

个人资料



weidi1989

访问: 273862次

积分: 5493分

排名: 第1002名

原创: 144篇 转载: 3篇

译文: 1篇 评论: 1070条

小工具

码地址

订阅到QQ邮箱

博客专栏

Android实战小项目

文章: 10篇

阅读: 58139



文章搜索

文章分类

Android (118)

Android FrameWork (1)

Java (12)

ubuntu (10)

杂谈 (6)

c (1)

文章存档

2014年02月 (1)

【大声说出你的爱】CSDN社区情人节特别活动

专访李云：从通讯行业的架构师到互联网“新兵”

电子版《程序员》杂志免费领取

Android之基于XMPP协议即时通讯软件(二)

分类: Android 2013-12-18 23:37 1377人阅读 评论(22) 收藏 举报

2013博客之星投票请移步: <http://vote.blog.csdn.net/blogstaritem/blogstar2013/weidi1989>

昨天给大家粗略的介绍了一下我那个简单的项目：**Android之基于XMPP协议即时通讯软件(一)**

从今天开始，就详细展开的介绍设计思路，一是给自己做个总结，二是希望能给各位朋友一点点帮助吧！

也许和很多刚刚开始入行程序猿的朋友一样，我最初也是每开始一个新项目，只是稍微思考一下便开始写代码。一个星期后，尽管实现了部分功能，却发现自己有点乱了，因为又突然生出很多新的想法，于是又顺着思路写下去，再持续一段时间，就会发现这个项目做不下去了。代码重复很多，而且bug也是层出不穷。因此不得不推翻重新开始，浪费了很多时间。所以说，开始前的准备工作是相当重要的。

一般说来完成一个项目可以分成三步：

首先：画几天时间好好理清思路，可以在草稿纸上写写画画，列出必须实现的功能已经界面特效等等，做到胸有成竹。

其次：开始搭建框架，定义好所有功能的接口，并且注释好做什么用，这是方便团队合作的项目，个人项目也可以养成写注释的好习惯。

最后：分开实现各部分功能，并且分开调试，可以单独独立成一个应用或者单元测试模块。比如说本应用的所有界面效果都是独立成一个应用测试无误后才最后合入的。

我这个项目呢，一开始是完成了所有的界面模块工作，因为是直接从之前的项目移植过来的，稍微修改了一下，所以这部分省掉了很多时间。

重心就直接放在了功能实现这方面了。功能模块部分就是下面两个包：**com.way.service**和**com.way.smack**

com.way.service

BaseService.java

IConnectionStatusCallback.java

XXService.java

com.way.slidinglayer

com.way.slidingmenu

com.way.smack

Smack.java

SmackImpl.java

首先是定义通用接口，主要就是将第三方包**asmack.jar**中需要用到东西拿出来，这个包是非常庞大的，但是我们这个小应用并不是需要实现所有的功能，所以，抽取自己需要的那部分就可以了，定义成接口，方便以后扩展需要的功能，代码如下：

```
[java]
1. /**
2.  * 首先定义一些接口，需要实现一些什么样的功能，
3.  *
4.  * @author way
5.  *
6.  */
7. public interface Smack {
8.     /**
9.      * 登陆
10.      *
```

http://blog.csdn.net/way_ping_li/article/details/17403929

1/6

2014年01月 (1)
2013年12月 (5)
2013年11月 (2)
2013年10月 (4)
展开

阅读排行
Android之高仿手机QQ聊 (20132)
Android之基于百度云推 (11998)
Android之高仿墨迹天气 (11345)
Android之消息推送实现 (9524)
Android之网络学习资源 (6065)
Android之高仿微信聊天 (5935)
Android之BaseExpandable (5064)
Android将程序崩溃信息 (4870)
Java之Socket简单聊天实 (4737)
Android之指南针学习 (4210)

评论排行
Android之高仿手机QQ聊 (228)
Android之基于百度云推 (130)
Android之网络学习资源 (50)
Android之简洁天气 (47)
Android之极致拟物化空 (36)
Android之基于XMPP协议 (34)
Java之Socket简单聊天实 (34)
Android之高仿墨迹天气 (32)
Android之基于XMPP协议 (22)
Android之仿网易V3.5新 (22)

最新评论
Android4.0设置界面修改总结 tiao zhan maid i : 楼主 请问一下 你的settings的标题栏的背景是如何修改的, 急用, 谢谢了。
Android4.0设置界面修改总结 weidi1989: @sk yin myheart:学Android, 首先就需要入门咯, 入门之后, 可以在网上下一些简单的例子...
Android4.0设置界面修改总结 weidi1989: @manzer_zhou:没有源码, 但是关键代码都已经分享, 相信你认真看看, 研究一下, 绝对能实现这样...
Android4.0设置界面修改总结 weidi1989: @tiao zhan maid i :关键代码已经分享出来了, 源码分享出来, 你也编不过。
Android4.0设置界面修改总结 tiao zhan maid i : 确实非常不错, 请问楼主可以共享一下源代码吗
Android之高仿手机QQ聊天 qq530918474: 大神, 我把客户端安装在手机上, 无法运行, 请问这个问题该怎么解决?
Java之Socket简单聊天实现 (QC) fangqingyin: 不错 看上去很Nb 马上试试
ViewPager切换动画PageTransf 各种代码各种敲: 为什么我跑起来没效果呢?
Android4.0设置界面修改总结 manzer_zhou: 有源代码吗? 有发一份
Android之基于XMPP协议即时通 北鸣游羽: 楼主 账号密码不是很

```
11.      * @param account
12.      *          账号
13.      * @param password
14.      *          密码
15.      * @return 是否登陆成功
16.      * @throws XXException
17.      *          抛出自定义异常, 以便统一处理登陆失败的问题
18.      */
19.      public boolean login(String account, String password) throws XXException;
20.
21.      /**
22.       * 注销登陆
23.       *
24.       * @return 是否成功
25.       */
26.      public boolean logout();
27.
28.      /**
29.       * 是否已经连接上服务器
30.       *
31.       * @return
32.       */
33.      public boolean isAuthenticated();
34.
35.      /**
36.       * 添加好友
37.       *
38.       * @param user
39.       *          好友id
40.       * @param alias
41.       *          昵称
42.       * @param group
43.       *          所在的分组
44.       * @throws XXException
45.       */
46.      public void addRosterItem(String user, String alias, String group)
47.          throws XXException;
48.
49.      /**
50.       * 删除好友
51.       *
52.       * @param user
53.       *          好友id
54.       * @throws XXException
55.       */
56.      public void removeRosterItem(String user) throws XXException;
57.
58.      /**
59.       * 修改好友昵称
60.       *
61.       * @param user
62.       *          好友id
63.       * @param newName
64.       *          新昵称
65.       * @throws XXException
66.       */
67.      public void renameRosterItem(String user, String newName)
68.          throws XXException;
69.
70.      /**
71.       * 移动好友到新分组
72.       *
73.       * @param user
74.       *          好友id
75.       * @param group
76.       *          新组名
77.       * @throws XXException
78.       */
79.      public void moveRosterItemToGroup(String user, String group)
80.          throws XXException;
81.
82.      /**
83.       * 重命名分组
84.       *
85.       * @param group
86.       *          之前的组名
87.       * @param newGroup
88.       *          新组名
89.       */
```

理解 什么意思。。自定义的吗 还是用你的

```

90.         public void renameRosterGroup(String group, String newGroup);
91.
92.         /**
93.          * 请求好友重新授权,用在添加好友失败时,重复添加 再次向对方发出申请
94.          *
95.          * @param user
96.          *      好友id
97.          */
98.         public void requestAuthorizationForRosterItem(String user);
99.
100.        /**
101.         * 添加新分组
102.         *
103.         * @param group
104.         */
105.        public void addRosterGroup(String group);
106.
107.        /**
108.         * 设置当前在线状态
109.         */
110.        public void setStatusFromConfig();
111.
112.        /**
113.         * 发送消息
114.         *
115.         * @param user
116.         * @param message
117.         */
118.        public void sendMessage(String user, String message);
119.
120.        /**
121.         * 向服务器发送心跳包,保持长连接 通过一个闹钟控制,定时发送,
122.         */
123.        public void sendServerPing();
124.
125.        /**
126.         * 从jid中获取好友名
127.         *
128.         * @param jid
129.         * @return
130.         */
131.        public String getNameForJID(String jid);
132.    }

```

真正的实现就留给他的子工具类**SmackImpl**。昨天我说过,本应用只有一个服务:**XXService**,他时继承自**BaseService**,**BaseService**也是完成了一部分通用的功能,比如说通知栏消息处理等。在**XXService**中,我们主要是处理**Activity**与**Smack**中连接工作。每个**Activity**在**onResume**时,就与该**XXService**绑定,绑定之后,便可调用该**Service**中的所有公共函数,比如说登陆、发消息等等,然后**Service**再调用**Smack**中已经定义好的各种接口函数,即可完成与服务器的交互。没错,其实这就是**MVC**设计模式。**Service**担当这个**C**的角色,**Activity**就是**V**的角色,**Smack**就是**M**了。

现在我们的**Activity**可以调用**Service**中的函数提交消息给服务器了,这是单方面的,我们同时需要获得服务器返回的信息,比如说是否登陆成功、收消息之类。这个任务就自然的落到了回调接口**IConnectionStatusCallback**身上了,在绑定服务成功的同时,我们注册此回调接口,下面是此接口定义的唯一一个函数。也许眼尖的朋友会问了,为什么没有收到新消息的回调,其实一开始我是加了的,后面因为将数据库改成了**ContentProvider**,就可以去掉了,因为收消息可以通过监听数据库动态变化嘛,对吧?这就是**ContentProvider**的好处所在,这个留待后续详细说,这里只是提及一下。

```

[java]
01. /**
02.  * 服务器回调接口
03.  *
04.  * @author way
05.  *
06.  */
07. public interface IConnectionStatusCallback {
08.     /**
09.      * 连接状态改变
10.      *
11.      * @param connectedState
12.      *      连接状态,有连接、未连接、链接中三种

```

```
13.         * @param reason
14.         *             连接失败的原因
15.         */
16.         public void connectionStatusChanged(int connectedState, String reason);
17.     }
```

在绑定服务成功的时候，将此回调传递给Service，比如登陆界面的ServiceConnection：

```
[java]
01. ServiceConnection mServiceConnection = new ServiceConnection() {
02.
03.     @Override
04.     public void onServiceConnected(ComponentName name, IBinder service) {
05.         mXxService = ((XXService.XXBinder) service).getService();
06.         //注册回调
07.         mXxService.registerConnectionStatusCallback(LoginActivity.this);
08.     }
09.
10.     @Override
11.     public void onServiceDisconnected(ComponentName name) {
12.         mXxService.unregisterConnectionStatusCallback();
13.         mXxService = null;
14.     }
15.
16. };
```

就这样，整体与服务器交互的基本框架就完成了。很简单对吧？
OK，今天就到这里了，以上仅是本人的一点心得体会，后续将详细讲解功能接口Smack的实现以及如何处理联系人和聊天记录数据库等功能，非常感谢你看到了文章末尾，话说，你是否忘记投票了呢？0(∩_∩)0哈哈~

更多 0

上一篇：[Android之基于XMPP协议即时通讯软件\(一\)](#)
下一篇：[Android之基于XMPP协议即时通讯软件\(三\)](#)

顶 11 踩 0

跨平台表格控件 WinForms · ASP.NET · HTML5 · WinRT · WPF · Silverlight

Spread Studio 7 全新发布

免费下载

查看评论

11楼 [jinzhi kai12](#) 2014-01-23 14:14发表



我蛮希望楼主可以细分一下，你的应用的功能点与对应的包，我感觉自己看的蛮吃力的

10楼 [文笔天下](#) 2014-01-15 16:34发表



楼主，请问为什么我修改了ip地址和端口号以及服务器名称，也使用了正确的用户名，却登陆不进去呢，提示服务器没有响应，但我搭建的openfire服务器确实可以用啊

9楼 [GISChina](#) 2013-12-26 15:22发表



“每个Activity在onResume时，就与该XXService绑定，绑定之后，便可调用该Service中的所有公共函数”---你的每个Activity中都要与XXService绑定，每个Activity中都有这部分重复的代码，不是很繁琐吗，有没有其他的方法呢？

Re: [weidi1989](#) 2013-12-26 15:30发表



回复GISChina：你再仔细看看都是在onResume中绑定服务的吗？

Re: [GISChina](#) 2013-12-31 09:43发表

回复weidi1989：还有的是在onWindowFocusChanged这个里，我想知道为什么要频繁的绑定服务



和解绑服务, 谢谢你的回复

8楼 蓝斯 2013-12-22 00:03发表



LZ不错啊, 都甄选博客之星了
窝都好久没更新博客了, 无力中。。。

Re: weidi1989 2013-12-23 12:42发表



回复geniuseoe2012: 我也是意外入选的, 呵呵, 其实你可以争取一下的!

7楼 wnygg 2013-12-21 14:56发表



楼主, 什么时候推出三呢, 期待中!

Re: weidi1989 2013-12-23 12:41发表



回复wnygg: 已经出了, 欢迎关注: http://blog.csdn.net/way_ping_li/article/details/17490377

6楼 busyboxs 2013-12-20 22:33发表



```
SmackConfiguration.setKeepAliveInterval(-1);  
SmackConfiguration.setDefaultPingInterval(0);
```

像这些代码, 如果不是花很多时间去看它的Javadoc, 很难才能找到吧。

5楼 busyboxs 2013-12-20 17:23发表



大神, 你是在哪学的呀? 越看你的代码就越崇拜你! 看了那么多的demo, 你的是最与众不同, 最让人着迷的! 膜拜啊

Re: weidi1989 2013-12-20 21:45发表



回复busyboxs: 在学校里面找点视频自己学啊, 然后自己敲两个简单的项目, 就入门了。工作中就靠自己发挥了

Re: busyboxs 2013-12-20 22:44发表



回复weidi1989: 我现在大三了, 感觉再不努力大学就过了, 现在在帮老师做xmpp的项目, 资料都是上csdn或者eoe上找的, 唉, 都是泪, 很多文章只告诉你怎样login和发信息等等很简单的东西, javadoc又全是英文, 哭了。看了你的代码, 我会高兴很多天的。我真的很想知道, 你是不是真的把smack的javadoc全看了呀? 而且你没有把整个asmack整个导进来, 只取有用的部分, 说明你已经知道他里面都能干啥了呀。
、、学渣哭泣中、、膜拜

Re: weidi1989 2013-12-20 23:21发表



回复busyboxs: 当然没有全部看完, 这个库有文档的, 我一般是需要什么功能, 才会去详细找。再不理解, 就上百度找, 有很多人写了简单例子, 其实这个很好, 因为有时候, 我就仅仅只需要理解那部分功能

Re: busyboxs 2013-12-21 12:38发表



回复weidi1989: 文档是有, 但是都是简单几句, 主要还是看英文的JavaDoc啊, 哭了

4楼 busyboxs 2013-12-20 17:07发表



看了大神的源码, 发现大神写activity没有使用aquarey或者afinal之类的框架, 真大神啊!

Re: weidi1989 2013-12-20 18:44发表



回复busyboxs: 额, 没那么夸张吧? 我这个本来就不需要用到那个框架啊!

Re: busyboxs 2013-12-20 21:20发表



回复weidi1989: 大神回复我了, 大神学了多久了呀

3楼 busyboxs 2013-12-20 17:05发表



大神!

2楼 华华金英 2013-12-19 16:41发表



已经运行起来了, 用的是openfire, 非常感谢你的分享!

1楼 zx593017083 2013-12-19 16:25发表



讲解的非常细致, 期待大牛的进一步讲解。还有源码, 哈哈!

Re: weidi1989 2013-12-23 12:43发表



回复z593017083: http://blog.csdn.net/way_ping_li/article/details/17490377
新的文章已发布，谢谢关注

发表评论

用户名: u013295109

评论内容:

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

专区推荐内容

- 企业安全技术（1）：检查平台AE...
- 使用非 html5 实现 js ...
- 跟燕青一起学Android应用开...
- 苹果的顶级应用程序：设计，语言，...
- 2013年热门Android应用
- 如何在Windows下编译Ope...

<< >>

更多招聘职位

- 【深圳市职帮人力资源有限公司】Java
- 【浪腾信息】asp.net 程序员
- 【英特尔亚太研发中心（上海）】Big I
- 【猎头中国】虚拟化资深设计师/架构师
- 【畅捷通信息技术股份有限公司】JAVA
- 【北京搜狗科技发展有限公司】大数据

核心技术类目

- 全部主题
- Java VPN Android iOS ERP IE10 Eclipse CRM JavaScript Ubuntu NFC
- WAP jQuery 数据库 BI HTML5 Spring Apache Hadoop .NET API HTML SDK
- IIS Fedora XML LBS Unity Splashtop UML components Windows Mobile Rails
- QEMU KDE Cassandra CloudStack FTC coremail OPhone CouchBase 云计算 iOS6
- Rackspace Web App SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby
- Hibernate ThinkPHP Spark HBase Pure Solr Angular Cloud Foundry Redis Scala
- Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告
QQ客服 微博客服 论坛反馈 联系邮箱: webmaster@csdn.net 服务热线: 400-600-2320

京 ICP 证 070598 号

北京创新乐知信息技术有限公司 版权所有
江苏乐知网络技术有限公司 提供商务支持

Copyright © 1999-2014, CSDN.NET, All Rights Reserved

