im功能修改

在清单文件中添加YuntxNotifyReceiver声明，将${package}改成跟包名一致

<!-- // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ECSDK核心服务类 End\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->  
<receiver  
 android:name="com.yuntongxun.im.service.YuntxNotifyReceiver"  
 android:exported="false">  
 <intent-filter>  
 <action android:name="${package}.action.VOIP\_CALL" />  
 <action android:name="${package}.action.MESSAGE" />  
 <action android:name="${package}.action.NOTIFICATION\_CLICK" />  
 <action android:name="${package}.action.PERSON\_VERSION" />  
 <action android:name="${package}.action.SOFT\_UPDATER" />  
 </intent-filter>  
</receiver> <!-- // 应用处理程序逻辑服务 -->  
  
<service  
 android:name="com.yuntongxun.im.service.YuntxNotifyReceiver$NotifyService"  
 android:exported="false"  
 android:permission="android.permission.BIND\_JOB\_SERVICE" />

<!-- 接收容联 SDK 消息广播权限 -->  
<permission  
 android:name="${package}.permission.RECEIVE\_MSG"  
 android:protectionLevel="signature" />  
<permission  
 android:name="${package}.permission.INCOMING\_CALL"  
 android:protectionLevel="signature" /> <!-- 接收小米 SDK 消息广播权限 -->  
<permission  
 android:name="${package}.permission.MIPUSH\_RECEIVE"  
 android:protectionLevel="signature" />  
<permission  
 android:name="${package}.push.permission.MESSAGE"  
 android:protectionLevel="signature" />  
<permission  
 android:name="${package}.permission.C2D\_MESSAGE"  
 android:protectionLevel="signature" />  
  
<uses-permission android:name="${package}.permission.RECEIVE\_MSG" />  
<uses-permission android:name="${package}.permission.INCOMING\_CALL" />  
<uses-permission android:name="${package}.permission.MIPUSH\_RECEIVE" />  
<uses-permission android:name="{package}.push.permission.MESSAGE" />  
<uses-permission android:name="{package}.permission.C2D\_MESSAGE" />

添加RXPluginHelper文件，需要与WrapManager.getInstance().app\_AutoConnect同时进行初始化

WrapManager.*getInstance*().app\_AutoConnect(this);  
  
RXPluginHelper mPluginHelper = new RXPluginHelper(this);  
mPluginHelper.init();   
mPluginHelper.initVoIP();  
mPluginHelper.initCallLogClick();

标记所有消息为已读

WrapManager.*getInstance*().markAllSessionIsRead();

查看消息免打扰

WrapManager.*getInstance*().onMessageChatNewsNotify(conversation);

设置消息免打扰

WrapManager.*getInstance*().onSwitchNewsNotify(conversation,checked);

收藏

FavoriteManager.*startFavoriteListActivity*(getActivity(), 0);

无密码登陆

WrapManager.*getInstance*().app\_LoginImNo(account,"6fd990fb41be07ea940f97de8a8c16ac","RXe2mgmji4","67528a069e6749d193afd8193ca057bd", new SDKCoreHelper.OnConnectStateListener() {  
 @Override  
 public void onConnectSuccess() {  
 dismissDialog();  
 doLauncherAction(LoginActivity.this, LauncherUI.class);  
 }  
  
 @Override  
 public void onConnectFailed(ECError error) {  
 dismissDialog();  
 }  
});