



《移动终端开发技术》 电子教案

第四单元 广播的最佳实践

所属专业（教研室）： 计算机软件技术

制定人： 陈媛媛

合作人：

制定时间： 2018年2月

日照职业技术学院



单元标题	广播的最佳实践	单元教学学时	4 课时
		在整体设计中的位置	第 19 次
授课班级		上课地点	一体化教室
上课时间	周 月 日第 节		
教学目标	能力目标	知识目标	素质目标
	能够熟练设计并实现用户登录功能，能够发送和接收系统广播、自定义广播及本地广播。	1、掌握用户登录界面设计的方法； 2、掌握用户登录功能的实现； 3、掌握发送和接收自定义广播的方法； 4、掌握弹出对话框的方法。	1、养成积极主动学习意识； 2、养成勤于动手的习惯。
教学重点、难点	教学重点：设计并实现用户登录功能 教学难点：发送和接收自定义广播的方法		
教学方法	采用反转课堂教学模式，课前学生学习微课了解知识点，课上采用教师引导、演示，学生分组练习、讨论等教学方法。 运用多媒体、AndroidStudio 开发环境、实训助手、教学平台等辅助授课。		
课前需掌握的知识点	1、发送和接收广播； 2、Activity 的生命周期； 3、LinearLayout 布局的使用。		
教学任务分解	任务一、设计思路及准备工作 任务二、登录界面的设计 任务三、登录功能的实现 任务四、强制下线功能的实现		
教学总结			



一、情景导入

强制下线功能算是比较常见的，很多的应用程序都具备这个功能，比如 QQ 号在别处登录，就会将你强制挤下线。其实实现强制下线功能的思路比较简单，只需要在界面上弹出一个对话框，让用户无法进行任何操作，必须要点击对话框中的确定按钮，然后回到登录界面即可。

二、复习与提问

如何发送和接收本地广播，请小组讨论并上台演示。

三、知识讲解

任务一、设计思路及准备工作

强制下线功能算是比较常见的，很多的应用程序都具备这个功能，比如 QQ 号在别处登录，就会将你强制挤下线。其实实现强制下线功能的思路比较简单，只需要在界面上弹出一个对话框，让用户无法进行任何操作，必须要点击对话框中的确定按钮，然后回到登录界面即可。

准备工作：

(1) 创建 ActivityCollector 类用于管理所有活动。

```
public class ActivityCollector {
    public static List<Activity> activities=new ArrayList<>();
    public static void addActivity(Activity activity){
        activities.add(activity);
    }
    public static void removeActivity(Activity activity){
        activities.remove(activity);
    }
    public static void finishAll(){
        for(Activity activity:activities){
            if(!activity.isFinishing()){
                activity.finish();
            }
        }
        activities.clear();
    }
}
```

(2) 创建 BaseActivity 类作为所有活动的父类。

```
public class BaseActivity extends AppCompatActivity {
    private ForceOffLineReceiver receiver;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ActivityCollector.addActivity(this);
    }
}
```



```
@Override
protected void onDestroy() {
    super.onDestroy();
    ActivityCollector.removeActivity(this);
}
}
```

任务二、登录界面的设计

新建 LoginActivity，生成布局文件

activity_login.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:orientation="horizontal">
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:textSize="18sp"
    android:text="Account:" />
<EditText
    android:id="@+id/account"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_gravity="center_vertical"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:orientation="horizontal">
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:textSize="18sp"
    android:text="Password:" />
<EditText
```



```
        android:id="@+id/password"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_gravity="center_vertical"
        android:inputType="textPassword"/>
    </LinearLayout>
    <Button
        android:id="@+id/login"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:text="Login"/>
    </LinearLayout>
```

任务三、登录功能的实现

LoginActivity.java

```
public class LoginActivity extends BaseActivity {
    private EditText accountEdit;
    private EditText passwordEdit;
    private Button login;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        accountEdit=(EditText) findViewById(R.id.account);
        passwordEdit=(EditText) findViewById(R.id.password);
        login=(Button) findViewById(R.id.login);
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String account=accountEdit.getText().toString();
                String password=passwordEdit.getText().toString();
                //如果账号是 admin, 密码是 123456, 则登录成功, 跳转到 MainActivity
                if (account.equals("admin")&&password.equals("123456")) {
                    Intent intent=new Intent(LoginActivity.this,MainActivity.class);
                    startActivity(intent);
                    finish();
                }
                else {
                    Toast.makeText(LoginActivity.this,"account or password is
invalid",Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
```



```
}  
}
```

任务四、实现强制下线功能

广播接收器需要弹出一个对话框来阻塞用户的正常操作，如果创建的使一个静态注册的广播接收器，是没有办法在 `onReceive()` 方法里弹出对话框这样的 UI 控件的，我们也不可能在每个活动中都去注册一个动态的广播接收器。

思路：自需要在 `BaseActivity` 中动态注册一个广播接收器就可以。

只有在栈顶的活动才能收到强制下线的广播，所以需要重写 `onResume()` 方法，失去栈顶的位置自动取消广播接收器的注册，所以需要重写 `onPause()` 方法。

MainActivity

activity_main.xml

```
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/force_offline"  
    android:text="Send force offline broadcast"/>
```

MainActivity.java

```
public class MainActivity extends BaseActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Button forceOffline=(Button)findViewById(R.id.force_offline);  
        forceOffline.setOnClickListener(new View.OnClickListener() {  
            @Override  
            //点击按钮发送一条广播，广播的值为  
com.example.cyy.broadcastbestpractice.FORCE_OFFLINE  
            public void onClick(View view) {  
                Intent intent=new  
                Intent("com.example.cyy.broadcastbestpractice.FORCE_OFFLINE");  
                sendBroadcast(intent);  
            }  
        });  
    }  
}
```

在 `BaseActivity` 注册广播接收器

```
public class BaseActivity extends AppCompatActivity {  
    private ForceOfflineReceiver receiver;  
    @Override  
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        ActivityCollector.addActivity(this);  
    }  
}
```



```
}
@Override
protected void onDestroy() {
    super.onDestroy();
    ActivityCollector.removeActivity(this);
}
@Override
//只有在栈顶的活动才能收到强制下线的广播
protected void onResume() {
    super.onResume();
    IntentFilter intentFilter=new IntentFilter();

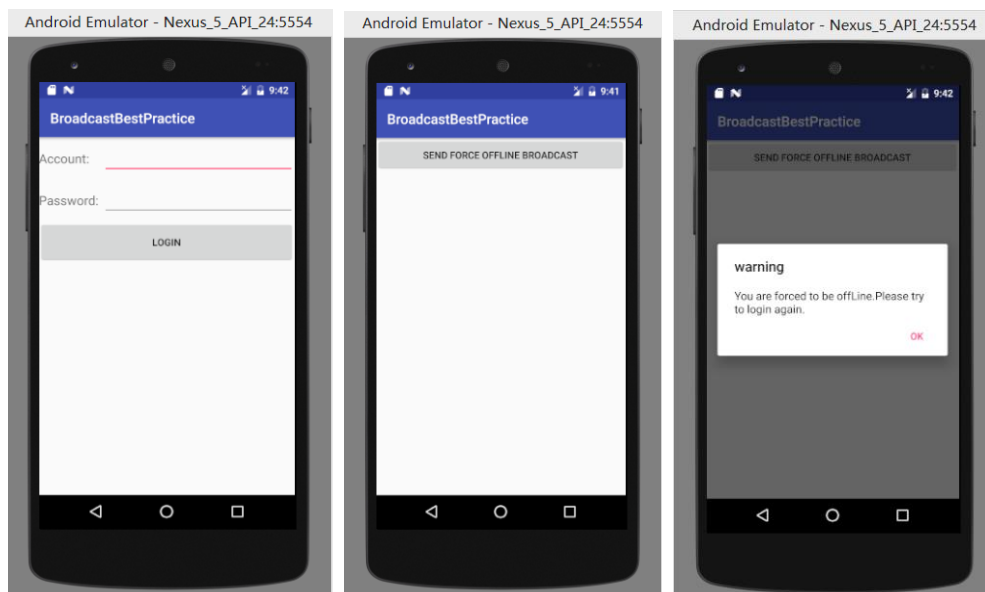
intentFilter.addAction("com.example.cyy.broadcastbestpractice.FORCE_OFFLINE");
    receiver=new ForceOffLineReceiver();
    registerReceiver(receiver, intentFilter);
}
@Override
//失去栈顶位置自动取消广播接收器的注册
protected void onPause() {
    super.onPause();
    if(receiver!=null){
        unregisterReceiver(receiver);
        receiver=null;
    }
}
class ForceOffLineReceiver extends BroadcastReceiver{

    @Override
    public void onReceive(final Context context, Intent intent) {
        AlertDialog.Builder builder=new AlertDialog.Builder(context);
        builder.setTitle("warning");
        builder.setMessage("You are forced to be offLine.Please try to login
again.");
        builder.setCancelable(false);
        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                ActivityCollector.finishAll();//销毁所有活动
                Intent intent=new Intent(context, LoginActivity.class);
                context.startActivity(intent);//重新启动 LoginActivity
            }
        });
        builder.show();
    }
}
```

```

    }
}
}
修改主活动 AndroidManifest
<activity android:name=".MainActivity">
</activity>
<activity android:name=".LoginActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

```



登录界面

主界面

强制下线

四、演示作品

各小组演示汇报作品，教师与其他小组总结评价作品，各小组课后完善作品并提交到教学平台。

五、知识巩固

总结知识点，使用教学平台中的随堂练习题巩固本所学知识。

拓展
作业

优化强制下线功能

教学
后记