



OUR BLOG



PASSING OBJECTS VIA INTENT IN ANDROID

 **Shweta** January 3, 2017

2 Comments

[ACTIVITY](#) [ANDROID](#) [ANDROID DEVELOPMENT](#) [ANDROID TIPS](#) [ANDROID TUTORIAL](#) [PARCELABLE](#) [SERIALIZATION](#)



Hi Coders,

Android developers generally pass data between activities and components. If data is primitive type then it is easy to pass via intent. However passing an object is not as simple as a primitive type. Whenever you want to pass an object, you can either make your objects Parcelable or Serializable. Both approaches are a way to serialize and deserialize an object in android. Let's look at the both approaches.

Parcelable is an interface. A class who implements Parcelable can write to and read from a Parcel.

You need to follow 3 points to create a Parcelable class.

Search

DESIGN IQ QUIZ

CONTACT US

Name

Email Address

Company

Your Budget

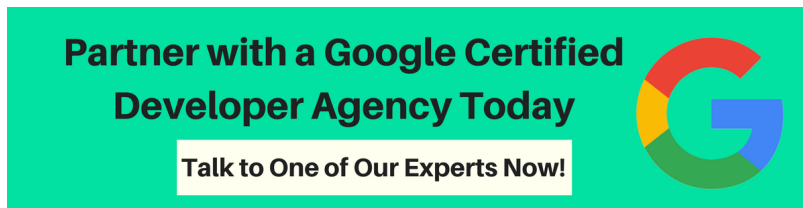
Message

GET A QUOTE



Parcelable.Creator interface.

– Override writeToParcel method and write member variable in Parcel. Make sure to read variables in the same sequence in which they are written in Parcel. Order of read and write matters.



Below is the typical implementation of Parcelable.

```
public class Movie implements Parcelable {  
    // Member variables  
    private String title;  
    private String overview;  
    private String release_date;  
    private String original_language;  
    private int id;  
}
```

// In constructor you will read the variables from Parcel. Make sure to read them in the same sequence in which you have written them in Parcel.

```
public Movie(Parcel in) {  
    title = in.readString();  
    overview = in.readString();  
    release_date = in.readString();  
    id = in.readInt();  
    original_language = in.readString();  
}
```

// This is where you will write your member variables in Parcel. Here you can write in any order. It is not necessary to write all members in Parcel.

```
@Override  
public void writeToParcel(Parcel dest, int flags) {  
    // Write data in any order  
    dest.writeString(title);  
    dest.writeString(overview);  
    dest.writeString(release_date);  
    dest.writeInt(id);  
    dest.writeString(original_language);  
}
```

// This is to de-serialize the object

```
public static final Parcelable.Creator<Movie> CREATOR = new Parcelable.Creator<Movie>(){  
    public Movie createFromParcel(Parcel in) {
```





```
public Movie[] newArray(int size) {  
    return new Movie[size];  
}  
};  
}
```

Now you are good to pass objects of Movie type from one activity to other activity.

This is how you can pass a Parcelable object using Intent.

```
// In activity or fragment  
Movie movie = new Movie();  
// using context and next component class to create intent  
Intent intent = new Intent(this, NextActivity.class);  
// using putExtra(String key, Parcelable value) method  
intent.putExtra("parcel_data", movie);  
startActivity(intent);
```

You can access this data in NextActivity –

```
public class NextActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // Using getParcelableExtra(String key) method  
        Movie movie = (Movie) getIntent().getParcelableExtra("parcel_data");  
    }  
}
```

Serializable – To serialize an object means to convert its state to byte stream so that it can be read to get the object back. To make a class Serializable, the class needs to implement the Serializable interface. Rest everything is handled by Java. This is how you will make your class Serializable.

```
// Implementing Serializable in the previous example only  
public class Movie implements Serializable {  
    private String title;  
    private String overview;  
    private String release_date;  
    private String original_language;  
    private int id;  
}
```

This is how you will pass a Serializable object –

```
// In activity or fragment  
Movie movie = new Movie();
```

[in](#)[f](#)[t](#)[@](#)[G+](#)[About Us](#)[Why Us](#)[Our Offerings](#)[News](#)[Blog](#)[Join Us](#)[Contact](#)



```
// using putExtra(String key, Serializable value) method  
intent.putExtra("serialize_data", movie);  
startActivity(intent);
```

While accessing this data in NextActivity –

```
public class NextActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // Using getSerializableExtra(String key) method  
        Movie movie = (Movie) getIntent().getSerializableExtra("serialize_data");  
    }  
}
```

Serializable is a marker interface that is why there is no method to implement. Using Serializable is easier than Parcelable. Developer need not to write much code when they use Serializable and even code looks cleaner. Still google recommend to use Parcelable instead of Serializable. Why?

Different between Parcelable and Serializable

Serializable use reflection that is why it is a slow process.

Serializable creates a lot of temporary objects and cause quite a bit of garbage collection.

Conclusion, Parcelable is faster and uses less memory but it requires bit coding and time. Whereas Serialization needs less coding and quick to implement but it is slower and consumes more memory.

Hope this is useful information for you!! If you have any doubt you can post the comment below. Enjoy coding.

Reference links –

<https://developer.android.com/reference/android/os/Parcelable.html>

<https://docs.oracle.com/javase/tutorial/jndi/objects/serial.html>

in

f

tw

ig

G+



FREE EBOOK

A Complete Guide to Planning your Mobile Strategy in 2017!

Mobile is revolutionizing the way business is done. Use this strategy guide to keep yourself ahead of the game!

GET IT NOW!





All stories by this author

PREVIOUS ARTICLE

CHALLENGES IN MOBILE APPLICATION TESTING

NEXT ARTICLE

TIME MANAGEMENT FOR PROJECTS

in

f

twitter

ig

G+

2 Comments Techjini

1 Lo

Recommend 2 Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

Fabricio • 2 months ago

Thanks man, your article helped me a lot.

^ | ▾ • Reply • Share ›

Eric Jerome • 3 months ago

How to pass data objects using kotlin?

^ | ▾ • Reply • Share ›

Subscribe Add Disqus to your site Add Privacy

HOME

About Us
News
Blog
Join Us
Contact

WHY US

Startups
Brands
Enterprises
Agencies

OUR OFFERINGS

Mobile App Development
Web Development
Cloud Solutions
UX and UI Design
Digital Media Services
IOT Application
Virtual Reality
Data Science

LATEST BLOGS

Exploring the Practicality Of AR
How does the BYOD Policy affect Enterprise Mobility?
Healthcare Interoperability and why should you care (a...
Can AR and VR Help Marketing Campaigns?
How will Chatbots Drive Value to Your Enterprise in...



UNITED STATES

Suite #100, 510 Thornall Street
Edison, New Jersey 08837
Phone: +1 (609) 785-1151
Fax: +1 (609) 228-7549
E-mail: sales@techjini.com

INDIA

#948, Gaayathri Chambers,
2nd Floor, 24th Main
J P Nagar 2nd Phase,
Bangalore 560078
Phone: +91 (80) 415-07899
E-mail: sales@techjini.com



in

f



G+

