

## Lab 5 – Basic App Development II

This lab is the extension of the previous lab **Basic App Development I** for Android apps. You will add a functionality to the **Send** button and additional requirements and components to your program that will help you in the design of the Water Tank Controller. You will learn about how to build and receive an Intent, create another Activity and respond on the click of the **Send** button

### 1.1 Another Activity

In the previous lab you made a program that has a **Send** button, a text field and a checkbox. The button does not have any functionalities... until now. Your app is supposed to respond when the button is clicked by sending the text written in the text field to another activity.

- **Respond on the Button click**

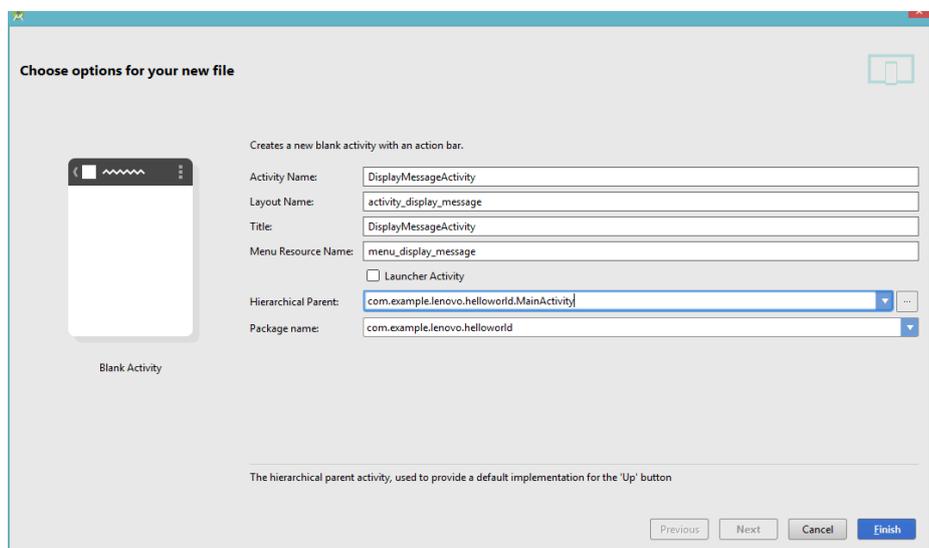
1. In the **res/layout <Button>** tag add **android:onClick** with the value **sendMessage**
2. Go to the .java file created in the beginning of the creation of your project
3. Write the sendMessage() method which is called when the user click the Send button. The method must be void and public. It will have only one parameter. The parameter is the View. For now this method is empty until you learn and be able to create an Intent in order to start an Activity.

- **New Activity**

In the **OnCreate()** method is where the Activity receives the intent with the message. All subclasses of Activity must implement this method.

This method will be automatically created when you will create your new **DisplayMessageActivity**.

Right click on the project package -> **New -> Activity -> Blank Activity**



## 1.2 Intents

An Intent is an abstract description of an operation to be performed, not for event notification. They for example can be used to start a single activity.

Intent is constructed by one component that wants some work done. Received by one activity that can perform that work.

- Setting the Intent Action

```
import Android.content.Intent;
```

```
...
```

```
Intent intent = new Intent(this, DisplayMessageActivity.class);
```

The two parameters in the constructor are a Context and a class

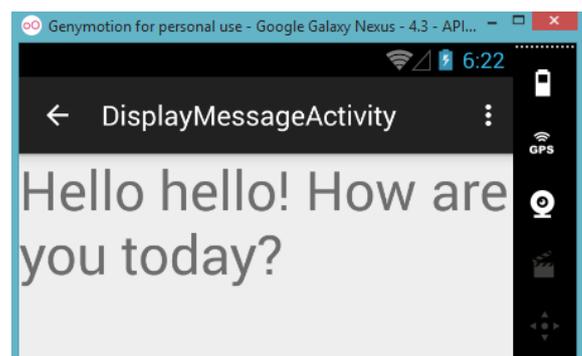
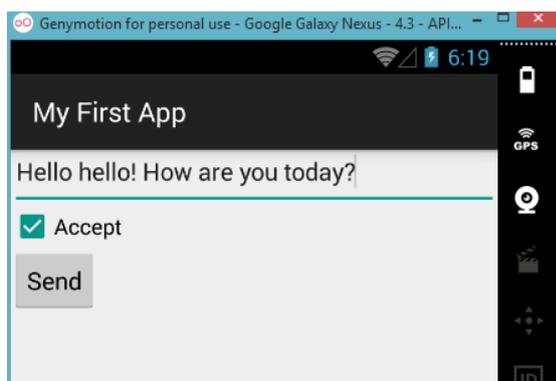
- A Context – **this** is used because the **Activity** class is a subclass of **Context**
- A class – the Activity that should be started

1. Set the Intent appropriately inside the **sendMessage(...)**
2. In order to get the **EditText** element use the **findViewById()**  
`EditText text = (EditText) findViewById(R.id.edit_message);`
3. Assign the text to a local message variable, use the help code below  
`text.getText().toString();`
4. Use the **putExtra()** method to add the text value to the intent from above. This method takes the key name as the first parameter and the value as the second parameter. Add the definition of the first argument:  

```
public final static String EXTRA_MSG =  
"com.mycompany.myfirstapp.MESSAGE";
```
5. Start the Activity.

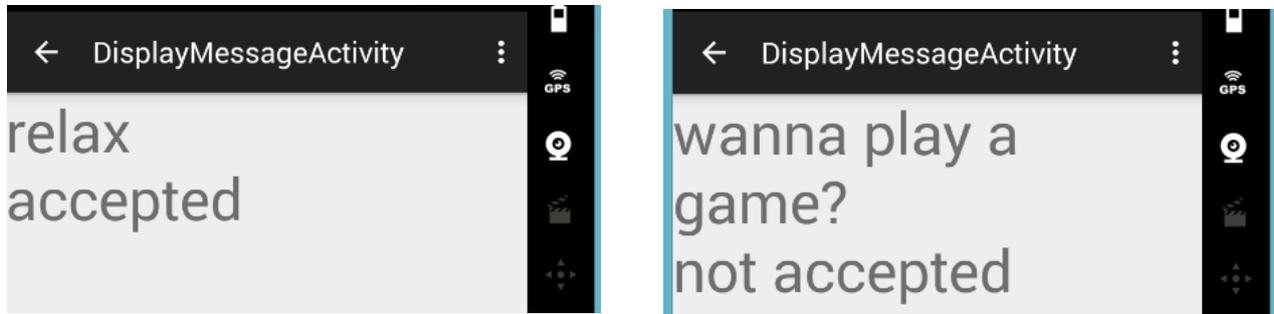
At this point by running the program and clicking on the button you will be able to see the new page, however the text written won't be the text from the text field. It will be "Hello World" that comes from the automatically generated code in the newly created Activity.

1. In the **DisplayMessageActivity.java** get the message from the intent by replacing `setContentView(R.layout.activity_display_message);` with **Intent intent = getIntent();**
2. Extract the message from the intent with the **getStringExtra()** method
3. Create the text view and set its size to 40 and its text to the String to which you extracted the message.
4. Finally, set the text view as the activity layout.



## Extra:

Use the checkbox that you already created and in the new Activity transfer the state of the checkbox like shown on the picture:



### 1.3 Getting started with java Programming, Eclipse

If you are not familiar with Java programming and Eclipse environment, then it is a good practice for you to do this assignment as well. Since you will program the server for your next lab using Java let's start with some basic stuff like creating a project and running our first **Hello World** app there.

**File -> New -> Java Project -> give your project a name -> click Finish**

You will notice in the **Package Explorer** that a new project was just created.

Extend your project, click on **src** and create a new Package (**right click on src -> New -> package**)

In the newly created package create a class (**right click on your package -> New -> class**)

So, you have just created a whole new project and you can start coding now.

Double click on the class. In the main window you will see an empty class.

Add the following code in your empty class:

```
public static void main(String[] args) {  
    System.out.println("Hello, World");  
}
```

Execute the following program and see the output. Do some more exercises and watch some tutorials if you haven't been programming in Java.

## References:

Intents - <http://developer.android.com/reference/android/content/Intent.html>