

Anotacion Extra

La documentación de esta anotación la veremos en el siguiente enlace:

<https://github.com/androidannotations/androidannotations/wiki/Extras>

La anotación "@Extra" la usaremos para recibir mensajes (por ejemplo) desde otra activity.

Esta anotación tiene tres maneras para usarse, en una variable global a la clase, en un método con un único parámetro de entrada y en un método con más de un parámetro de entrada.

Manifest

Recordemos que cada nueva actividad se debe registrar en el AndroidManifest.xml haciendo una nueva entrada en la sección de actividades (dentro del tag application):

```
<activity android:name="com.example.openwebinar.myapplication.ac
</activity>
```

Variable global a la clase

La siguiente variable en la actividad de destino será la que reciba la información:

```
@Extra
<DATA_TYPE> myReceivedMessage;
```

Para poder llamar a la activity y que ésta reciba la información con la anotación "@Extra" la deberemos invocar empleando:

```
<ACTIVITY_GENERATED_ANDROID_ANN>_.intent(this).myReceivedMessage(<ME
<ACTIVITY_GENERATED_ANDROID_ANN>_ -> Será la clase generada por
myReceivedMessage -> Es el nombre de la variable
<MESSAGE> -> Será la info a enviar
```

Este comando lanzará la nueva activity (usando ".start()") e inicializará la variable "myReceivedMessage" con el mensaje <MESSAGE>.

Método con un único parámetro de entrada

Para emplear esta forma es necesario definir el siguiente método en la actividad de destino

```
@Extra
void myMethodMessage(String myStringExtra){...
```

Para usar este extra será necesario llamar a la actividad como se muestra a continuación:

```
<ACTIVITY_GENERATED_ANDROID_ANN>_.intent(this).myMethodMessage(<MESS  
    <ACTIVITY_GENERATED_ANDROID_ANN>_ -> Será la clase generada por  
    myMethodMessage                      --> Es el nombre del método qu  
    <MESSAGE>                            --> Será la info a enviar
```

Esta línea lanzará la nueva activity usando el método ".start()" y lanzará el método "myMethodMessage" con la información <MESSAGE>.

Método con más de un parámetro de entrada

Para esta opción es necesario definir el siguiente método en la actividad de destino:

```
void myMethodMessage(@Extra String myStringExtraOne, @Extra String m
```

Para usar esta extra For use the extra info you need to call the new activiy using:

```
<ACTIVITY_GENERATED_ANDROID_ANN>_.intent(this).myMethodMessage(<MESS  
    <ACTIVITY_GENERATED_ANDROID_ANN>_ --> Será la clase generada por  
    myMethodMessage                  --> Es el nombre del método qu  
    <MESSAGE_ONE>                    --> Será la info a enviar  
    <MESSAGE_TWO>                    --> Será la info a enviar
```

Este comando lanzará una nueva actividad nueva (usando ".start()") invocando al método "myMethodMessage" pasando el mensaje <MESSAGE_ONE> y <MESSAGE_TWO>.

Desarrollo

Crea una nueva activity llamada "SecondActivity" al mismo nivel que la clase "MainActivity". Esta clase debe estar anotada con "@EActivity" y deberá heredar de "Activity" (u otro tipo de activity). Ambos casos deberán estar anotados con "@Extra".

```
@EActivity  
public class SecondActivity extends Activity {  
  
    @Extra  
    String myReceivedMessage;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        this.finish();  
    }  
}
```

```

    }

    @Extra
    void myMethodMessage(String myStringExtra){
        System.out.println("Received Message =====> " + myRece
        System.out.println("INFO ++++++++> " + myStri
    }

    void myMethodMessage(@Extra String name, @Extra String surname){
        System.out.println("Received Message =====> " + myRece
        System.out.println("NAME *****> " + name);
        System.out.println("SURN *****> " + surnam
    }
}

```

En la clase de "MainActivity" escribiremos:

```

private final String MESSAGE = "Avengers are really cool";
private final String MESSAGE_NEW = "I love Avengers' Lego";
private final String MESSAGE_NAME = "Iron";
private final String MESSAGE_SURNAME = "Man";

```

Y añadiremos en el "onCreate":

```

SecondActivity_.intent(this).myReceivedMessage(MESSAGE).start() ;
SecondActivity_.intent(this).myMethodMessage(MESSAGE_NEW).start() ;
SecondActivity_.intent(this).myMethodMessage(MESSAGE_NAME, MESSAGE_S

```

Launch it!

Finalmente en el logger se deberá ver:

```

Received Message =====> ?
NAME *****> ?
SURN *****> ?

```

Donde "?" serán los datos que hemos enviado desde el "onCreate" de la "MainActivity".