# Parcelable Interface

Jigisha Patel

# Parcelable

- Parcelable objects are intended to be used across process boundaries such as with IPC/Binder transactions, between activities with intents, and to store transient state across configuration changes.

- Parcelable is an interface for classes whose instances can be written to and restored from a Parcel.

# Parcel

- Parcel is container for a message (data and object references) that can be flattened and unflattened to read from and to write to when exchanged between Activity classes.

- Parcel is not a general-purpose serialization mechanism, and you should never store any Parcel data on disk or send it over the network.

# Using Parcelable to send data between activities

- In some cases, you may need a mechanism to send composite or complex objects across activities.
- You can use putParcelableArrayListExtra() with intents to send data to next activity.

- In such cases, the custom class should implement Parcelable, and provide the appropriate writeToParcel(Parcel, int) and readFromParcel(Parcel) method.
- It must also provide a non-null field called CREATOR that implements the Parcelable.Creator interface, whose createFromParcel() method is used for converting the Parcel back to the current object.

# Methods of Parcelable Interface

- writeToParcel(Parcel, int) – Flatten this object into a Parcel
- createFromParcel() - used for converting the Parcel back to the object
- describeContents() - describe the kinds of special objects contained in this Parcelable instance's representation

# References

- https://developer.android.com/reference/android/os/Parcelable
- https://developer.android.com/guide/components/activities/parcelables-and-bundles
- https://developer.android.com/reference/android/os/Parcel