



# Mobile Programming Laboratory

ANDROID  
Intent



# Teachers

Ing. Tarquini Francesco, Ph.D

Ph.D in Computer Science Engineering

[francesco.tarquini@univaq.it](mailto:francesco.tarquini@univaq.it)

Ing. D'Errico Leonardo

Ph.D Student in Computer Science Engineering

[leonardo.derrico@graduate.univaq.it](mailto:leonardo.derrico@graduate.univaq.it)



# Teaching Materials

Available on MOODLE platform

<http://www.didattica.univaq.it>

Google Drive Repository

<https://drive.google.com/drive/folders/1ISqZfn0i9Ub3eWNXbvW00rd0hD9ya8OL?usp=sharing>



# Topics

- Intent
  - Change screen
  - Open an Application



# Intent

The developer can use the Intent class to request an action of another Activity (in the same or different application).

Intent facilitates the communication between components.

Use Intents in these fundamental use cases:

- Starting an Activity

- Starting a Service

- Delivering a Broadcast



# Change the screen

Using Intent the developer can open a new Screen (Activity) in foreground.

```
Intent intent = new Intent(<context>, <JavaActivityClass.class>);
```

## Example

```
Intent intent = new Intent(getApplicationContext(), OtherActivity.class);
```

After the creation of the intent object the developer have to start the action to open a new Activity using the Context method:

```
startActivity(intent);
```

The Intent class permits to the developer to send data Activity putting them on a Bundle object

```
Intent intent = new Intent(getApplicationContext(), OtherActivity.class);  
intent.putExtra("key", "value to pass");  
startActivity(intent);
```



# Change Screen

The Bundle is an object that permits to encapsulate inside it the values of the primitive types and objects of classes extending the parcelable or serializable class.

```
intent.putExtra("byte", 0xFF);  
intent.putExtra("boolean", true);  
intent.putExtra("char", 'A');  
intent.putExtra("double", 10.0D);  
intent.putExtra("float", 10.0F);  
intent.putExtra("integer", 10);  
intent.putExtra("long", 10L);  
intent.putExtra("short", 10);  
intent.putExtra("String", "String value");
```

It is allowed the array structure too.



# Change Screen

The developer retrieves the Intent, used to open another Activity, using the Activity method `getIntent()` and inside it can get the passed values.

```
Intent intent = getIntent();  
String stringValue = intent.getStringExtra("String");  
int intValue = intent.getIntExtra("integer", 0);  
boolean booleanValue = intent.getBooleanExtra("boolean", false);
```





# Change Screen

If the developer are waiting a result from another Activity, he can use to start the second Activity the following method:

```
startActivityForResult(intent, REQUEST_CODE);
```

In the opened Activity to set the result value the developer can use the Activity method

```
setResult(resultCode, data);
```

When opened Activity is in Stop, the first Activity return in foreground and an Activity callback is called using the value of setResult() method

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
}
```

The REQUEST\_CODE is an integer to link the result at the right request.



# Open an Application

The developer can use the Intent in implicit way, asking to the system if exist one or more applications to perform the Intent action.

Example: Send a message

```
Intent sendIntent = new Intent();  
sendIntent.setAction(Intent.ACTION_SEND);  
sendIntent.putExtra(Intent.EXTRA_TEXT, "My Text");  
sendIntent.setType("text/plain");
```

Example: Take a picture

```
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
startActivityForResult(intent, REQUEST_CODE);
```