第2周实验

欢迎界面需要获取服务器中程序的版本号，实现

【版本号比对】

🡪【弹出更新提醒对话框】

🡪【弹出下载APK进度条】

🡪【替换安装程序】。

由于这个逻辑是一个整体，因此将其抽取出来放在VersionUpdateUtils类中。

private static final int MESSAGE\_NET\_EEOR = 101;

private static final int MESSAGE\_IO\_EEOR = 102;

private static final int MESSAGE\_JSON\_EEOR = 103;

private static final int MESSAGE\_SHOEW\_DIALOG = 104;

protected static final int MESSAGE\_ENTERHOME = 105;

## 1.本地版本号

private String mVersion;

private Activity context;

private ProgressDialog mProgressDialog;

private VersionEntity versionEntity;

public VersionUpdateUtils(String Version, Activity activity) {

mVersion = Version;

context = activity;

}

## 2.获取服务器版本号

public void getCloudVersion() {

try {

HttpClient client = new DefaultHttpClient();

/\*连接超时\*/

HttpConnectionParams.setConnectionTimeout(client.getParams(), 5000);

/\*请求超时\*/

HttpConnectionParams.setSoTimeout(client.getParams(), 5000);

HttpGet httpGet = new HttpGet(

"http://192.168.100.9:8080/updateinfo.html");

HttpResponse execute = client.execute(httpGet);

if (execute.getStatusLine().getStatusCode() == 200) {

// 请求和响应都成功了

HttpEntity entity = execute.getEntity();

String result = EntityUtils.toString(entity, "gbk");

// 创建jsonObject对象

JSONObject jsonObject = new JSONObject(result);

versionEntity = new VersionEntity();

String code = jsonObject.getString("code");

versionEntity.versioncode = code;

String des = jsonObject.getString("des");

versionEntity.description = des;

String apkurl = jsonObject.getString("apkurl");

versionEntity.apkurl = apkurl;

if (!mVersion.equals(versionEntity.versioncode)) {

// 版本号不一致

handler.sendEmptyMessage(MESSAGE\_SHOEW\_DIALOG);

}

}

## 3.弹出更新提示对话框

private void showUpdateDialog(final VersionEntity versionEntity) {

// 创建dialog

AlertDialog.Builder builder = new Builder(context);

builder.setTitle("检查到新版本：" + versionEntity.versioncode);// 设置标题

builder.setMessage(versionEntity.description);// 根据服务器返回描述,设置升级描述信息

builder.setCancelable(false);// 设置不能点击手机返回按钮隐藏对话框

builder.setIcon(R.drawable.ic\_launcher);// 设置对话框图标

// 设置立即升级按钮点击事件

builder.setPositiveButton("立即升级",

new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

initProgressDialog();

downloadNewApk(versionEntity.apkurl);

}

});

// 设置暂不升级按钮点击事件

builder.setNegativeButton("暂不升级",

new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.dismiss();

enterHome();

}

});

// 对话框必须调用show方法 否则不显示

builder.show();

}

## 4.初始化进度条对话框

private void initProgressDialog() {

mProgressDialog = new ProgressDialog(context);

mProgressDialog.setMessage("准备下载...");

mProgressDialog.setProgressStyle(ProgressDialog.STYLE\_HORIZONTAL);

mProgressDialog.show();

}

## 5.下载新版本

protected void downloadNewApk(String apkurl) {

DownLoadUtils downLoadUtils = new DownLoadUtils();

downLoadUtils.downapk(apkurl, "/mnt/sdcard/mobilesafe2.0.apk", new MyCallBack() {

@Override

public void onSuccess(ResponseInfo<File> arg0) {

// TODO Auto-generated method stub

mProgressDialog.dismiss();

MyUtils.installApk(context);

}

@Override

public void onLoadding(long total, long current, boolean isUploading) {

// TODO Auto-generated method stub

mProgressDialog.setMax((int) total);

mProgressDialog.setMessage("正在下载...");

mProgressDialog.setProgress((int) current);

}

@Override

public void onFailure(HttpException arg0, String arg1) {

// TODO Auto-generated method stub

mProgressDialog.setMessage("下载失败");

mProgressDialog.dismiss();

enterHome();

}

});

}

private void enterHome() {

handler.sendEmptyMessageDelayed(MESSAGE\_ENTERHOME, 2000);

}

}

## 1.2.6 版本信息的实体类

public class VersionEntity {

/\*\*服务器版本号\*/

public String versioncode;

/\*\*版本描述\*/

public String description;

/\*\*apk下载地址\*/

public String apkurl;

}

## 1.2.7 欢迎界面逻辑

public class SplashActivity extends Activity {

/\*\* \* 应用版本号 \*/

private TextView mVersionTV;

/\*\* \* 本地版本号 \*/

private String mVersion;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// 设置没有标题栏 在加载布局之前调用

requestWindowFeature(Window.FEATURE\_NO\_TITLE);

setContentView(R.layout.activity\_splash);

mVersion = MyUtils.getVersion(getApplicationContext());

initView();

final VersionUpdateUtils updateUtils = new VersionUpdateUtils(mVersion,

SplashActivity.this);

new Thread() {

public void run() {

// 获取服务器版本号

// 获取服务器版本号

updateUtils.getCloudVersion();

} ;

}.start();

}

/\*\* 初始化控件\*/

private void initView() {

mVersionTV = (TextView) findViewById(R.id.tv\_splash\_version);

mVersionTV.setText("版本号 " + mVersion);

}

}

请求网络下载APK到手机内存，需要添加网络权限和SD卡的权限

<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

# 1.3 主界面

## 1.3.1 主界面UI

Activit\_home.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:background="@drawable/bg\_home"  
 android:orientation="vertical" >  
  
 <LinearLayout  
 android:id="@+id/ll"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:gravity="center"  
 android:orientation="horizontal" >  
  
 <ImageView  
 android:layout\_width="100dp"  
 android:layout\_height="107.5dp"  
 android:background="@drawable/superman" />  
  
 <TextView  
 android:id="@+id/tv\_ad"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:paddingTop="30dp"  
 android:text="主银，我是您的手机小护卫"  
 android:textColor="#90000000"  
 android:textScaleX="1.1"  
 android:textSize="16sp"  
 android:typeface="normal" />  
 </LinearLayout>  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <GridView  
 android:id="@+id/gv\_home"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:cacheColorHint="#00000000"  
 android:gravity="center"  
 android:horizontalSpacing="10dp"  
 android:numColumns="3"  
 android:verticalSpacing="10dp" >  
 </GridView>  
 </RelativeLayout>  
  
</LinearLayout>

<ImageView  
 android:layout\_width="100dp"  
 android:layout\_height="107.5dp"  
 android:background="@drawable/superman" />  
  
 <TextView  
 android:id="@+id/tv\_ad"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:paddingTop="30dp"  
 android:text="主银，我是您的手机小护卫"  
 android:textColor="#90000000"  
 android:textScaleX="1.1"  
 android:textSize="16sp"  
 android:typeface="normal" />  
 </LinearLayout>  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <GridView  
 android:id="@+id/gv\_home"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:cacheColorHint="#00000000"  
 android:gravity="center"  
 android:horizontalSpacing="10dp"  
 android:numColumns="3"  
 android:verticalSpacing="10dp" >  
 </GridView>  
 </RelativeLayout>  
  
</LinearLayout>

### 1.3.2 主界面Item布局

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:orientation="vertical" >  
 <ImageView  
 android:id="@+id/iv\_icon"  
 android:layout\_width="80dp"  
 android:layout\_height="80dp"  
 android:background="@drawable/atools" />  
 <TextView  
 android:id="@+id/tv\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="高级工具"  
 android:layout\_marginTop="3dp"  
 android:textColor="#80000000"  
 android:textSize="18sp" />  
</LinearLayout>

## 1.3.3 数据适配器 HomeAdapter

public class HomeAdapter extends BaseAdapter {  
  
 int[] imageId = {R.drawable.safe,  
 R.drawable.callmsgsafe, R.drawable.app, R.drawable.trojan, R.drawable.sysoptimize, R.drawable.taskmanager,  
 R.drawable.netmanager, R.drawable.atools, R.drawable.settings};  
 String[] names = {"手机防盗", "通讯卫士", "软件管家", "手机杀毒", "缓存清理", "进程管理",  
 "流量统计", "高级工具", "设置中心"};  
 private Context context;  
  
 public HomeAdapter(Context context) {  
 this.context = context;  
 }  
 // 设置gridView一个多少个条目  
 @Override  
 public int getCount() {  
 return 9;  
 }  
 // 设置每个条目的界面  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 View view = View.inflate(context,  
 R.layout.item\_home, null);  
 ImageView iv\_icon = (ImageView) view.findViewById(R.id.iv\_icon);  
 TextView tv\_name = (TextView) view.findViewById(R.id.tv\_name);  
 iv\_icon.setImageResource(imageId[position]);  
 tv\_name.setText(names[position]);  
 return view;  
 }  
 // 后面两个方法暂时不需要设置  
 @Override  
 public Object getItem(int position) {  
 return null;  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return 0;  
 }  
}

## 1.3.4 主界面逻辑 HomeActivity

public class HomeActivity extends Activity {

/\*\*

\* 声明GridView 该控件类似ListView

\*/

private GridView gv\_home;

/\*\*

\* 申请权限

\*/

private long mExitTime;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// 初始化布局

requestWindowFeature(Window.FEATURE\_NO\_TITLE);

setContentView(R.layout.activity\_home);

msharedPreferences = getSharedPreferences("config", MODE\_PRIVATE);

// 初始化GridView

gv\_home = (GridView) findViewById(R.id.gv\_home);

gv\_home.setAdapter(new HomeAdapter(HomeActivity.this));

// 设置条目的点击事件

gv\_home.setOnItemClickListener(new OnItemClickListener() {

// parent代表gridView,view代表每个条目的view对象,postion代表每个条目的位置

@Override

public void onItemClick(AdapterView<?> parent, View view,

int position, long id) {

switch (position) {

case 0: // 点击手机防盗

if (isSetUpPassword()) {

// 弹出输入密码对话框

showInterPswdDialog();

} else {

// 弹出设置密码对话框

showSetUpPswdDialog();

}

break;

case 1: // 点击通讯卫士

startActivity(SecurityPhoneActivity.class);

break;

case 2: // 软件管家

startActivity(AppManagerActivity.class);

break;

case 3:// 手机杀毒

startActivity(VirusScanActivity.class);

break;

case 4:// 缓存清理

startActivity(CacheClearListActivity.class);

break;

case 5:// 进程管理

startActivity(ProcessManagerActivity.class);

break;

case 6: // 流量统计

startActivity(TrafficMonitoringActivity.class);

break;

case 7: // 高级工具

startActivity(AdvancedToolsActivity.class);

break;

case 8: // 设置中心

startActivity(SettingsActivity.class);

break;

}

}

});

// 1.获取设备管理员

policyManager = (DevicePolicyManager) getSystemService(DEVICE\_POLICY\_SERVICE);

// 2.申请权限, MyDeviceAdminReciever继承自DeviceAdminReceiver

componentName = new ComponentName(this, MyDeviceAdminReciever.class);

// 3.判断,如果没有权限则申请权限

boolean active = policyManager.isAdminActive(componentName);

if (!active) {

//没有管理员的权限，则获取管理员的权限

Intent intent = new Intent(DevicePolicyManager.ACTION\_ADD\_DEVICE\_ADMIN);

intent.putExtra(DevicePolicyManager.EXTRA\_DEVICE\_ADMIN, componentName);

intent.putExtra(DevicePolicyManager.EXTRA\_ADD\_EXPLANATION, "获取超级管理员权限，用于远程锁屏和清除数据");

startActivity(intent);

}

}

/\*\*\*

\* 弹出设置密码对话框

\*/

private void showSetUpPswdDialog() {

final SetUpPasswordDialog setUpPasswordDialog = new SetUpPasswordDialog(

HomeActivity.this);

setUpPasswordDialog

.setCallBack(new cn.itcast.mobliesafe.chapter02.dialog.SetUpPasswordDialog.MyCallBack() {

@Override

public void ok() {

String firstPwsd = setUpPasswordDialog.mFirstPWDET

.getText().toString().trim();

String affirmPwsd = setUpPasswordDialog.mAffirmET

.getText().toString().trim();

if (!TextUtils.isEmpty(firstPwsd)

&& !TextUtils.isEmpty(affirmPwsd)) {

if (firstPwsd.equals(affirmPwsd)) {

// 两次密码一致,存储密码

savePswd(affirmPwsd);

setUpPasswordDialog.dismiss();

// 显示输入密码对话框

showInterPswdDialog();

} else {

Toast.makeText(HomeActivity.this, "两次密码不一致！", 0).show();

}

} else {

Toast.makeText(HomeActivity.this, "密码不能为空！", 0).show();

}

}

@Override

public void cancle() {

setUpPasswordDialog.dismiss();

}

});

setUpPasswordDialog.setCancelable(true);

setUpPasswordDialog.show();

}

/\*\*

\* 弹出输入密码对话框

\*/

private void showInterPswdDialog() {

final String password = getPassword();

final InterPasswordDialog mInPswdDialog = new InterPasswordDialog(

HomeActivity.this);

mInPswdDialog.setCallBack(new MyCallBack() {

@Override

public void confirm() {

if (TextUtils.isEmpty(mInPswdDialog.getPassword())) {

Toast.makeText(HomeActivity.this, "密码不能为空！", 0).show();

} else if (password.equals(MD5Utils.encode(mInPswdDialog

.getPassword()))) {

// 进入防盗主界面

mInPswdDialog.dismiss();

startActivity(LostFindActivity.class);

} else {

// 对话框消失，弹出土司

mInPswdDialog.dismiss();

Toast.makeText(HomeActivity.this, "密码有误，请重新输入！", 0).show();

}

}

@Override

public void cancle() {

mInPswdDialog.dismiss();

}

});

mInPswdDialog.setCancelable(true);

// 让对话框显示

mInPswdDialog.show();

}

/\*\*\*

\* 保存密码

\*

\* @param affirmPwsd

\*/

private void savePswd(String affirmPwsd) {

Editor edit = msharedPreferences.edit();

// 为了防止用户隐私被泄露，因此需要加密密码

edit.putString("PhoneAntiTheftPWD", MD5Utils.encode(affirmPwsd));

edit.commit();

}

/\*\*\*

\* 获取密码

\*

\* @return sp存储的密码

\*/

private String getPassword() {

String password = msharedPreferences.getString("PhoneAntiTheftPWD",

null);

if (TextUtils.isEmpty(password)) {

return "";

}

return password;

}

/\*\*

\* 判断用户是否设置过手机防盗密码

\*/

private boolean isSetUpPassword() {

String password = msharedPreferences.getString("PhoneAntiTheftPWD",

null);

if (TextUtils.isEmpty(password)) {

return false;

}

return true;

}

/\*\*

\* 开启新的activity不关闭自己

\*

\* @param cls 新的activity的字节码

\*/

public void startActivity(Class<?> cls) {

Intent intent = new Intent(HomeActivity.this, cls);

startActivity(intent);

}

/\*\*\*

\* 按两次返回键退出程序

\*/

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

if (keyCode == KeyEvent.KEYCODE\_BACK) {

if ((System.currentTimeMillis() - mExitTime) > 2000) {

Toast.makeText(this, "再按一次退出程序", Toast.LENGTH\_SHORT).show();

mExitTime = System.currentTimeMillis();

} else {

System.exit(0);

}

return true;

}

return super.onKeyDown(keyCode, event);

}

}

switch (position) {

case 0: // 点击手机防盗

if (isSetUpPassword()) {

// 弹出输入密码对话框

showInterPswdDialog();

} else {

// 弹出设置密码对话框

showSetUpPswdDialog();

}

break;

case 1: // 点击通讯卫士

startActivity(SecurityPhoneActivity.class);

break;

case 2: // 软件管家

startActivity(AppManagerActivity.class);

break;

case 3:// 手机杀毒

startActivity(VirusScanActivity.class);

break;

case 4:// 缓存清理

startActivity(CacheClearListActivity.class);

break;

case 5:// 进程管理

startActivity(ProcessManagerActivity.class);

break;

case 6: // 流量统计

startActivity(TrafficMonitoringActivity.class);

break;

case 7: // 高级工具

startActivity(AdvancedToolsActivity.class);

break;

case 8: // 设置中心

startActivity(SettingsActivity.class);

break;

}

}

});

/\*\*

\* 开启新的activity不关闭自己

\*

\* @param cls 新的activity的字节码

\*/

public void startActivity(Class<?> cls) {

Intent intent = new Intent(HomeActivity.this, cls);

startActivity(intent);

}

/\*\*\*

\* 按两次返回键退出程序

\*/

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

if (keyCode == KeyEvent.KEYCODE\_BACK) {

if ((System.currentTimeMillis() - mExitTime) > 2000) {

Toast.makeText(this, "再按一次退出程序", Toast.LENGTH\_SHORT).show();

mExitTime = System.currentTimeMillis();

} else {

System.exit(0);

}

return true;

}

return super.onKeyDown(keyCode, event);

}

}