Demo24-广播实现强制下线功能

1. 导入工具类：Activity集中管理类---ActivityCollector.java

import android.app.Activity;

import java.util.ArrayList;

import java.util.List;

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();

}

}

}

}

1. Activity给一个父类，方便统一管理
2. 嵌套一个广播接收器的结果处理类（内部类）
3. 动态注册广播接收器，在onResume那里注册才是对的，防止前面的多个Activity也跳转，没必要，只要当前运行的Activity跳转即可

import android.app.AlertDialog;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.IntentFilter;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

public class BaseActivity extends AppCompatActivity {

private ForceOfflineReceiver receiver;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

ActivityCollector.addActivity(this);

}

@Override

protected void onResume() {

super.onResume();

IntentFilter intentFilter = new IntentFilter();

intentFilter.addAction("com.example.broadcastbestpractice.FORCE\_OFFLINE");

receiver = new ForceOfflineReceiver();

registerReceiver(receiver, intentFilter);

}

@Override

protected void onPause() {

super.onPause();

if (receiver != null) {

unregisterReceiver(receiver);

receiver = null;

}

}

@Override

protected void onDestroy() {

super.onDestroy();

ActivityCollector.removeActivity(this);

}

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 dialog, int which) {

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();

}

}

}

1. 写一个登录的LoginActivity, 布局和逻辑都很简单

activity\_login.xml

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

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

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<LinearLayout

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="60dp">

<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:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="60dp">

<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>

LoginActivity.java

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

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 v) {

String account = accountEdit.getText().toString();

String password = passwordEdit.getText().toString();

// 如果账号是admin且密码是123456，就认为登录成功

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\_SHORT).show();

}

}

});

} }

1. MainActivity发送广播，触发广播接收器的onReceive方法

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

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

public void onClick(View v) {

Intent intent = new Intent("com.example.broadcastbestpractice.FORCE\_OFFLINE");

sendBroadcast(intent);

}

});

}

}

activity\_main.xml

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

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

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<Button

android:id="@+id/force\_offline"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Send force offline broadcast" />

</LinearLayout>

 