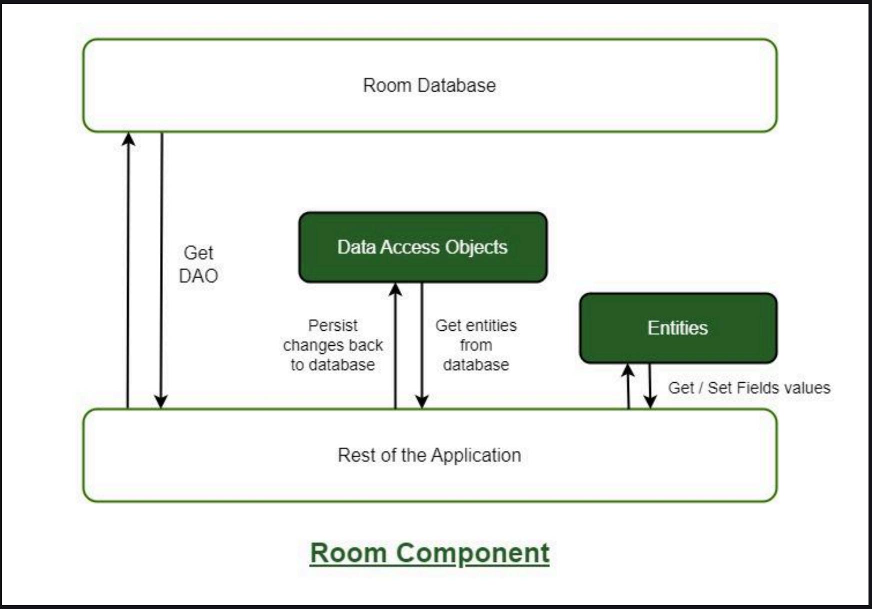
To read more, refer to the article: Jetpack Architecture Components in Android

48. Room in Android Architecture Component.

The requirement of a database in Android is fulfilled by SQLite from the very beginning. However, it comes with some severe drawbacks like not checking the queries at compile-time, it does not save plain-old-Java Objects(commonly referred to as POJOs). Developers also need to write a lot of boilerplate code to make the SQLite database work in the Android OS environment. The Room component comes into the picture as an **SQLite Object Mapping Library** which overcomes all the mentioned challenges. Room converts queries directly into objects, checks errors in queries at the compile-time, and is also capable of persisting the Java POJOs.

Moreover, it produces LiveData results/observables from the given query result. Because of this versatile nature of the Room component, Google officially supports and recommends developers to use it. The Room consists of the following sub-components:

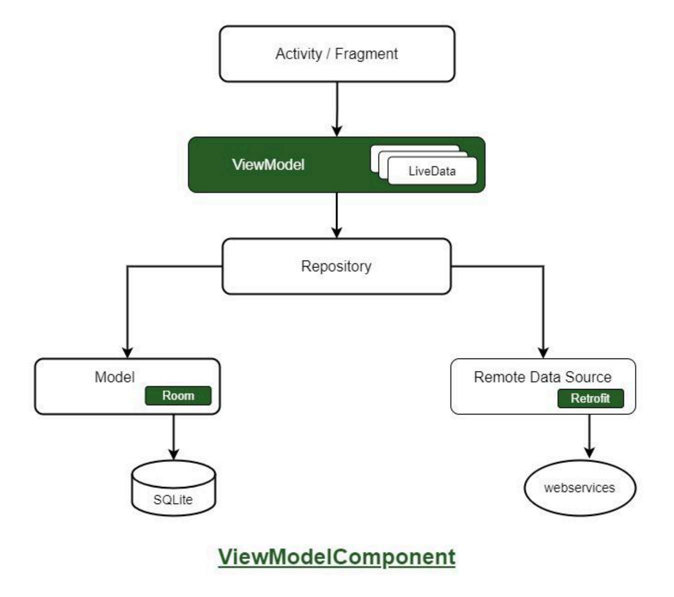
1. **Entity:** It is the annotated class for which the Room creates a table within the database. The field of the class represents columns in the table.
2. **DAO(Data Access Object):** It is responsible for defining the methods to access the database and to perform operations.
3. **Database:** It is an abstract class that extends **RoomDatabase** class and it serves as the main access point to the underlying app's relational data.



To read more, refer **to** ***the article:*** Jetpack Architecture Components in Android

49. ViewModel in Android.

ViewModel is one of the most critical classes of the Android Jetpack Architecture Component that support data for UI components. Its purpose is to hold and manage the UI-related data. Moreover, its main function is to maintain the integrity and allows data to be serviced during configuration changes like screen rotations. Any kind of configuration change in Android devices tends to recreate the whole activity of the application. It means the data will be lost if it has not been saved and restored properly from the activity which was destroyed. To avoid these issues, it is recommended to store all UI data in the ViewModel instead of an activity.



To read more, refer* to *the article: [Jetpack Architecture Components in Android]

50. What's Android KTX?

KTX library is the only one among the foundation components which was introduced for the first time with the release of the Jetpack. Android KTX is a collection of Kotlin extensions that are designed to facilitate developers to remove boilerplate code as well as to write concise code while developing android applications with Kotlin language. Here **KTX** in the name stands for **Kotlin Extensions**. Below is an example of a piece of code without using and after using the Android KTX library:

Code snippet of SQLite without using KTX library:

```
db.beginTransaction()
try {
// insert data
db.setTransactionSuccessful()
}
finally {
db.endTransaction()
}
```

Above code after using KTX library:

```
db.transaction {
  // insert data
}
```

To read more, refer* to *the article: [Foundation Components of Android Jetpack]

A Amiy... 24

Next Article

Top 50 Android Interview Questions & Answers - SDE I to SDE III

Similar Reads

Top 20 Phone Interview Questions &amp; Answers

Phone interviews are an initial step in the hiring process, enabling recruiters to evaluate candidates and ensure they meet the necessary...

15+ min read

15 Top Blockchain Interview Questions &amp; Answers 2024

Blockchain is the backbone technology of digital cryptocurrencies. Bitcoin. The blockchain is a distributed database that stores records of...

9 min read

Top 25 Android Interview Questions and Answers For Experienced

The need for new technologies and their developers is increasing as the world is dynamically transforming digitally. Everyone nowadays relies ...

15+ min read

TCS SDE Sheet: Interview Questions and Answers

What is TCS NQT ? TCS NQT is National Qualifier Test conducted by Tata Consultancy Services, it is a prerequisite for all the tests. The...

8 min read

Apple SDE Sheet: Interview Questions and Answers

Apple is one of the world's favorite tech brands, holding a tight spot as one of the tech Big Four companies. Apple products have found their...

11 min read

View More Articles

Article Tags : interview-questions Android Interview Questions

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

DevOps

Git

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

Accountancy

Business Studies

Economics

Management

HR Management

Finance

Income Tax

Databases

SQL

MYSQL

PostgreSQL

PL/SQL

MongoDB

Preparation Corner

Company-Wise Recruitment Process

Resume Templates

Aptitude Preparation

Puzzles

Company-Wise Preparation

Companies

Colleges

Competitive Exams

JEE Advanced

UGC NET

UPSC

SSC CGL

SBI PO

SBI Clerk

IBPS PO

IBPS Clerk

More Tutorials

Software Development

Software Testing

Product Management

Project Management

Linux

Excel

All Cheat Sheets

Free Online Tools

Typing Test

Image Editor

Code Formatters

Code Converters

Write & Earn

Write an Article

Improve an Article

Pick Topics to Write

Share your Experiences