Multiscreen Android Apps

Jigisha Patel

Agenda

- Context
- Intent
- Serializable

Context

- Context is an interface to ***global information about an application environment***.
- This is an ***abstract class*** whose implementation is provided by the Android system.

- Both the `Activity` and `Application` classes extend the `Context` class.

Why Context?

- It allows access to application-specific resources and classes, as well as up-calls for application-level operations such as launching activities, broadcasting and receiving intents, etc.

- Context provides the facility to access
  - information regarding the activity and application,
  - resources, databases, and shared preferences, etc.

Types of Context

Application Context:

- Should be used to get ***a context of entire application***
- Singleton instance
- ***Tied to app lifecycle***
- Can be accessed through ***applicationContext***
- Example: you need to pass context to singleton object, use application context

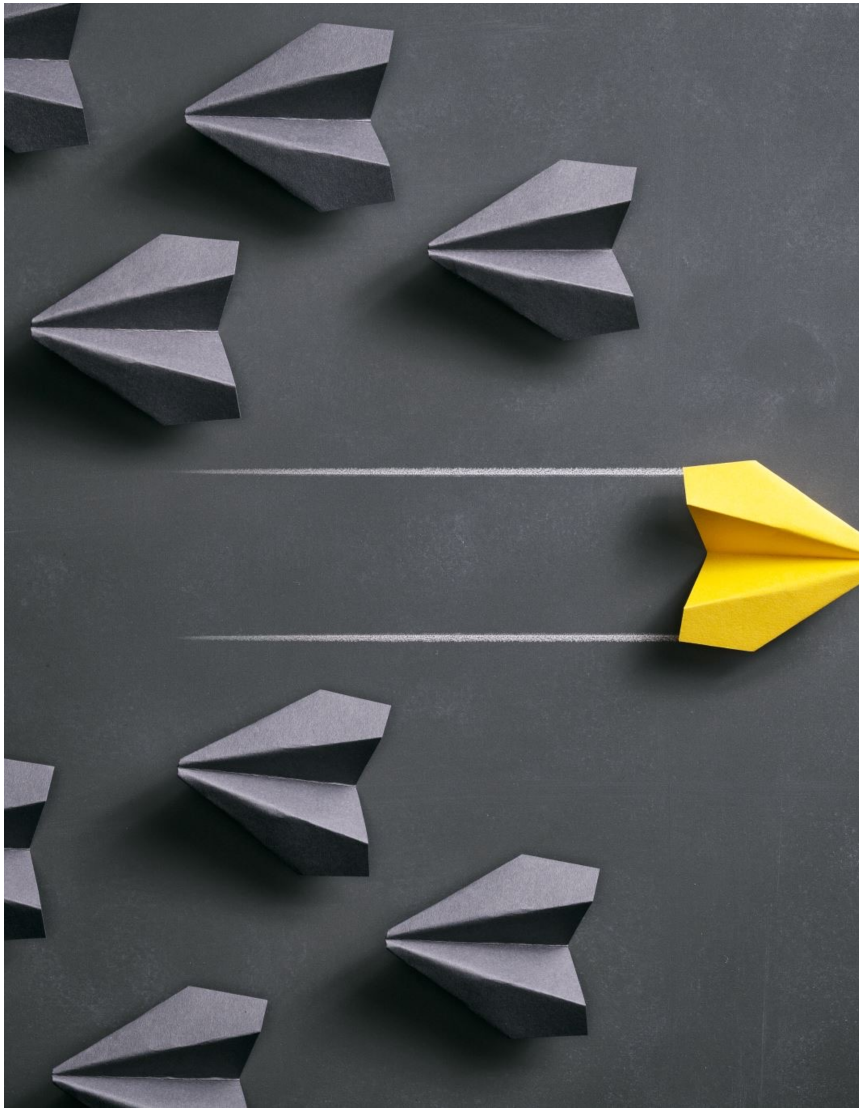
Activity Context:

- Should be used to get ***a context of activity***
- ***Tied to activity lifecycle***
- Example: If context is needed for an object whose lifecycle is attached to an activity such as toasts and dialogs

Intents

- An Intent is a messaging object you can use to request an action from another app component.

- There are 3 fundamental use cases of Intents:
  1. Starting an activity
  2. Starting a service
  3. Delivering a broadcast



Intents – Starting an activity

- An Activity represents a single screen in an app.
- You can start a new instance of an Activity by passing an Intent to **startActivity().**

- The Intent describes the activity to start and carries any necessary data.

Intent Filter

- An intent filter is an **expression** in an app's **manifest file** that specifies the type of intents that the component would like to receive.

- For instance, by declaring an intent filter for an activity, you make it possible for other apps to directly start your activity with a certain kind of intent.

- Likewise, if you do *not* declare any intent filters for an activity, then it can be started only with an explicit intent.

Building an intent

- An Intent object carries information that the Android system uses to determine which component to start, plus information that the recipient component uses in order to properly perform the action (such as the action to take and the data to act upon).

- The primary information contained in an Intent is Component name, Action, Data, Category, Extras and Flags.

Component name

- The name of the component to start.
- This is **optional**, but it's the critical piece of information that makes an intent ***explicit***, meaning that the intent should be delivered only to the app component defined by the component name.
- Without a component name, the intent is ***implicit*** and the system decides which component should receive the intent based on the other intent information (such as the action, data, and category).
- If you need to start a specific component in your app, you should specify the component name.

Extras

- Extras are key-value pairs that carry additional information required to accomplish the requested action.
- You can add extra data with various putExtra() methods, each accepting two parameters: the key name and the value.
- You can also create a Bundle object with all the extra data, then insert the Bundle in the Intent with putExtras().

- For example, when creating an intent to send an email with ACTION_SEND, you can specify the to recipient with the EXTRA_EMAIL key, and specify the subject with the EXTRA_SUBJECT key.

Serializable interface

- Serialization an object means to convert its state to a byte stream so that the byte stream can be reverted back into a copy of the object.
- Deserialization is the process of converting the serialized form of an object back into a copy of the object.
- Serializability of a class is enabled by the class implementing the java.io.Serializable interface.

- In order to pass the objects of the user-defined class using Intent to the next Activity, the class should implement the Serializable interface.
- The serialization interface has no methods or fields and serves only to identify the semantics of being serializable.

References

- https://developer.android.com/guide/components/intents-filters
- https://developer.android.com/guide/components/intents-common
- https://developer.android.com/reference/android/content/Context
- https://blog.mindorks.com/understanding-context-in-android-application-330913e32514