

实验二

事件处理

【实验目的】

- 1.了解 Android 编程基础
- 2.熟悉 ImageView、Button、RadioButton 等基本控件，能够处理这些控件的基本事件
- 3.学会弹出基本的对话框，能够定制对话框中的内容，能对确定和取消按钮的事件做处理

【实验内容】

Lab2

中山大学学生信息系统



请输入学号

请输入密码

☒ 学生 ☐ 教职工

登录

注册

实现一个 Android 应用，界面呈现与实验一基本一致，要求：

- (1) ~~该界面为应用启动后看到的第一个界面~~
- (2) ~~输入学号和密码的控件要求用 `TextInputLayout` 实现~~

(3) 点击图片，弹出对话框如下图：



~~点击“拍摄”选项，弹出 Toast 信息“您选择了[拍摄]”；~~

~~点击“从相册选择”选项，弹出 Toast 信息“您选择了[从相册选择]”；~~

~~点击“取消”按钮，弹出 Toast 信息“您选择了[取消]”。~~

(4) ~~切换 RadioButton 的选项，弹出 Snackbar 提示“您选择了 xx”；~~

~~例如从选项“学生”切换到选项“教职工”，则提示“您选择了教职工”；~~



~~点击 Snackbar 上的“确定”按钮，则弹出 Toast 信息“Snackbar 的确定按钮被点击了”~~

(5) 点击登录按钮

~~依次判断学号是否为空，密码是否为空，用户名和密码是否正确（正确的学号和密码分别为“123456”，“6666”）；不正确则给出错误信息，如学号和密码都正确则提示“登陆成功”，如图：~~

The figure consists of four screenshots of a login form, arranged in a 2x2 grid, illustrating different states of the login process:

- Top Left:** The form is empty. The "学号" (Student ID) field is labeled "请输入学号" (Please enter student ID). The "密码" (Password) field is labeled "请输入密码" (Please enter password). Below the fields are two radio buttons: "学生" (Student) and "教职工" (Teacher/Staff), with "教职工" selected. At the bottom are "登录" (Login) and "注册" (Register) buttons. A red error message "学号不能为空" (Student ID cannot be empty) is displayed below the ID field.
- Top Right:** The "学号" field is filled with "1234567". The "密码" field is labeled "请输入密码". A red error message "密码不能为空" (Password cannot be empty) is displayed below the password field.
- Bottom Left:** The "学号" field is filled with "1234567". The "密码" field is filled with "....". A red error message "学号或密码错误" (Student ID or password error) is displayed at the bottom of the form, with a "确定" (Confirm) button next to it.
- Bottom Right:** The "学号" field is filled with "123456". The "密码" field is filled with "....". A green success message "登录成功" (Login successful) is displayed at the bottom of the form, with a "确定" (Confirm) button next to it.

(6) 点击注册按钮

~~如果切换选项时，RadioButton 选中的是“学生”，那么弹出 Snackbar 信息“学生注册功能尚未启用”，如果选中的是“教职工”，那么弹出 Toast 信息“教职工注册功能尚未启用”。~~



【基础知识】

1、在 java 文件中引用布局文件中的控件

在上一次实验中，在 `onCreateView(Bundle savedInstanceState)` 方法中调用 `setContentView()` 方法将布局加载进来。如果需要用到布局中的某些控件的话，首先需要给控件一个 `id`：

```
<ImageView
    android:id="@+id/image"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

定义 `id` 的语法和引用资源类似，`@+id/id` 名称，在同一个布局文件中不允许有重复的 `id`，即使是不同控件也不行，但是不同的布局文件中可以使用同一个 `id`。

之后在 `java` 文件中将布局加载之后，也就是 `setContentView()` 之后，使用 `findViewById()` 方法可以获得该控件：

```
mImage = (ImageView) findViewById(R.id.image);
```

`findViewById()` 方法带一个参数，就是刚刚定义的那个 `id`，参数形式为 `R.id.XXX`，其中 `XXX` 就是刚刚定义的那个 `id`，由于 `findViewById()` 方法返回的是一个 `View` 类型，所以需要强制类型转换为 `ImageView` 类型。

获得这个 `ImageView` 之后，就可以对这个 `Button` 进行后续的操作了。

2、弹出 Toast 信息

在需要弹出 `Toast` 信息的地方，写上：

```
Toast.makeText(MainActivity.this, "您选择了[取消]", Toast.LENGTH_SHORT).show();
```

这个方法带三个参数，`context`，`text`，`duration`。`context` 是一个上下文类型，写上使用这个方法的 java 类名加上`.this` 即可，`text` 是 Toast 要显示的信息，`duration` 是 Toast 显示的时间长度，有 `Toast.LENGTH_SHORT` 和 `Toast.LENGTH_LONG` 可选，最后记得调用 `show()` 方法将 Toast 显示出来。

3、事件处理

此次实验用到的事件处理方法一共三个，一个是 `ImageView` 的点击事件，

`mImage.setOnClickListener()`，一个是 `Button` 的单击事件，`mButton.setOnClickListener()`，

一个是 `RadioGroup` 的切换事件，`mRadioGroup.setOnCheckedChangeListener()`，下面以 `Button` 为例简单介绍一下事件处理。

要处理事件，首先要将与该事件有关的控件引入进来，比如要判断用户名是否为空，那么除了 `Button` 需要引入外（因为 `Button` 是触发事件的主体），还需要引入用户名的输入框。引入之后，`button.setOnClickListener()` 中做处理，这里的 `mLogin` 是一个变量名，

```
mLogin.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {
```

`onClick` 方法带的 `view` 参数是触发这个单击事件的控件的引用，在这个例子中，`view` 就是触发事件的 `Button`，在 `onClick()` 方法中做事件的处理，例如判断学号是否为空：

```
String number = mNumberEdit.getText().toString();  
String password = mPassEdit.getText().toString();  
if (TextUtils.isEmpty(number)) {
```

关于 `EditText`，`Button`，`RadioGroup` 各自有的一些方法建议查看 API 文档，不在这里一一说明了。

4、简单对话框的使用

Android 中最基本的对话框是 `AlertDialog`，之所以说它最简单，是因为布局和使用的方法都很简单，布局是写好的，标题，通过 `setTitle()` 方法设置，图标，通过 `setIcon()` 方法设置，显示在中间的主要信息，通过 `setMessage()` 方法显示，等等。显示一个 `AlertDialog` 的基本步骤如下：

- 1、创建 `AlertDialog.Builder` 对象
- 2、调用上面的方法进行设置
- 3、如果需要设置取消按钮，方法原型如下：

```
mBuilder.setNegativeButton("取消", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialogInterface, int i) {
```

确定按钮怎么设置自行查阅资料

- 4、调用 `AlertDialog.Builder` 的 `create()` 方法创建 `AlertDialog` 对象，再调用 `AlertDialog` 对象的 `show()` 方法将对话框显示出来。

【拓展知识】

1、Snackbar

Google 在 2015 年的 IO 大会上发布了较之前更为详细的 Material Design 规范和全新的 Android Design Support Library，Snackbar 就是其中的一个控件，可以用来代替 Toast 信息，除了用户体验更好以外，功能也更强大。先来看一下语法：

```
Snackbar.make(mRootView, message, Snackbar.LENGTH_SHORT)  
    .setAction("Undo", new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {}  
    })  
    .setActionTextColor(getResources().getColor(R.color.colorPrimary))  
    .setDuration(5000)  
    .show();
```

make 方法中的三个参数分别是根布局、要显示的消息和时长，这个方法是必须的，之后的方法都不是必须的。定义完之后调用 show()方法就可以显示 Snackbar 了。

setAction 方法使该 Snackbar 右侧按钮可以被点击并处理一些事件，两个参数分别是按钮显示的文本以及处理点击事件的逻辑，形式与 Button 一样。

setActionTextColor()是设置右侧按钮文本的颜色，setDuration()方法可以定义 Snackbar 弹出时间长度，单位为 ms，当右侧按钮被点击之后，Snackbar 会立刻消失。

静态的 Snackbar 如下图：



在 api 为 19 的模拟器下，按钮显示不出来，可能是不支持，换了 api 22 的模拟器就好了。

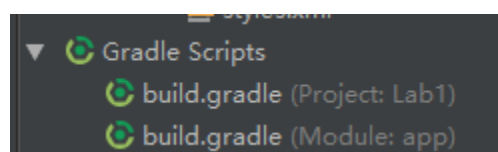
2、TextInputLayout

TextInputLayout 也是 Android Design Support Library 中的一个控件，用于接受用户输入，与 EditText 配合使用有更好的用户体验，TextInputLayout 集合了输入提示，报错等功能，并且自带动画效果。下面来看一下怎么实现：

首先是 xml 布局：



一个 TextInputLayout 中只能包含一个 EditText，如果想要在预览中看到 TextInputLayout 布局的话，需要在修改对应的 build.gradle 文件中的配置：




```

apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    buildToolsVersion "26.0.1"
    defaultConfig {
        applicationId "com.example.small.lab1"
        minSdkVersion 15
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:26.0.0-alpha1'
    testCompile 'junit:junit:4.12'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    compile 'com.android.support:design:26.0.0-alpha1'
}

```

在 dependencies 中加上 compile 'com.android.support:design:26.0.0-alpha1'，在修改了 build.gradle 文件之后，也可以点击 Sync Now 同步项目。

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly. [Sync Now](#)

```
1 apply plugin: 'com.android.application'
```

然后在 java 文件中引入 TextInputLayout，通过 TextInputLayout 获取 EditText 中的内容：

```

mNumberText = (TextInputLayout) findViewById(R.id.more_number);
mPassText = (TextInputLayout) findViewById(R.id.more_password);
mNumberEdit = mNumberText.getEditText();
mPassEdit = mPassText.getEditText();

```

TextInputLayout 有特色的一点是可以显示提示信息：

```
mNumberText.setErrorEnabled(true);  
mNumberText.setError(emptyNumber);
```

首先允许使用提示信息，然后显示提示信息，如果想使提示信息消失，设置 setErrorEnabled(false)即可。

其它方法的话自行查阅 API 文档。

【检查内容】

- 1、 布局显示是否正常
- 2、 点击图片时弹出的对话框是否正常（显示内容及按钮点击的反应）
- 3、 在学号为空时提示是否正常，在密码为空时提示是否正常，点击登录按钮后，学号和密码正确时弹出提示是否符合要求
- 4、 RadioButton 切换时的反应是否正常
- 5、 点击注册按钮后事件处理是否符合要求

【提交说明】

- 1、 deadline：下一次实验课前一天晚上 12 点
- 2、 提交作业地址：<ftp://edin.sysu.edu.cn>
- 3、 文件命名及格式要求：学号_姓名_labX.zip（姓名中文拼音均可）
- 4、 目录结构：

```
15331111_huashen_lab1 --  
|  
-- lab1实验报告.pdf  
|  
-- lab1_code (包含项目代码文件)
```

其中项目代码文件为项目文件夹，*提交之前先 clean