**Android消息推送完美方案**

推送功能在手机应用开发中越来越重要，已经成为手机开发的必须。在Android应用开发中，由于众所周知的原因，Android消息推送我们不得不大费周折。本文就是用来和大家共同探讨一种Android消息推送的完美解决方案。

**1.消息推送基础**

         消息推送，就是在互联网上通过定期传送用户需要的信息来减少信息过载的一项新技术。推送技术通过自动传送信息给用户，来减少用于网络上搜索的时间。它根据用户的兴趣来搜索、过滤信息，并将其定期推给用户，帮助用户高效率地发掘有价值的信息

当我们开发需要和服务器交互的移动应用时，基本上都需要和服务器进行交互，包括上传数据到服务器，同时从服务器上获取数据。

一般情况下，客户端与服务器之间通讯客户端是主动的，但这就存在一个问题就是一旦服务器数据有更新或者服务器要下发通知给客户端只能等客户端连接的时候才能实现。这种方式使消息失去了实时性。

如何使客户端能够实时的收到服务器的消息和通知，总体来说有两种方式，第一种是客户端使用Pull（拉）的方式，就是隔一段时间就去服务器上获取一下信息，看是否有更新的信息出现。第二种就是 服务器使用Push（推送）的方式，当服务器端有新信息了，则把最新的信息Push到客户端上。这样，客户端就能自动的接收到消息。

虽然Pull和Push两种方式都能实现获取服务器端更新信息的功能，但是明显来说Push方式比Pull方式更优越。因为Pull方式更费客户端的网络流量，更主要的是费电量，还需要我们的程序不停地去监测服务端的变化。

**2. 几种常见的解决方案实现原理**

1）轮询(Pull)方式：客户端定时向服务器发送询问消息，一旦服务器有变化则立即同步消息。

2）SMS(Push)方式：通过拦截SMS消息并且解析消息内容来了解服务器的命令，但这种方式一般用户在经济上很难承受。

 3）持久连接(Push)方式：客户端和服务器之间建立长久连接，这样就可以实现消息的及时行和实时性。

**3、消息推送解决方案概述**

**A、C2DM云端推送方案**

在Android手机平台上，Google提供了C2DM（Cloudto Device Messaging）服务。Android Cloud to Device Messaging (C2DM)是一个用来帮助开发者从服务器向Android应用程序发送数据的服务。该服务提供了一个简单的、轻量级的机制，允许服务器可以通知移动应用程序直接与服务器进行通信，以便于从服务器获取应用程序更新和用户数据。

该方案存在的主要问题是C2DM需要依赖于Google官方提供的C2DM服务器，由于国内的网络环境，这个服务经常不可用。

**B、MQTT协议实现Android推送**

采用MQTT协议实现Android推送功能也是一种解决方案。MQTT是一个轻量级的消息发布/订阅协议，它是实现基于手机客户端的消息推送服务器的理想解决方案。

wmqtt.jar 是IBM提供的MQTT协议的实现。我们可以从这里（https://github.com/tokudu/AndroidPushNotificationsDemo）下载该项目的实例代码，并且可以找到一个采用PHP书写的服务器端实现（https://github.com/tokudu/PhpMQTTClient）。

**C、RSMB实现推送功能**

Really Small Message Broker (RSMB) ，是一个简单的MQTT代理，同样由IBM提供，其查看地址是：http://www.alphaworks.ibm.com/tech/rsmb。缺省打开1883端口，应用程序当中，它负责接收来自服务器的消息并将其转发给指定的移动设备。SAM是一个针对MQTT写的PHP库。我们可以从这个http://pecl.php.net/package/sam/download/0.2.0地址下载它.

**D、XMPP协议实现Android推送**

Google官方的C2DM服务器底层也是采用XMPP协议进行的封装。XMPP(可扩展通讯和表示协议)是基于可扩展标记语言（XML）的协议，它用于即时消息（IM）以及在线探测。这个协议可能最终允许因特网用户向因特网上的其他任何人发送即时消息。

androidpn是一个基于XMPP协议的java开源Android push notification实现。它包含了完整的客户端和服务器端。但也存在一些不足之处：

1） 比如时间过长时，就再也收不到推送的信息了。

2）性能上也不够稳定。

3）如果将消息从服务器上推送出去，就不再管理了，不管消息是否成功到达客户端手机上。

如果我们要使用androidpn，则还需要做大量的工作，需要理解XMPP协议、理解Androidpn的实现机制，需要调试内部存在的BUG。

**E、使用第三方平台**

目前国内、国外有一些推送平台可供使用，但是涉及到收费问题、保密问题、服务质量问题、扩展问题等等，又不得不是我们望而却步。

**4、消息推送完美方案**

     综合以上论述，在建立Android消息推送方面可谓方案多多，但每一款方案都有其优缺点。但无论如何，还是自己搭建一个推送平台是上策。因为你有、他有不如自己有。

         在搭建自有推送平台上建议使用《九日升Android消息推送组件》（http://www.bjjrs.net/product/13629681868537.html）。该组不仅可以拿来即用，并且还可以提供源码以便扩展，实现自己的特殊需求。

**A、推送原理**

    九日升Android消息推送组件基于XMPP协议实现Android推送。XMPP（可扩展通讯和表示协议）是基于可扩展标记语言（XML）的协议，它用于即时消息（IM）以及在线探测。这个协议可能最终允许因特网用户向因特网上的其他任何人发送即时消息。

    九日升Android消息推送组件实现原理见下图：

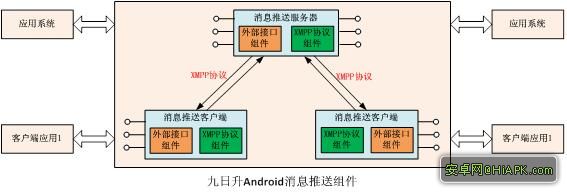


                                                  图1-消息推送原理图

    九日升Android消息推送组件由服务器部分和客户端部分组成。每一部分都由XMPP协议组件和外部接口组件构成。XMPP协议组件负责服务器和Android客户端间的连接管理、消息通讯，外部接口组件负责接收应用系统、客户端应用的命令，向应用系统发送接收到的通知消息。

    九日升Android消息组件提供基于Tomcat的服务器应用和Android开发jar包。其中基于Tomcat的服务器应用直接在Tomcat上部署即可，Android开发jar包引入Android项目即可。

**B 集成方式**

    1）、服务器部署

    九日升Android消息组件Tomcat的服务器应用直接部署在Tomcat中，端口号任意设定。

    2）、客户端jar包引用

    在Android项目中建立libs目录，然后将提供的Android开发jar包复制到该目录即可。见下图：

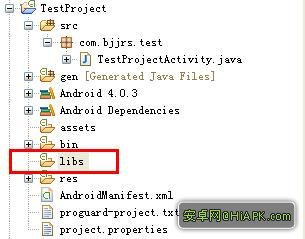


图2-jar包引入图

    3）、Android项目AndroidManifest.xml文件修改

  在该文件中增加以下权限：

     <uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

     <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

     <uses-permission android:name="android.permission.INTERNET" />

     <uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

     <uses-permission android:name="android.permission.CHANGE\_WIFI\_STATE" />

     <uses-permission android:name="android.permission.VIBRATE" />

   在该文件中注册服务：

     <service android:enabled="true"

     android:name="com.bjjrs.server.NotificationService"

     android:label="NotificationService">

          <intent-filter>

             <action android:name="com.bjjrs.server.NotificationService" />

         </intent-filter>

    </service>

   至此，九日升Android消息组件集成工作完成。

**C、接口方式**

    1）、服务器端接口采用基于http协议的访问方式，采用http协议从服务器中获取各种信息，实现通知消息的推送。

如使用以下方式和参数就可以实现各种用户消息的查询：

<http://localhost:8080/user.do?action=getAllUser&isOnline=&userID=&userType=&deptID=&deptName=&realName=>

    使用如下方式就可以实现各种消息的推送：

<http://localhost:8080/notification.do?action=pushNoti&userNames=&title=&content=>

    2）、Android客户端接口采用广播机制。

    消息接收：当XMPP协议组件接收到推送消息时，将按照一定格式广播该消息，通知客户端其他应用接收并处理该消息。

    消息发送：客户端应用需要向服务器或者其他客户端发送即时消息时，只需按一定格式广播该消息，XMPP组件就会自动接收该消息并发送到指定的其他客户端。

**D、优势特点**

    1）、系统集成简单，无需复杂的设置。

    2）、Android客户端应用和九日升Android消息推送组件完全分离，通过接口相互调用，实现模块应用最优化。

    3）、客户端通讯机制采用广播方式，给客户端应用带来极大的灵活性和可扩展性，可以自由处理接收到的推送消息。

    4）、九日升Android消息推送组件在服务器端具备消息存储、消息重发、消息路由等功能，在客户端部分具备断线重连、、收到确认、阅读确认、消息发送、命令执行等功能，确保消息能够推送到客户端，同时也保证客户端能够收到、阅读消息。

**E、 应用范围**

  九日升Android消息推送组件可在以下场景中使用：

    1）、用于消息推送。如：通知下达、应急指挥等。

    2）、用户及时消息交互。如在线聊天、工作情况交互等。

    3）、用于远程控制。如控制远程客户端的状态、数据上报等。

最后，希望转载的朋友能够尊重作者的劳动成果，加上转载地址。