Md Rahman

1. What is the difference in an started service and a bound service?

2. Explain how map, flattop, and range works in RxJava.

3. What are the differences in RxJava 1 and 2?

4. Define what a observer/ subscriber relation is.

5. Define what a client/server relation is.

1. What is the difference between a started service and a bound service?

A startend service is also known as a normal service. It starts when an application starts it by calling startService(), and stopSelf() to end the service once its task is over. The running code is going to be in the onStartCommand().

Started services will then run on the background until the service is stopped or destroyed by the Android runtime system. It can continue to run in the background even if the app that started it leaves the foreground or destroyed. These services run on the main thread.

Bound services utilize a client/server relationship. Started services typically don’t return results, but bound services allows the component launching it to interact with and receive results from the service. A client (typically an application component) binds to the service via onBind(), where it can receive results from the service via different IPC implementations. Once the client gets everything it desires, we call the unbindService().

2. Explain how map, flatmap, and range works in RxJava.

Map - Takes results from an observable and maps it to a function. For example, if you have a function map(x -> 10 \* x) where x are observables, then any observable passed to the map will be multiplied by 10. So if the numbers 3, 4, 5 are passed, they come out as 30, 40, and 50.

Flatmap - Transforms observables to other types of observables. Used to map over asynchronous operations.

Range - Create an observable that emits a particular range of integers between x and y. These integers are sequential and in order.

3. What are the differences in RxJava 1 and 2?

RxJava 2 has been rewritten from scratch because it targets the Reactive-Streams API, which is an API meant to provide a standard for asynchronous stream processing with non-blocking back pressure. This helps improve our memory consumption and performance. Some other common differences are:

-RxJava 2 no longer accepts Null values for Observables

-Observable is now non-back pressured while Flowable is.

-The Single Observable type is now an interface with 3 methods:

onSubscribe, onSuccess and onError.

-Maybe is a brand new type.

4. Explain what a subscriber/observer relationship is.

This relationship is different for Rx Java 1 and 2. For 1:

An observer is an object that gets data from a data source, with the data source being an Observer. An observer can only subscribe. A subscriber is an observer that can subscribe and unsubscribe from that data source.

For 2: Observer is used to subscribe to an observable, and subscriber is used to subscribe to a flowable.

5. What is a client/server relationship?

The means of communication between a client application and the service application. The server application provides the information and the client receives it. We define how information is sent and how threads communicate with each other.