


Lab1a: Edit Texts

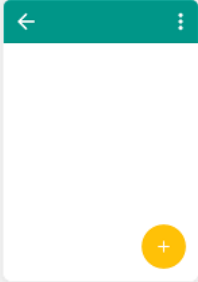
Lab Grading Rules

- Create your app from scratch. The package name should be
 - `your_name.lab_id.app_name`
- Demo your app to the TA
- Answer questions from the TA
 - These are simple questions testing your understanding. Should be easy if you have worked on it.
- Submit your program to catcourse
 - We will run plagiarism check on your program

naming of your app

Create New Project

 Configure Your Project



Basic Activity

Creates a new basic activity with the Navigation component

Name

Package name


Save location


Language

Java

Minimum SDK

API 24: Android 7.0 (Nougat)

 Your app will run on approximately **94.4%** of devices.
[Help me choose](#)

☐ Use legacy android.support libraries 

Previous

Next

Cancel

Finish

Lab 1.a Basic UI

- The app has a interface that contains a button and a text input field.
- Once the button is pushed, generate an intent that invokes a second activity
- The second activity will display the input string
- Reference: <https://developer.android.com/training/basics/firstapp/creating-project.html>

Build this app



Implement a Basic UI

- Open “app > res > layout > activity_main.xml”
- Select the text tab. We are working with the xml codes

Implement a Basic UI (2)

Delete everything and paste the following xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="horizontal">
</LinearLayout>
```

Understand Text Field

```
<EditText android:id="@+id/edit_message"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:hint="@string/edit_message" />
```

- android:id
- android:layout_width
- android:hint

Implement a Basic UI (3)

- Add a Text Field Element:

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout_width="match_parent"

android:layout_height="match_parent"

android:orientation="horizontal">

<EditText android:id="@+id/edit_message"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:hint="@string/edit_message" />

</LinearLayout>

Implement a Basic UI (4)

- Define the String field
- Go to : res/values/strings.xml
- Delete everything and paste:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
    <string name="app_name">My First App</string>
```

```
    <string name="edit_message">Enter a message</string>
```

```
    <string name="button_send">Send</string>
```

```
</resources>
```

Implement a Basic UI (5)

- Add a Button in activity_main.xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <EditText android:id="@+id/edit_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send" />
</LinearLayout>
```

Run the app



Play around UI

- Change the string field

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
  <string name="app_name">My First App</string>
```

```
  <string name="edit_message">Enter a message</string>
```

```
  <string name="button_send">Send</string>
```

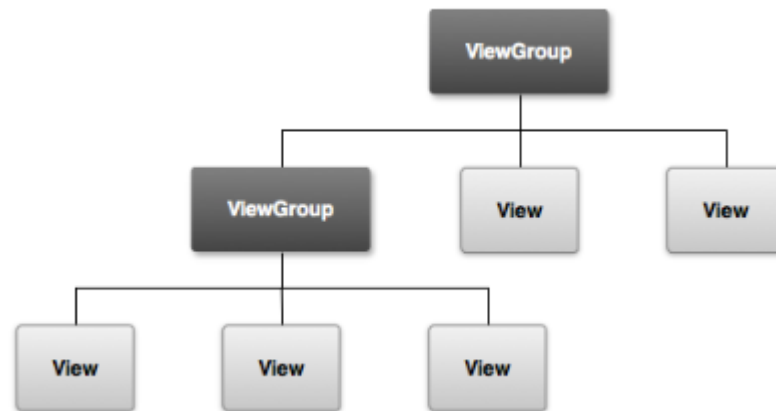
```
</resources>
```

- Change wi

```
<EditText android:id="@+id/edit_message"
  android:layout_weight="1"
  android:layout_width="0dp"
  android:layout_height="wrap_content"
  android:hint="@string/edit_message" />
```

Understand Basic UI Elements

- ViewGroup and View:
 - View is a basic element. ViewGroup is the organization of Views or ViewGroups
 - Text field, Button are views
 - LinearLayout defines a ViewGroup



Control the UI

- Goal: know how to push a button, generate a message, and generate a new screen.

Control the UI

- Add a field to the Button View

<Button

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="@string/button_send"

android:onClick="sendMessage" />



Control the UI (2)

- Go to `java > com.example.myfirstapp > MainActivity.java` , insert the following code:

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
    /** Called when the user clicks the Send button */  
    public void sendMessage(View view) {  
        // Do something in response to button  
    }  
}
```

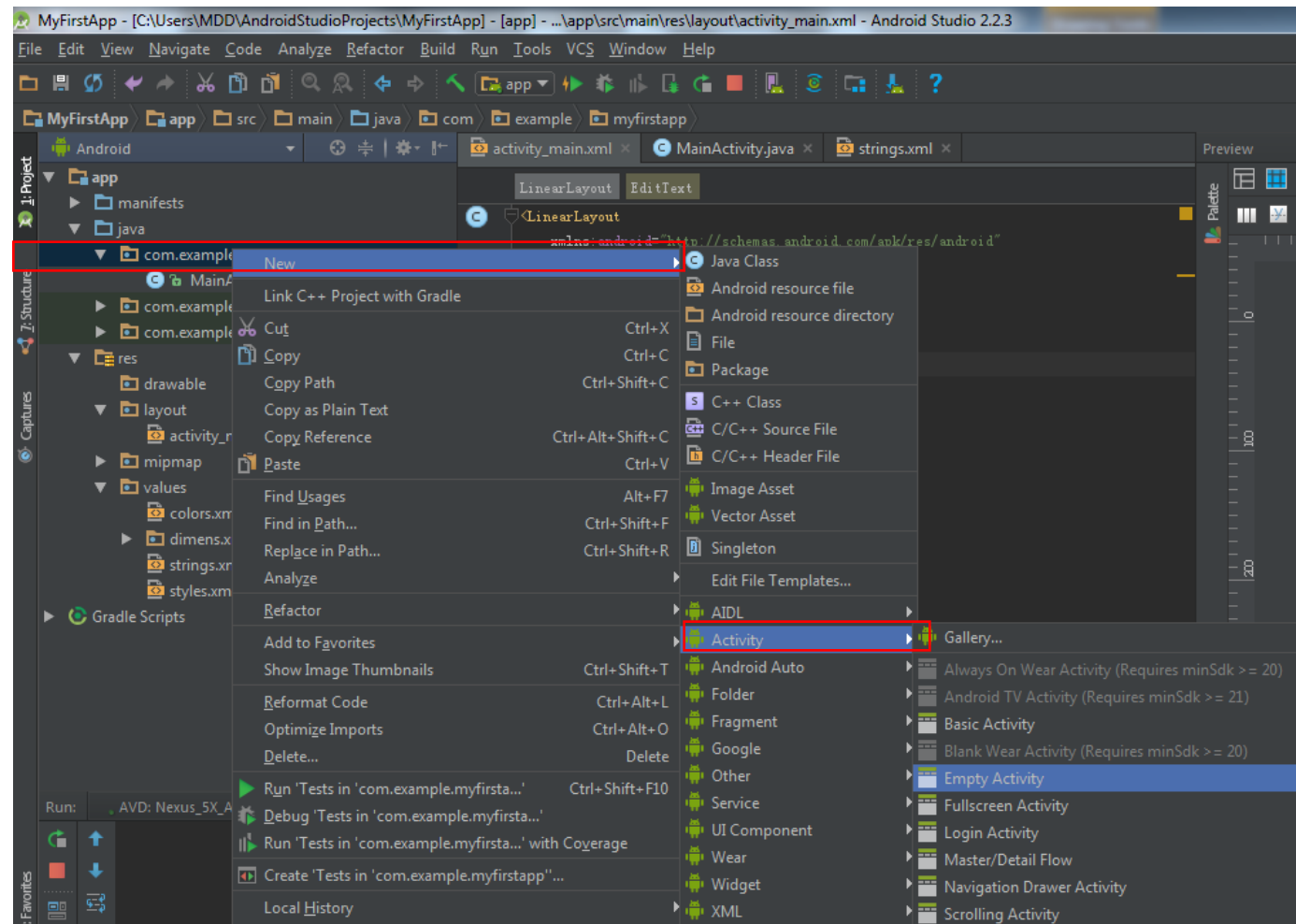
Control the UI (3)

- Create a Intent
 - Intent: an object that deliver message in run time between separate components, such as two activities.

```
public final static String EXTRA_MESSAGE = "com.example.myfirstapp.MESSAGE";  
public void sendMessage(View view) {  
    Intent intent = new Intent(this, DisplayMessageActivity.class);  
    EditText editText = (EditText) findViewById(R.id.edit_message);  
    String message = editText.getText().toString();  
    intent.putExtra(EXTRA_MESSAGE, message);  
    startActivity(intent);  
}
```

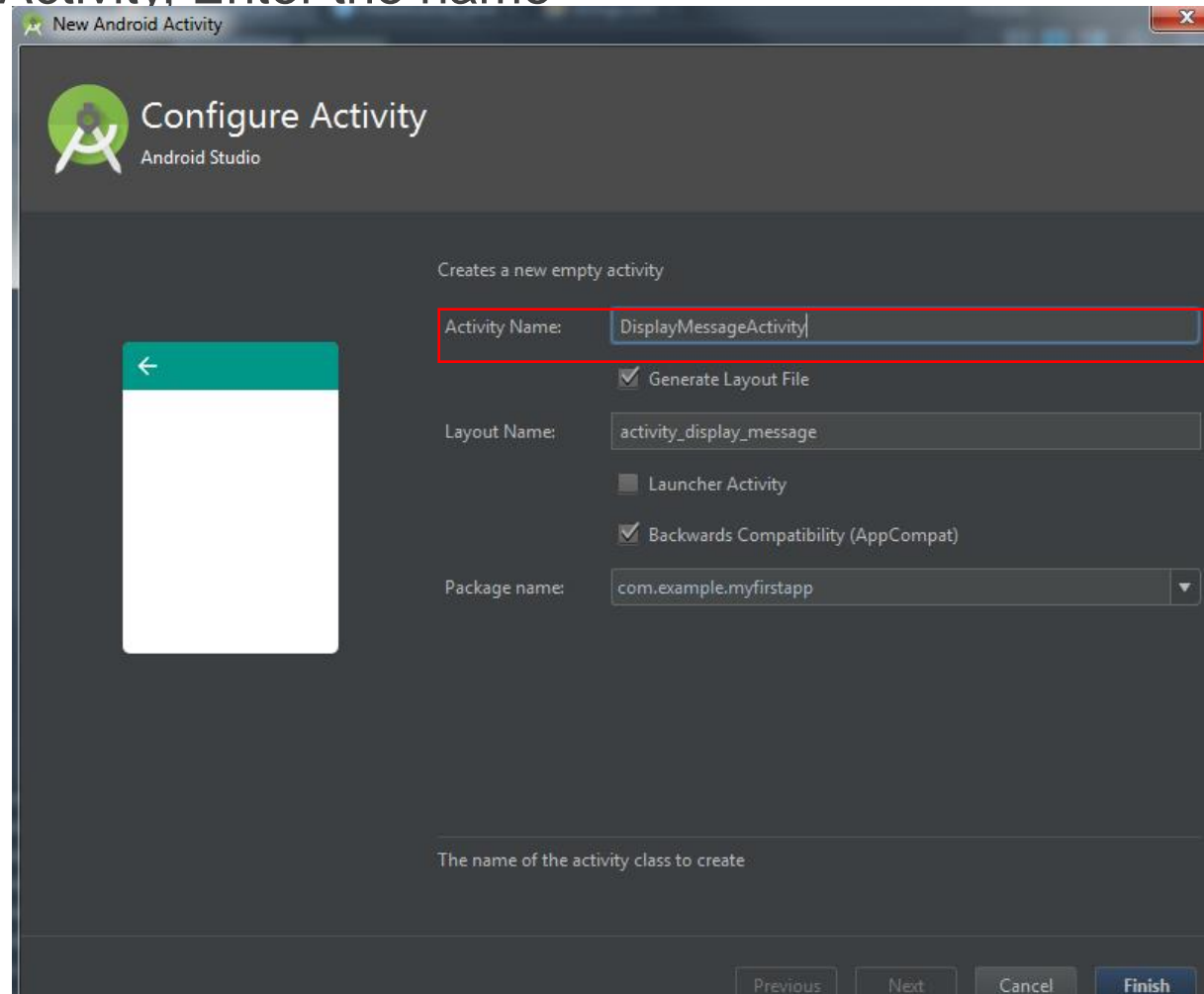
Control the UI (3)

- Create a new Activity



Control the UI (3)

- Create a new Activity, Enter the name



Control the UI (3)

In DisplayMessageActivity.java, implement the following:

```
public class DisplayMessageActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_message);

        Intent intent = getIntent();
        String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
        TextView textView = new TextView(this);
        textView.setTextSize(40);
        textView.setText(message);

        ViewGroup layout = (ViewGroup) findViewById(R.id.activity_display_message);
        layout.addView(textView);
    }
}
```

Under Manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.MyApplication"
        tools:targetApi="31">

        <!-- Main Activity definition with android:exported -->
        <activity
            android:name=".DisplayMessageActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>

</manifest>
```

Run the App

