

# Android Intent

Android Intent lets you navigate from one android activity to another.

Android Intent can be defined as a simple message objects which is used to communicate from 1 activity to another.

Intents define intention of an Application . They are also used to transfer data between activities.

An Android Intent can be used to perform following 3 tasks :

1. Open another Activity or Service from the current Activity
2. Pass data between Activities and Services
3. Delegate responsibility to another application. For example, you can use Intents to open the browser application to display a URL.

Intent can be broadly classified into 2 categories. There are no keywords for this category and just a broad classification of how android intents are used.

## Explicit Android Intent

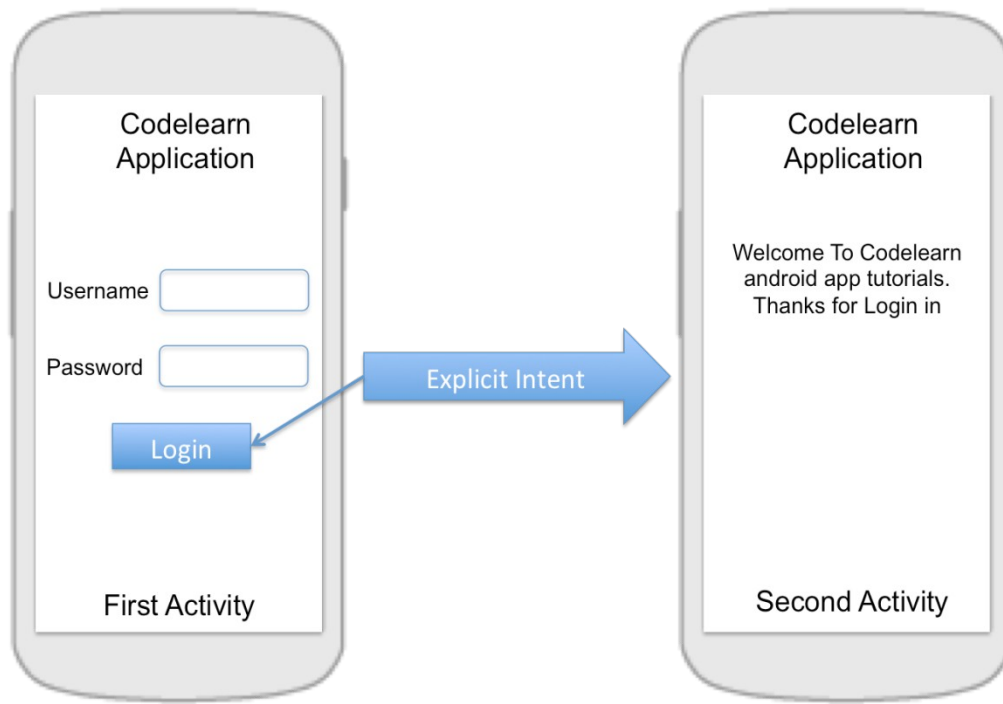
Explicit Android Intent is the Intent in which you explicitly define the component that needs to be called by Android System.

When you open an activity from another activity in the same Android app, you use Explicit Intents. Let us understand through an example :

```
Intent CodeLearnFirstIntent = new Intent (getApplicationContext(),  
SecondActivity.class);
```

If you notice here, you are passing the SecondActivity class name as an identifier for this intent.

Let's see an example



## Implicit Android Intent

Implicit Android Intents is the intent where instead of defining the exact components, you define the action you want to perform. The decision to handle this action is left to the operating system. The OS decides which component is best to run for implicit intents.

Whenever you delegate responsibility to another application from your application, you use Implicit Intents.

Let us see an example:

```
Intent sendIntent = new Intent();  
sendIntent.setAction(Intent.ACTION_SEND);
```

This is typically used when you want to share the data from 1 application to another. Sharing data over email, sms, social network etc. is a classic example of this category. See the image below, this is a implicit intent for sending email

## Ways in which intents can be used in android

### Open Another Activity

```
Intent openNewActivity = new Intent(getApplicationContext(),  
Activity2.class);  
startActivity(openNewActivity);
```

With these 2 lines of code, you can open an activity from another in your app.

Let us understand the parameters being passed to create an Intent object :

**Application Context:** we need to pass the context to be used by the Intent. You can either use **getApplicationContext()** method, or use **this** qualifier.

**Activity Class:** This is the class which you want to open using Intent.

Once we have the Intent object ready, we can do a lot of things with it. We will see in the next section how we can pass data using this Intent object. You can also use Intent's `startActivity` method to start an activity whenever you want. In our example, we are doing this at a button click event.

### Pass Data

Have a look at the same `openActivity` method. Notice the next line after the Intent object is created.

There are multiple ways to store data in an Intent. You can use an explicit bundle and set bundle using **putExtra**, or you can directly use **putExtra**.

**putExtra** is the method which is used to store data in an Intent object. This data can be of different types : string, char, Boolean, Bundle etc. This data gets set as key value pair which you can retrieve in the called activity.

Let's see this example:

```
openNewActivity.putExtra("UserName", "Pranay");  
openNewActivity.putExtra("isRegistered", true);
```

After creating the intent object, we just invoked the `putExtra` method. In the first line, we are setting String data with key "UserName" and in the second line we are setting the Boolean data. We will learn to retrieve this data in our next section.

## Alternate Method to put data

In the previous method, we were setting the data directly into the Intent object. A lot of developers do this alternatively. They set the data in a Bundle and then associate the created bundle with an Intent.

Bundle object is used to pass data between activities. You can still use putExtra method to associate a bundle with an Intent.

Lets see the example code

```
Bundle dataBundle = new Bundle();
dataBundle.putString("BundleUserName", "Pranay");
openNewActivity.putExtra("data", dataBundle);
```

Here we create the bundle object and put our data using various methods like putString, putBoolean etc.

## Retrieving Data

In the above example, we have seen how we pass data from one activity to another, but once data is passed it also needs to be collected back into other activity.

Open Activity2 class from the example code.

The code below describes as to how we collect data into the Intent object.

```
Intent intentObject = getIntent();
String userName = intentObject.getStringExtra("UserName");
boolean isRegistered = intentObject.getBooleanExtra("isRegistered", true);
```

The first step to extract the data is to get the Intent object using getIntent() method.

Once you get hold of this Intent object, you can call different methods to get the data back. For example, getStringExtra retrieves the string data. In case if there is no data, you will get null or the default value which is already set. With getBooleanExtra, the second parameter is the default value which will be returned if there is no value with that key.

## Opening Other App

So far we have seen how to open one activity from another and how to pass and consume data. Let us now see an example as how to open other apps from our app.

## Opening Webpage

Given the following code :

```
Intent webIntent = new Intent(Intent.ACTION_VIEW,
Uri.parse("http://www.codelearn.org"));
startActivity(webIntent);
```

In the first line, we create a new Intent object. But this time we use Action instead of Activity class. The intent ACTION\_VIEW requests the Android operating system to display data to the user. Based on the context passed, Android automatically decides which application to use.

If user has multiple applications which can be used for this action, android will display a list of all applications and ask the user to choose the one they want.

### **Sharing Data**

Similar to opening webpage we have shareData method in which you can see how we use ACTION\_SEND to share data over email or any social network.

In this chapter, we have seen how we can use android intent to communicate between different activities or to pass data. Also, we now know how to use Intents for opening other applications from our app.