Version 1.0

2018-JuL-29



Interview Feedback

Homework

Presenter: Arturo Sanchez

Mobile Apps Company

3301 Windy Ridge Parkway, Suite 400, Atlanta, GA 30339

# **Interview Feedback**

## Introduction

### Elevator

My name is Arturo Sanchez Chavarria.

I have 6 Years of I.T. experience in Software Development, 5 years as android development experience and 5 Android Apps in Play.

I have experience developing mobile applications throughout the entire software development lifecycle (SDLC) right from collecting requirements, providing guidelines for design & Analysis of Customer Specifications, creating application architecture, implementation that includes various stages in development, QA, code signing and releasing to Market, Unit testing and Production support, using Agile Scrum.

I also have worked with an internal QA team on system, performance, and acceptance testing.

I have implemented design patters such as MVC, MVP and MVVM.

I have experienced integrating apps with web services using Retrofit, OkHTTP.

I have also worked with third-party APIs and web services like Google, Facebook, Twitter and YouTube Player.

### Previous Experience

I am currently working with Delta Airlines as a Senior Android Developer.

I am using Android Studio as the primary IDE to develop, test and deploy the Android application.

My responsibilities are:

* Refactoring code
* Migrate from MVC to MVP
* Transition to RxJava and Retrofit.
* Fix memory leaks.

We use design patterns such as MVP, Factory, Singleton, and Decorator.

I get to improve performance and security tracking and fixing memory leaks using Leak Canary.

For testing we implemented automated testing with Espresso.

Some of the features implemented are:

* Usign Zxing for scanning to provide various user features in airport.
* Implemented various third-party technologies to optimize the app functionality including, Firebase Cloud Messaging, SQLite, Glide, Shared Preferences, MixPanel, EventBus, RxJava

We used Bitbucket to manage Git repositories and versioning control.

## Questions Asked

### What are ContentProvider used for?

1. Sharing access to your application data with other applications
2. Sending data to a widget
3. Returning custom search suggestions for your application through the search framework using SearchRecentSuggestionsProvider
4. Synchronizing application data with your server using an implementation of AbstractThreadedSyncAdapter
5. Loading data in your UI using a CursorLoader

### How to create a ContentProvider?

1. Create a class that extends ContentProvider.
2. Create a contract class.
3. Create the UriMatcher definition.
4. Implement the onCreate() method.
5. Implement the getType() method.
6. Implement the CRUD methods.
7. Add the content provider to your AndroidManifest.xml

### How to make ContentProvider secure?

Content providers offer a structured storage mechanism that can be limited to your own application or exported to allow access by other applications. If you do not intend to provide other applications with access to your ContentProvider, mark them as android:exported=false in the application manifest. Otherwise, set the android:exported attribute to true to allow other apps to access the stored data.

To enforce your own permissions, you must first declare them in your AndroidManifest.xml using one or more <permission> elements.

### What is the data flow from provider to view?

Using an instance of content resolver to perform a query gets you a cursor, that cursor needs an interface (or any other way to communicate components) and implementing the callback in the view allows you to have the data in the view from the provider.

### What common components use a ContentProvider?

Widgets, Fragments, Activities, Services

### How can you use different data sources with provider?

Implementing different implementations depending on the URI. Each implementation should go to a different data source.

### What is Firebase? What are some services by Firebase?

Is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

Services:

* Analytics.
* Cloud Messaging.
* Authentication.
* Realtime database.
* Storage.
* Remote Config.
* Test Lab.
* Crash Reporting.
* App Indexing.
* Dynamic links.
* Invites.
* Admob.
* Firestore.

### In what format the data is stored in Firebase?

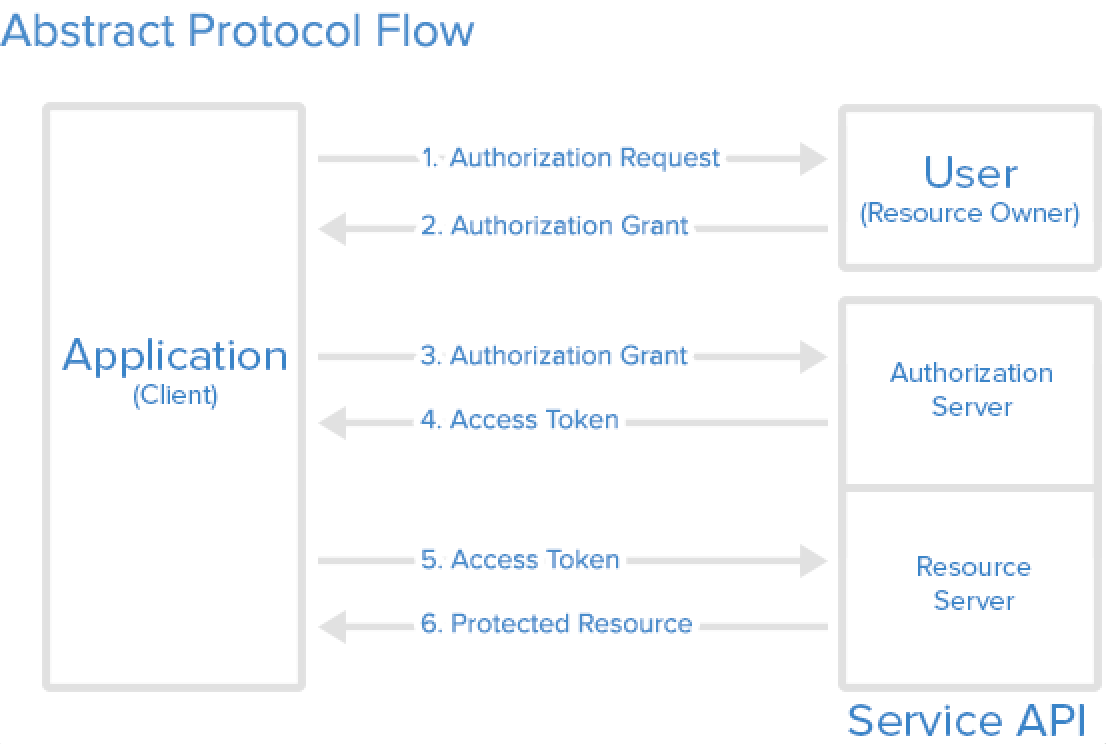
JSON Tree

### How does oAuth 2 works?

Open Authorization.

Is an open standard for token-based authentication and authorization on the Internet.

OAuth, allows an end user's account information to be used by third-party services, such as Facebook, without exposing the user's password.



### How do we use Facebook authentication with Firebase?

1. Connect your application with Firebase (using the Firebase Assistant)
   1. Create a new application (or use one already created)
   2. Create a new authentication with facebook.
      1. Set the App Key and App Secret from your facebook app
2. Create an application in facebook
   1. Get the App Id and App Secret
   2. Set the callback with the url from your firebase app
3. Integrate your facebook login SDK to your app.
4. Create an instance of FirebaseAuth.
5. Get the credential from the facebook login button
6. Call the loginWithCredential() method from the FirebaseAuth passing as a parameter the facebook credential.

### How do we use FirebaseAuth in Android Studio?

1. Connect your application to Firebase - Authentication Service.
   1. Login with a google account.
   2. Create a new project or choose an existing one.
2. Add Firebase Authentication to your app.
   1. Add dependencies to Gradle.
   2. Update Dependencies to Gradle.
3. Create interface for communication.
4. Create a Singletop manager for the FirebaseAuth.
   1. Implement methods with the FirebaseAuth instance.
5. In your activity.
   1. Create instance of the AuthManager.
   2. Implement the callbacks.
   3. Call the methods when needed.

### What is FCM? Why do we need it?

Firebase Cloud Messaging

Is a cross-platform messaging solution that lets you reliably deliver messages at no cost.

You can notify a client app that new email or other data is available to sync.

You can send notification messages to drive user re-engagement and retention.

Provide convenience and value to app users.

* Sports scores and news right on their lock screen
* Utility messages like traffic, weather and ski snow reports
* Flight check in, change, and connection information

For app publishers, push notifications are a way to speak directly to a user.

* They don't get caught in spam filters, or forgotten in an inbox
* Click-through rates can be twice as high as email.
* They can also remind users to use an app, whether the app is open or not.

### What are topics/user segments?

#### Topics

Works as Notification Channels.

* Token/device management not necessarily required.
* Unlimited number of subscribers.
* Can send to topics using the FCM API.
* Can easily subscribe/unsubscribe via the client app.

#### User Segments

Send push notifications to a specific and limited set of devices.

You can target predefined user segments or custom audiences created in Firebase Analytics.

### How do we create/subscribe to topics?

#### Create Topic

You can create a topic with http api:

https://iid.googleapis.com/iid/v1/IID\_TOKEN/rel/topics/TOPIC\_NAME

1. IID\_TOKEN = Device registration token, you can find it with following command on your android device :

String IID\_TOKEN = FirebaseInstanceId.getInstance().getToken();

2.TOPIC\_NAME = new a topic name

3.Authorization: key=YOUR\_API\_KEY. Set this parameter in the header. Look to screenshot: Creating new topic via Advanced rest client

YOUR\_API\_KEY: console.firebase.google.com

and send request and you will be receive http status "OK".

Then you can get infos about all your topics in your current project with following api :

https://iid.googleapis.com/iid/info/IID\_TOKEN?details=true

#### Subscribe to Topic

FirebaseMessaging.getInstance().subscribeToTopic("news");

### How to get location from the device?

1. Check permissions on Location
   1. If not available request for it.
2. Create an instance of FusedLocationProviderClient
   1. Execute the method getLastLocation()
   2. addOnSuccessListener()
   3. Override onSuccess()

### How to get periodic updates for location?

1. Check permissions on Location
   1. If not available request for it.
2. Create an instance of FusedLocationProviderClient
   1. Create a LocationRequest
   2. setInterval()
   3. setFastestInterval()
   4. setPriority()
3. Create a LocationSettingsRequest.Builder
   1. set the LocationRequest
4. Create a SettingsClient
   1. set it to LocationServices.getSettingsClient(context);
5. settingsClient.checkLocationSettings
   1. addOnSuccessListener()
6. Create a new LocationCallback.onLocationResult

### How to create a google map to show different markers?

1. Create a new Map Activity
2. Override onMapReady
3. Create a new LatLng (or get it from a service)
4. Set the marker with the addMarker() method from the googleMap instance.

### What is the different between HashMap and HashTable?

* Hashtable is synchronized, whereas HashMap is not. This makes HashMap better for non-threaded applications, as unsynchronized Objects typically perform better than synchronized ones.
* Hashtable does not allow null keys or values. HashMap allows one null key and any number of null values.
* One of HashMap's subclasses is LinkedHashMap, so in the event that you'd want predictable iteration order (which is insertion order by default), you could easily swap out the HashMap for a LinkedHashMap. This wouldn't be as easy if you were using Hashtable.

### LinkedList vs ArrayList?

LinkedList<E> allows for constant-time insertions or removals using iterators, but only sequential access of elements. In other words, you can walk the list forwards or backwards, but finding a position in the list takes time proportional to the size of the list. Javadoc says "operations that index into the list will traverse the list from the beginning or the end, whichever is closer", so those methods are O(n) (n/4 steps) on average, though O(1) for index = 0.

ArrayList<E>, on the other hand, allow fast random read access, so you can grab any element in constant time. But adding or removing from anywhere but the end requires shifting all the latter elements over, either to make an opening or fill the gap. Also, if you add more elements than the capacity of the underlying array, a new array (1.5 times the size) is allocated, and the old array is copied to the new one, so adding to an ArrayList is O(n) in the worst case but constant on average.

So depending on the operations you intend to do, you should choose the implementations accordingly. Iterating over either kind of List is practically equally cheap. (Iterating over an ArrayList is technically faster, but unless you're doing something really performance-sensitive, you shouldn't worry about this -- they're both constants.)

The main benefits of using a LinkedList arise when you re-use existing iterators to insert and remove elements. These operations can then be done in O(1) by changing the list locally only. In an array list, the remainder of the array needs to be moved (i.e. copied). On the other side, seeking in a LinkedList means following the links in O(n) (n/2 steps) for worst case, whereas in an ArrayList the desired position can be computed mathematically and accessed in O(1).

Another benefit of using a LinkedList arise when you add or remove from the head of the list, since those operations are O(1), while they are O(n) for ArrayList. Note that ArrayDeque may be a good alternative to LinkedList for adding and removing from the head, but it is not a List.

Also, if you have large lists, keep in mind that memory usage is also different. Each element of a LinkedList has more overhead since pointers to the next and previous elements are also stored. ArrayLists don't have this overhead. However, ArrayLists take up as much memory as is allocated for the capacity, regardless of whether elements have actually been added.

The default initial capacity of an ArrayList is pretty small (10 from Java 1.4 - 1.8). But since the underlying implementation is an array, the array must be resized if you add a lot of elements. To avoid the high cost of resizing when you know you're going to add a lot of elements, construct the ArrayList with a higher initial capacity.

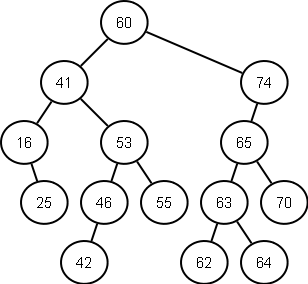
### What is a binary tree? How to create one?

A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child. A recursive definition using just set theory notions is that a (non-empty) binary tree is a tuple (L, S, R), where L and R are binary trees or the empty set and S is a singleton set.

A binary search tree is a binary tree in which every node contains a key that satisfies following criteria:

* The key in left child is less than the key in the parent node
* The key in the right child is more than the parent node
* The left and right child are again binary search trees.

Following diagram represents a binary search tree:

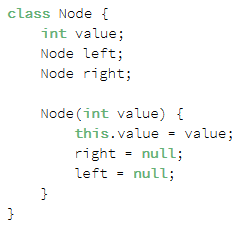
[](http://vitalflux.com/wp-content/uploads/2014/12/binary_search_tree.png)

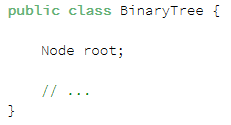
###### What are different kind of traversals?

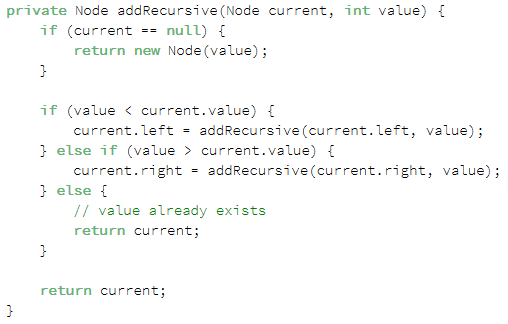
Following are three different kind of traversals:

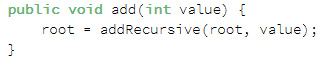
* **Preorder traversal**: In preorder traversal, the node is visted first and then, left and right sub-trees.
* **Inorder traversal**: In inorder traversal, the node is visited between left and right sub-tree.
* **Postorder traversal**: In postorder traversal, the node is visited after left and right subtrees.

Create a binary tree









### What is the fragment lifecycle?

