Java II

:: Activity

Activity: Intents (1h)

(Due: Thu, 08 Aug | Status: Not Completed)

Now that we understand the basics of what makes an Activity, we can begin to look at how Activities use Intents to communicate with the operating system as well as with other components in the system.

Objectives & Outcomes

Upon completion of this activity, the learner will be able to:

- Understand how to create Intents to manage application behavior.

Level of Effort

This activity should take approximately 60m to complete. It will require:

- 60m Research
- 0m Prep & Delivery
- 0m Work

If you find that this activity takes you significantly less or more time than this estimate, please contact me for guidance.

Reading & Resources

ADR: Intents and Intent Filters (helpful)

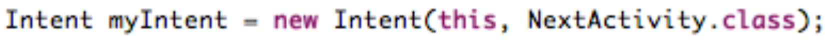
Android Reference for Intents.

Instructions

The Intent class is used for communication within the Android system. The name Intent is appropriate for this class as it is used to notify the operating system of your intention for something to happen. Up until now you have given instructions to the operating system through the manifest file. For example, your manifest instructs the operating system to start your Activity when the user clicks on your application icon. This works well when a certain behavior of your system never changes, but that will not always be the case. For example, you may want your application to display a different screen based on what button a user clicks on, or what data is entered. This behavior is accomplished through an Intent. In addition, the Intent can be used to pass data between components in your application.

Creating an Intent

An Intent can be instantiated as needed in your code to command the operating system to perform an action. As you are already familiar with the Activity class, this lesson will teach how an Activity uses an Intent to launch another Activity. The Intent also can be used with other components such as Services and Broadcast Recievers, as you will later learn. Below is an example of a constructor for an Intent to launch an Activity.

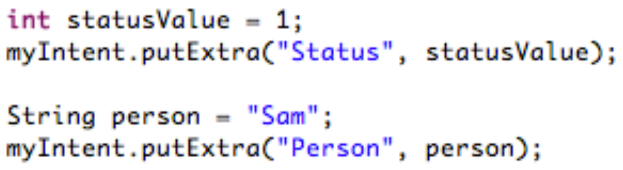


```
Intent myIntent = new Intent(this, NextActivity.class);
```

In the above constructor, the first parameter "this" is the current application context (ie. your calling Activity), and the second parameter is the class that the operating system is to launch, in this case NextActivity which is extended from the Activity class. Note that the launched component NextActivity must also be defined in the Manifest in order for the operating system to launch it, however it will remain dormant until this Intent is "started".

Passing Data through an Intent

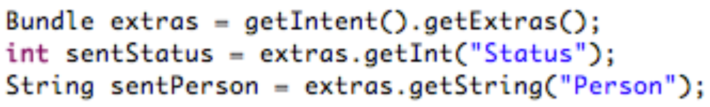
The Intent provides a means of passing data from one component to another. For example, an application may have a user enter data on one screen, and provide results on another. In this example the first Activity needs to pass the user entered data to the second Activity so it can display the requested results. The Intent makes it very simple to pass any type of data. It uses the key-value pair that is commonly used in software. The first parameter of putExtra is the key, a unique string name used to access the data. The second parameter is the value to pass with that key.



```
int statusValue = 1;
myIntent.putExtra("Status", statusValue);

String person = "Sam";
myIntent.putExtra("Person", person);
```

Notice how the same method "putExtra" is used for different types of data. This makes it very easy to pass any type of data into the intent, including arrays and classes. The created Intent is passed to the component started by the operating system, so the new component, in this case NewActivity, can now retrieve data from the Intent as follows.

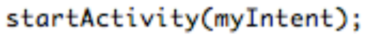


```
Bundle extras = getIntent().getExtras();
int sentStatus = extras.getInt("Status");
String sentPerson = extras.getString("Person");
```

The data is retrieved by first getting the data extras from the Intent through the "getExtras" method which returns a Bundle, which is a container for data passed between processes. Then the data is retrieved by passing the key string in the input parameter of the "get" method for the specific type of data.

Submitting the Intent to the Operating System

Now that you have created an Intent and loaded it with data to send to another Activity, how do you submit your Intent to the operating system so it can be acted upon? In the case of an Activity, you will call the method "startActivity" and pass your Intent in the input parameter. If you are using an Intent to start a Service, as you will later learn, you will call "startService", also passing the Intent in the input parameter. The Intent is a passive class as it just holds information and has methods to access that information. All of the control is in the operating system acting on that information.



```
startActivity(myIntent);
```

Deliverables

There is no deliverable information associated with this activity