Using Intents and notifications

Objectives

- Install google play services in the Android Emulator.
- Using Intent class to apply different functionalities.
- How to create Toast Notifications.
- How to incorporate Google Maps into an application.
- How to incorporate Gmail into an application.
- How to incorporate Dial Up into an application.
- How to post notifications in the Notification Bar.

Intents In android:

It is a data structure that represents an operation to be performed. Intents is constructed by one component that wants some work and it is received by one activity that can perform that work. For example, we can use the intent Class to start new activity.

Intent Fields:

1-Action: String represents a desired operation:

ACTION_MAIN: start as initial activity of app

ACTION_DIAL: dial a number

ACTION_EDIT: display data to edit

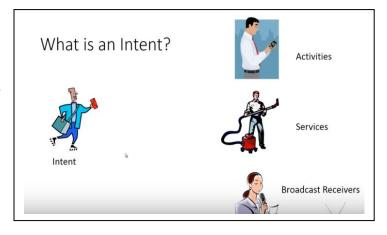
ACTION SYNC: synchronize device data with server

You can set the action of the activity by two ways:

A-by passing it in the constructor:

Intent newIntent = new Intent(Intent.ACTION_DIAL);

B-by setAction method



```
Intent newIntent = new Intent();
newIntent.setAction(Intent.ACTION DIAL)
```

2-Data: data associated with the intent. It is formatted as a uniform resource identifier (URI). For example, to dial a number, you can use the following URI

```
URI.parse("tel:+1555....");
```

Intent newIntent = new Intent(Intent.ACTION_DIAL, URI.parse("tel:+1555....");

3-category: additional information about the components that can handle the intent. For example,

CATEGORY LAUNCHER: can be the initial activity a task.

4-component: the component that should receive this intent. Use this field when there is exactly one component that should receive the intent.

5-Extras: what additional information you need to provide (key/value pairs)

There are two types of intents:

- Explicit intents specify the component to start by name (the fully-qualified class name). You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start. For example, start a new activity in response to a user action or start a service to download a file in the background.
- Implicit intents do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it. For example, if you want to show the user a location on a map, you can use an implicit intent to request that another capable app show a specified location on a map.

Implicit Activation of the activity: when the activity to be activated is not explicitly named, android tries to find activity that match the intent by comparing the contents of the intent to the intent filters declared in the manifest file of other apps on the device. This process called **intent resolution**. Android package manager is responsible for deciding which component is best suited to handle your intent.

Intent resolution relies on two kind of information

1-An Intent describing a desired operation.

2-Intents filters which describe which operations an activity can handle. It is specified either in Android Manifest file or programmatically.

What do you write inside an intent filter?

1-one or more action to indicate what actions can be handled by your component

2-one or more category to indicate what grouping does your component support

3-data to indicate what data can be handled by your component

User notifications:

Notifications basically are messages outside the user interface of the application, for example, if you download a book from the internet, you need to use the application while the book is downloading and let the user know when the download finishes. So the developer should display the message to user contains that information.

In this experiment we will talk about two kinds of user notifications

1-Toast Messages: Transitory messages that pop up on the current window. It automatically fade into & out of the view without user interaction or response.

2-Notification Area or status bar notification at the top of the device

You can create Toast messages by instantiating a Toast object by calling

Toast.makeText(context,text,duration)

Then if you would like to show Toast message by calling

Toast.show()

Look at the following example

```
Context context = getApplicationContext();
CharSequence text = "Hello toast!";
int duration = Toast.LENGTH_SHORT;

Toast toast = Toast.makeText(context, text, duration);
toast.show();
```

Android provides the notification area for alerting users about events. It also provides a notification drawer that user can pull down to see more detailed information about notifications. The operations on the notification area are managed by system service.

Procedure:

Part A: using Intent class to apply different functionalities.

1- Create new Button in the main activity and create new object of Intent in on Click method. Set the action and data to request Dial Up application.

```
Button dialButton = (Button) findViewById(R.id.dialButton);
dialButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent newIntent = new Intent();
        newIntent.setAction(Intent.ACTION_DIAL);
        newIntent.setData(Uri.parse("tel:+150000"));
        startActivity(newIntent);
    }
});
```



Figure 1: Request Dial Up application using Intent

2- Create new Button in the main activity and create new object of Intent in on Click method. Set the action and data to request Gmail application.

```
Button sendButton = (Button) findViewById(R.id.sendButton);
sendButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent();
        intent.setAction(Intent.ACTION_SENDTO);
        intent.setType("message/rfc822");
        intent.setData(Uri.parse("mailto:"));

        intent.putExtra(Intent.EXTRA_EMAIL, "mhmdateeq@yahoo.com");
        intent.putExtra(Intent.EXTRA_SUBJECT, "my subject");
        intent.putExtra(Intent.EXTRA_TEXT, "content of the message");

        startActivity(intent);
```

}
});

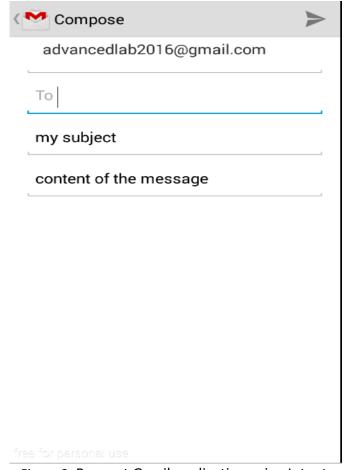


Figure 2: Request Gmail application using Intent

Send the message to your email and check it to be sure that the message was sent successfully.

3- Create new Button in the main activity and create new object of Intent in on click method. Set the action and data to request Google Map application.

```
Button mapButton = (Button) findViewById(R.id.mapButton);
mapButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent myIntent = new Intent();
        myIntent.setAction(Intent.ACTION_VIEW);
        myIntent.setData(Uri.parse("geo:19.076,72.8777"));
        startActivity(myIntent);
    }
});
```



Figure 3: Request Google Map application using Intent

Part B: Use the user notification:

1-Define the following constants in the main activity of your project.

```
private final String tickerText = "Notification Message";
private final String contentTiltle = "Notification";
private final String contentText = "you have been notified !";
```

2-Create new Button in the main activity and create new object of Notification.Builder to show the message in the notification area.

```
Button button = (Button) findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Notification.Builder builder = new
        Notification.Builder(getApplicationContext());
        builder.setTicker(tickerText);
        builder.setAutoCancel(true);
        builder.setContentIntent(mContentIntent);
        builder.setContentTitle(contentTitle);
        builder.setContentText(contentText);
        builder.setSmallIcon(R.drawable.images);
```

The output should be like figure 4.

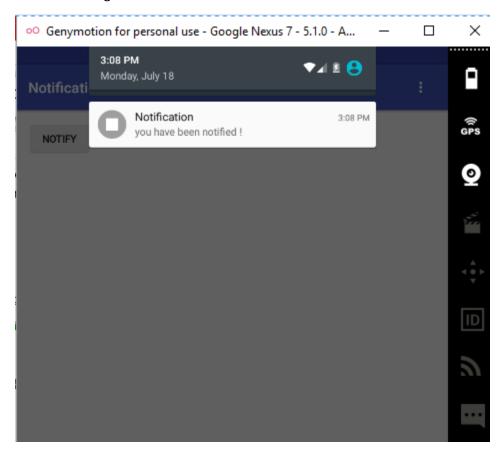


Figure4:show the message in the notification area