[Android之消息推送实现](http://blog.csdn.net/weidi1989/article/details/7948269)

分类： [Android](http://blog.csdn.net/weidi1989/article/category/1222852)

2012-09-05 21:31 4426人阅读 评论(13) [收藏](javascript:void(0);) 举报

[android](http://blog.csdn.net/tag/details.html?tag=android)[服务器](http://blog.csdn.net/tag/details.html?tag=%e6%9c%8d%e5%8a%a1%e5%99%a8)[xmpp](http://blog.csdn.net/tag/details.html?tag=xmpp)[http服务器](http://blog.csdn.net/tag/details.html?tag=http%e6%9c%8d%e5%8a%a1%e5%99%a8)[google](http://blog.csdn.net/tag/details.html?tag=google)[sms](http://blog.csdn.net/tag/details.html?tag=sms)

在开发Android和iPhone应用程序时，我们往往需要从服务器不定的向手机客户端即时推送各种通知消息，iPhone上已经有了比较简单的和完美的推送通知解决方案，可是Android平台上实现起来却相对比较麻烦，最近利用几天的时间对Android的推送通知服务进行初步的研究。

在Android手机平台上，Google提供了C2DM（Cloudto Device Messaging）服务，起初我就是准备采用这个服务来实现自己手机上的推送功能。

*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。C2DM服务负责处理诸如消息排队等事务并向运行于目标设备上的应用程序分发这些消息。*

但是经过一番研究发现，这个服务存在很大的问题：

1）C2DM内置于Android的2.2系统上，无法兼容老的1.6到2.1系统;

2）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，如果想要很好的使用，我们的App Server必须也在国外，这个恐怕不是每个开发者都能够实现的;

有了上述两个使用上的制约，导致我最终放弃了这个方案，不过我想利用另外一篇文章来详细的介绍C2DM的框架以及客户端和App Server的相应设置方法，可以作为学习与参考之用。

即然C2DM无法满足我们的要求，那么我们就需要自己来实现Android手机客户端与App Server之间的通信协议，保证在App Server想向指定的Android设备发送消息时，Android设备能够及时的收到。下面我来介绍几种常见的方案：

1）轮询：应用程序应当阶段性的与服务器进行连接并查询是否有新的消息到达，你必须自己实现与服务器之间的通信，例如消息排队等。而且你还要考虑轮询的频率，如果太慢可能导致某些消息的延迟，如果太快，则会大量消耗网络带宽和电池。

2）SMS：在Android平台上，你可以通过拦截SMS消息并且解析消息内容来了解服务器的意图。这是一个不错的想法，我就见过采用这个方案的应用程序。这个方案的好处是，可以实现完全的实时操作。但是问题是这个方案的成本相对比较高，你很难找到免费的短消息发送网关，关于这个方案的实现，可以参考如下链接：<https://labs.ericsson.com/apis/mobile-java-push/>。

3）持久连接：这个方案可以解决由轮询带来的性能问题，但是还是会消耗手机的电池。Apple的推送服务之所以工作的很好，是因为每一台手机仅仅保持一个与服务器之间的连接，事实上C2DM也是这么工作的。不过这个方案也存在不足，就是我们很难在手机上实现一个可靠的服务。Android操作系统允许在低内存情况下杀死系统服务，所以你的通知服务很可能被操作系统Kill掉了。

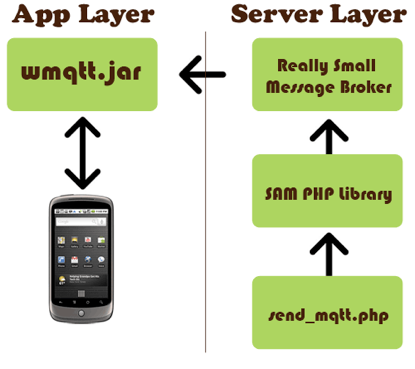
前两个方案存在明显的不足，第三个方案也有不足，不过我们可以通过良好的设计来弥补，以便于让该方案可以有效的工作。毕竟，我们要知道GMail，GTalk以及GoogleVoice都可以实现实时更新的。

Ø  采用MQTT协议实现Android推送

*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。*

我们可以从[这里](http://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](http://github.com/tokudu/PhpMQTTClient)。

架构如下所示：



**wmqtt.jar** 是IBM提供的MQTT协议的实现。你可以从如下站点[下载](http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24006006)它。你可以将该jar包加入你自己的Android应用程序中。

Really Small Message Broker (RSMB) ，他是一个简单的[MQTT代理](http://www.alphaworks.ibm.com/tech/rsmb)，同样由IBM提供。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](http://pecl.php.net/package/sam/download/0.2.0)。你可以从这个下载它.

send\_mqtt.php是一个通过POST接收消息并且通过SAM将消息发送给RSMB的PHP脚本。

实例代码：

可以从[GitHub](http://github.com/tokudu/AndroidPushNotificationsDemo)上下载实例应用。运行该应用以后，通过手机浏览器访问<http://tokudu.com/demo/android-push/>,在第一个输入框输入设备ID，在第二个输入框输入想要发送的消息内容，按下“Send Push Message”按钮，你就应该可以看到手机上收到了通知了。你也可以从这个[GitHub](http://github.com/tokudu/PhpMQTTClient)地址上下载android-push源代码，它包含了send\_mqtt.php脚本。

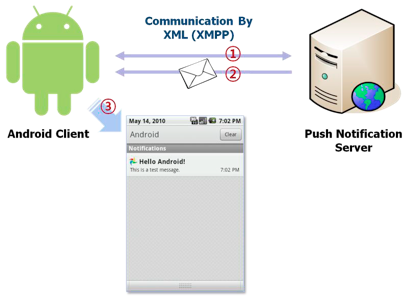


Ø  采用XMPP协议实现Android推送

这是我在项目中采用的方案。事实上Google官方的C2DM服务器底层也是采用XMPP协议进行的封装。

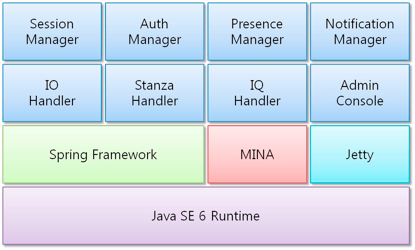
XMPP(可扩展通讯和表示协议)是基于可扩展标记语言（XML）的协议，它用于即时消息（IM）以及在线探测。这个协议可能最终允许[因特网](http://baike.baidu.com/view/1706.htm)用户向因特网上的其他任何人发送即时消息。

[androidpn](http://sourceforge.net/projects/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。经过源代码研究我发现，该服务器端基本是在另外一个开源工程[openfire](http://www.igniterealtime.org/projects/openfire/)基础上修改实现的，不过比较郁闷的是androidpn的文档是由韩语写的，所以整个研究过程基本都是读源码。它的实现示意图如下：



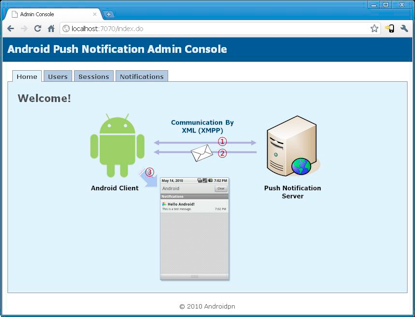
androidpn客户端需要用到一个基于java的开源XMPP协议包[asmack](http://code.google.com/p/asmack/)，这个包同样也是基于openfire下的另外一个开源项目[smack](http://www.igniterealtime.org/downloads/download-landing.jsp?file=smack/smack_3_2_0.tar.gz)，不过我们不需要自己编译，可以直接把androidpn客户端里面的asmack.jar拿来使用。客户端利用asmack中提供的XMPPConnection类与服务器建立持久连接，并通过该连接进行用户注册和登录认证，同样也是通过这条连接，接收服务器发送的通知。

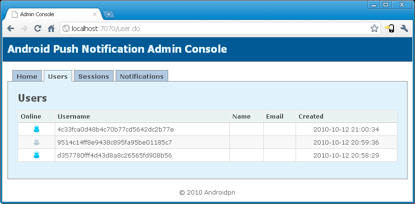
androidpn服务器端也是java语言实现的，基于openfire开源工程，不过它的Web部分采用的是spring框架，这一点与openfire是不同的。Androidpn服务器包含两个部分，一个是侦听在5222端口上的XMPP服务，负责与客户端的XMPPConnection类进行通信，作用是用户注册和身份认证，并发送推送通知消息。另外一部分是Web服务器，采用一个轻量级的HTTP服务器，负责接收用户的Web请求。服务器架构如下：

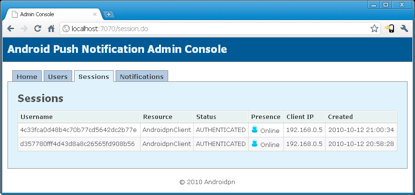


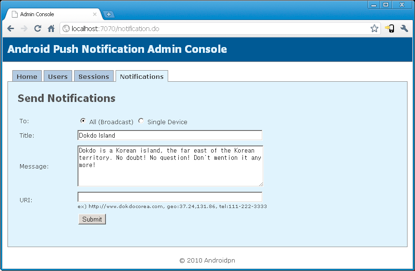
最上层包含四个组成部分，分别是SessionManager，Auth Manager，PresenceManager以及Notification Manager。SessionManager负责管理客户端与服务器之间的会话，Auth Manager负责客户端用户认证管理，Presence Manager负责管理客户端用户的登录状态，NotificationManager负责实现服务器向客户端推送消息功能。

服务器端界面如下，分别对应了上述的几个功能模块：

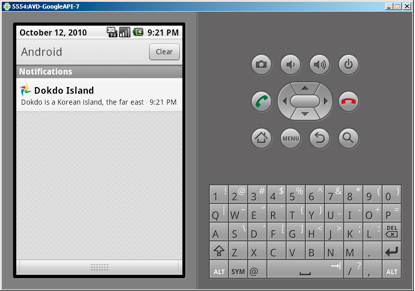








      发送以后，我们可以在手机端看到接收的消息：



      这个解决方案的最大优势就是简单，我们不需要象C2DM那样依赖操作系统版本，也不会担心某一天Google服务器不可用。利用XMPP协议我们还可以进一步的对协议进行扩展，实现更为完善的功能。

采用这个方案，我们目前只能发送文字消息，不过对于推送来说一般足够了，因为我们不能指望通过推送得到所有的数据，一般情况下，利用推送只是告诉手机端服务器发生了某些改变，当客户端收到通知以后，应该主动到服务器获取最新的数据，这样才是推送服务的完整实现。

**分享到：**

上一篇：[Java之对象池](http://blog.csdn.net/weidi1989/article/details/7943923)

下一篇：[Java之Socket与HTTP区别](http://blog.csdn.net/weidi1989/article/details/7948315)

[](http://www.baidu.com/cb.php?c=IgF_pyfqrjTzrHfznsKYTjYk0A7b5Hb4nWDkP0KbuHY4nHR3nHc0TAq15HDkP1nsP6K15yDLryn1uHm1njFWPhRknAf0uZfqnHnLnWDdnW0kn0KdThsqpZwYTjCEQv-8IAN_pgf8mLPbUB48ugfE0ZFb5HD0mhYqn0KsTWYs0ZNGujYzP163nWT10AqGujYYPWbYnsKWpyfqPWTsPHc0ULI85Hfdrj0z0APzm1YzPWDdns)

**查看评论**

9楼 [阿斯兰](http://blog.csdn.net/bhy_6655) 2013-06-07 17:02发表 [回复]

[](http://blog.csdn.net/bhy_6655)

我现在想用推送的原因：

两个客户端交互，就像QQ和微信一样。

client1与client2聊天，client1先将消息发送给server，server将更新消息的通知发给client2，然后client2根据通知去server端取数据（数据库中），大概就是想完成这样，之前用Socket实现了，不太稳定，server一旦重启或停止，客户端必须重新运行才能与server建立新的连接。所以才想用推送，如果实现微信这样的东西，有没有必要用推送

如果我实现的这个方向有问题，能不能指点一下，谢谢了

Re: [weidi1989](http://blog.csdn.net/weidi1989) 2013-06-07 17:28发表 [回复]

[](http://blog.csdn.net/weidi1989)

回复bhy\_6655：我用百度云推送写了个小例子，你参考一下，下载地址：http://url.cn/G3Iq5K

Re: [阿斯兰](http://blog.csdn.net/bhy_6655) 2013-06-08 10:40发表 [回复]

[](http://blog.csdn.net/bhy_6655)

回复weidi1989：你的那个例子我正在看

有几个疑问，

我们用百度推送的话，服务器不需要我们写么？光开发客户端么，如果方便的话，能用啥和您联系么，比如QQ什么的，我QQ 54541725

Re: [阿斯兰](http://blog.csdn.net/bhy_6655) 2013-06-08 09:41发表 [回复]

[](http://blog.csdn.net/bhy_6655)

回复weidi1989：谢谢你的帮助，我看看

关于androidpn-server-0.5.0-src.zip源码包，这server环境如何搭建呢。也不知道咋启动。官方那个例子倒是简单，运行个bat就好了，这个实在不清楚。能指点下吗，谢谢你了

8楼 [阿斯兰](http://blog.csdn.net/bhy_6655) 2013-06-07 16:55发表 [回复]

[](http://blog.csdn.net/bhy_6655)

你好，首先感谢你的这篇文章对我很有帮助。

我已经按照你说的，下载了客户端和服务端

我下载了客户端：androidpn-client-0.5.0.zip

服务端：androidpn-server-0.5.0-bin.zip（直接执行其中的run.bat完成了服务端搭建）

客户端也可以正常接收服务端的测试信息

现在有个问题，如果想根据需要去大家自己的服务端，必须用androidpn-server-0.5.0-src.zip这个包，我下载后，不知道咋用它，不知道如何搭建。。。。。。比如像web service还可以发布一下，然后启动。但是这个包我实在不知道咋启动它。。。。

能不能给我指点一下，万分感谢

7楼 [cai514667671](http://blog.csdn.net/cai514667671) 2013-06-04 17:06发表 [回复]

[](http://blog.csdn.net/cai514667671)

写的好 啊！向你学习。

6楼 [zhenglingkun](http://blog.csdn.net/zhenglingkun) 2013-05-27 16:50发表 [回复]

[](http://blog.csdn.net/zhenglingkun)

你说的最后一种方案是用在服务端的吧？

Re: [weidi1989](http://blog.csdn.net/weidi1989) 2013-05-27 19:11发表 [回复]

[](http://blog.csdn.net/weidi1989)

回复zhenglingkun：建议在天朝使用百度云推送，我最近测试了一下，效果不错。

http://developer.baidu.com/wiki/index.php?title=docs/cplat/push/api/list#query\_bindlist

5楼 [tangguoqiang1989](http://blog.csdn.net/tangguoqiang1989) 2013-05-13 11:31发表 [回复]

[](http://blog.csdn.net/tangguoqiang1989)

求介绍

4楼 [tangguoqiang1989](http://blog.csdn.net/tangguoqiang1989) 2013-05-13 11:31发表 [回复]

[](http://blog.csdn.net/tangguoqiang1989)

消息推送包在哪可以下载？

3楼 [peimm521](http://blog.csdn.net/peimm521) 2013-03-12 08:41发表 [回复]

[](http://blog.csdn.net/peimm521)

感谢分享!!!!

2楼 [xiaozhun1223](http://blog.csdn.net/xiaozhun1223) 2013-03-06 16:07发表 [回复]

[](http://blog.csdn.net/xiaozhun1223)

听的稍微懂点，却还是不知道怎么具体实现，需要例子 呵呵

1楼 [xp8812](http://blog.csdn.net/xp8812) 2012-12-07 10:54发表 [回复] [引用] [举报]

[](http://blog.csdn.net/xp8812)

不错，楼主写的很好

Android Push Notification实现信息推送使用

2012-4-14 09:01| 发布者: [benben](http://space.189works.com/space-uid-10034751.html)| 查看: 15066| 评论: 0

摘要: 本文介绍AndroidPn项目实现推送的使用过程。 上个博文中提到在**Android**中实现信息推送，其中一个比较成熟的解决方案便是使用XMPP协议实现。而AndroidPn项目就是使用XMPP协议实现信息推送的一个开源项目。在这里给大家 ...

|  |
| --- |
| **本文介绍[Android](http://bbs.189works.com/)Pn项目实现推送的使用过程。**    上个博文中提到在**Android**中实现信息推送，其中一个比较成熟的解决方案便是使用XMPP协议实现。而AndroidPn项目就是使用XMPP协议实现信息推送的一个开源项目。在这里给大家介绍其使用过程。    **Apndroid Push Notification的特点：**  快速集成:提供一种比C2DM更加快捷的使用方式,避免各种限制.  无需架设服务器:通过使用"云服务",减少额外服务器负担.  可以同时推送消息到网站页面,**android** 手机  耗电少,占用流量少.    **具体配置过程：**  首先， 我们需要下载androidpn-client-0.5.0.zip和androidpn-server-0.5.0-bin.zip。  下载地址：<http://sourceforge.net/projects/Androidpn/>    解压两个包，Eclipse导入client，配置好目标平台，打开raw/androidpn.properties文件，配置客户端程序。    1. 如果是模拟器来运行客户端程序,把xmppHost配置成10.0.2.2[模拟器把10.0.2.2认为是所在主机的地址，127.0.0.1是模拟器本身的回环地址，10.0.2.1表示网关地址，10.0.2.3表示DNS地址，10.0.2.15表示目标设备的网络地址]，关于模拟器的详细信息，大家可参阅相关资料，这里不再详述.  xmppPort=5222 是服务器的xmpp服务监听端口    运行androidpn-server-0.5.0\bin\run.bat启动服务器，从浏览器访问http://127.0.0.1:7070/index.do (androidPN Server有个轻量级的web服务器，在7070端口监听请求，接受用户输入的文本消息)    运行客户端，客户端会向服务器发起连接请求，注册成功后，服务器能识别客户端，并维护和客户端的IP长连接。    **2. 如果是在同一个局域网内的其他机器的模拟器测试(或者使用同一无线路由器wifi上网的真机) ，则需要把这个值设置为服务器机器的局域网ip.**  **例如 你的电脑和android手机 都通过同一个无线路由器wifi上网, 电脑的ip地址为 192.168.1.2 而 手机的ip地址为 192.168.1.3, 这个时候 需要把这个值修改为 xmppHost=192.168.1.1 或是电脑的IP地址，就可以在手机上使用了.**    3. 如果是不在同一个局域网的真机测试，我们需要将这个值设置为服务器的IP地址。    **具体配置如下图所示：**        **我的电脑IP是：192.168.8.107**    服务器运行主界面：              推送信息：    测试结果如下图：            模拟器和真机测试通过。    **其他使用过程介绍：**  当然了，以上是在自己的服务器上配置并实现信息推送，如果大家想省事的话，在这里给大家推荐个网址， 可以更轻松地实现推送，缺点是需要通过别人的服务器，关于这点，你懂的。  网址： <http://www.push-notification.org/>  只需申请个API即可使用。  至于其他的推送方案会在以后的博文中介绍。  完毕。 |