

实验二

事件处理

【实验目的】

1. 了解 Android 编程基础
2. 熟悉 Button、RadioButton、EditText 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】



实现一个 Android 应用，界面呈现如下效果（界面设计上的要求与实验一相同）：

要求：

- （1）该界面为应用启动后看到的第一个界面

(2) 点击登录按钮：

如果用户名为空，弹出 Toast 信息“用户名不能为空”；

如果密码为空，弹出 Toast 信息“密码不能为空”；

如果用户名输入为“Android”，密码为“123456”，弹出如下对话框：



点击“确定”，弹出 Toast 信息——对话框“确定”按钮被点击。，点击“取消”，弹出 Toast 信息——对话框“取消”按钮被点击。

否则，弹出如下对话框：



确定和取消按钮点击后效果参见上一个对话框

点击注册按钮：

如果 `RadioButton` 选中的是“学生”，那么弹出 `Toast` 信息“学生身份注册功能尚未开启”，如果选中的是“教师”，那么弹出 `Toast` 信息“教师身份注册功能尚未开启”，以此类推。

`RadioButton` 选择项切换：选择项切换之后，弹出 `Toast` 信息“XX 身份被选中”，例如从学生切换到教师，弹出 `Toast` 信息“教师身份被选中”

【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

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="Button"/>
```

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：

```
Button button = (Button) findViewById(R.id.button);
```

`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 Button 类型。

获得这个 Button 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, "一个Toast", 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两个，一个是 Button 的单击事件，`button.setOnClickListener()`，一个是 RadioGroup 的切换事件，`radioGroup.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 `button` 是一个变量名，记得换成自己定义的变量名。

```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        // 在这里进行事件的处理  
    }  
});
```

`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，例如判断用户名是否为空：

```
Button button = (Button) findViewById(R.id.button);  
final EditText username = (EditText) findViewById(R.id.username);  
  
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if (TextUtils.isEmpty(username.getText().toString())) {  
            Toast.makeText(MainActivity.this, "用户名为空", Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```

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：

```
builder.setNegativeButton("取消", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialogInterface, int i) {  
        // 点击按钮后的事件处理方法写在这里  
    }  
});
```

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 I/O 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(rootView, 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，需要注意的是，当右侧按钮被点击之后，Snackbar 会立刻消失。

文档放不了动图，放一下静态的 Snackbar，实验材料中的 apk 安装启动后，点击拓展启动的界面用的都是 Snackbar，可以看一下动态效果。



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，先放几张图：





可以看到，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：

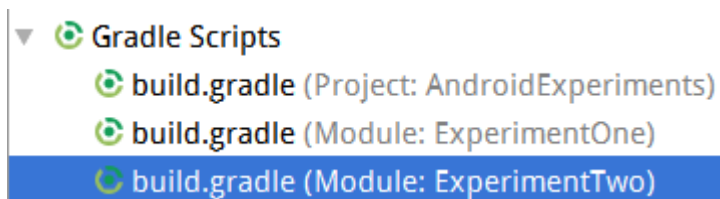
首先是 xml 布局：

```
<android.support.design.widget.TextInputLayout
    android:id="@+id/more_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp">

    <EditText
        style="@style/my_more_edittext_style"
        android:hint="请输入用户名" />

</android.support.design.widget.TextInputLayout>
```

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 24
    buildToolsVersion "24.0.0"

    defaultConfig {
        applicationId "com.study.android.kehan.experimenttwo"
        minSdkVersion 19
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:24.2.0'
    compile 'com.android.support:design:24.0.0'
}

```

在 dependencies 中加上 compile 'com.android.support:design:24.0.0'，然后再同步项目，就可以显示了：

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

在修改了 build.gradle 文件之后，也可以点击 Sync Now 同步项目。

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

```

usernameText = (TextInputLayout) findViewById(R.id.more_username);
passwordText = (TextInputLayout) findViewById(R.id.more_password);
usernameEdit = usernameText.getText();
passwordEdit = passwordText.getText();

```

TextInputLayout 有特色的一点是可以显示提示信息：

```
usernameText.setErrorEnabled(true);
usernameText.setError(emptyUsername);
```

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（与上次实验要求相同）
- 2、在用户名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，用户名和密码正确时弹出框是否符合要求（显示内容及按钮点击的反应），用户名或密码错误时弹出框是否符合要求（显示内容及按钮点击的反应）。
- 3、RadioButton 切换时的反应是否正常
- 4、点击注册按钮后事件处理是否符合要求
- 5、（拓展）使用 TextInputLayout 和 Snackbar，具体对于各种事件的处理安装压缩包中的 apk 后，点击拓展按钮可以看到。

【提交说明】

- 1、deadline：下一次实验课前一天晚上 12 点
- 2、提交 ftp：<ftp://222.200.185.18:1890/>，作业提交文件夹中对应的文件夹下
- 3、命名与目录结构要求：

附件命名及格式要求：学号_姓名_labX.zip（姓名中文拼音均可）

重复提交命名格式要求：学号_姓名_labX_Vn.zip，主题：学号_姓名_labX_Vn

目录结构：

```
14331111_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