登陆验证

参数：用户名（圈号或手机号或user\_name)、密码、验证码

返回：host\_id = user\_id；

注册

参数：相关信息（细化）

在数据库中新成立一个注册用户id，返回host\_id = user\_id；

查用户名是否存在

查手机号是否存在

主页话题列表

事件：应用启动，请求初步数据

数据返回格式： [@"room\_id",@"owner\_id",@"owner\_name",@"room\_type",@"room\_title",@"room\_postTime",@"room\_guestCount",@"room\_labels"]

其中，room\_guestCount需要通过服务器计算；room\_labels是一个该话题所有标签的数组。

请求动作：

0.1更新本地好友关系目录

参数：@user\_id

返回条件

如果有更新，则在后台下载更新本地好友关系目录；

疑问：如何检查更新？（上传上一次更新时间，在这更新时间以后查询记录）

0.2更新用户本人档案

参数：@user\_id

如果有更新，则在后台下载用户本人最新档案信息；

1.查看全部收到的话题

参数：@user\_id，上一次缓存读取到的最新一条信息的@room\_id和@room\_postTime

返回条件：

删选：用户接收到的话题

排序：按时间排序

数量：最新的30条；

本地：主页TableView显示，同时在后台开始请求下载 话题详细 和 好友档案 的信息；

疑问：请求时，好友关系目录在服务器上生成，还是用本地的好友关系目录？（应该在服务器上生成）

2.查看全部与我相关的话题

参数：@host\_id

返回条件：用户发起或参与过的话题

排序：先按是否有最新回复排序（reply\_unread是否有host\_id），按时间排序

数量：最新的30条；

本地：主页TableView显示

疑问：需不需要同时在后台开始请求下载 话题详细 和 好友档案 的信息？（如果相应速度很快，就不需要）

3.查看更多的话题（具体要改）

参数：@user\_id，@已成功请求次数n（客户端本地用来记录请求数量的值）

返回条件：返回数据库中第n\*30+1至(n+1)\*30条的数据

本地：加入到主页TableView加载数据的MutableArray中

4.直接搜索话题标签关键字

另起一个View Controller，重复1，2，3的动作；

参数：@user\_id，@search\_word

返回条件：用户能够看到的房间中，标题内包含关键字的部分

5.直接搜索话题分类

另起一个View Controller，重复1，2，3的动作；

参数：@user\_id，@room\_label

返回条件：用户能够看到的房间中，对应标签的部分

6.发起新话题；

7.检索好友的话题

8.只检索自己的回复

9.只检索自己的话题

10. 从服务器下载分类列表

话题详细（回复列表）

事件：用户点击话题明细，进入话题详细页面

数据返回格式：[@"topic\_id",@"reply\_id",@"reply\_time",@"reply\_senderID",@"reply\_sender",@"reply\_content",@"reply\_visibility",@"reply\_hostAgreeCount",@"reply\_guestAgreeCount",@"reply\_level",@“reply\_receiver”，@“reply\_unread"]

其中，reply\_senderID即是sender的用户id；reply\_\*\*AgreeCount是 房主或客人的点赞；reply\_level是回复的楼层数；reply\_receiver是话题@的人（Array），默认有owner\_id；reply\_unread用来指定待读的人，初始化时Copy reply\_receiver，

请求动作：

1.显示话题的明细（即回复列表）

参数：@room\_id，@host\_id

返回条件：该Room的相应详细信息，显示visible的信息（用host\_id和reply\_visibility及owner\_id比对）；同时将reply\_unread中的host\_id去掉；

2.发起新的回复

参数：自动生成符合数据格式的新数据

动作：上传服务器（后台进行），若上传成功，则在本地的数据中先更新（无须重新请求）；

3.Guest给回复点赞

参数：@reply\_id

动作：上传服务器计数加1（后台进行），前台直接加1

4.Host给回复点赞

参数：@reply\_id

动作：上传服务器hostAgreeCount计数加1，host\_id和reply\_id好友关系值加1（按友好互动次数计），前天直接加1

5.分享功能

使用其他软件KPI，另行设计；

档案页面

上传所要查看用户的id，返回该用户的档案信息（字典格式），每个档案的数据格式如下： [@"profile\_id",@"profile\_image",@"profile\_name",@"profile\_gender",@"profile\_intro",@"profile\_city",@"profile\_field",@"profile\_org",@"profile\_title",@"profile\_friendCount",@"profile\_friendRank",@"profile\_roomCount",@“profile\_replyCount”，@“relation\_type"]

其中最后四个用户行为统计，都是要通过Server计算出来的；profile\_image应该是图片的URL地址。relation\_type是描绘host和该用户之间关系的词语；

疑问：reply\_type可能需要一个单独的请求

和本地好友比对，若为一度好友，不请求；若为二度好友，发送请求；若为二度以上，不请求；

二度好友请求：

参数：@host\_id；@profile\_id；

动作：返回一个两个id共同好友的集合；

1.更新头像

图片，本地处理，上传图片

疑问：图片服务器怎么设？

好友页面

上传索要查看用户的id，返回该用户的一度好友列表及二度好友统计，每个一度好友列表的数据如下： [@"friend\_id",@"friend\_name",@"friend\_image"]

每个二度好友统计，应该返回三个数组，分别是： 城市数组：全部二度好友的城市分布，按频率排序； 行业数组：全部二度好友的行业分布，按频率排序； 职位数组：全部二度好友的职位分布，按频率排序；

0.下载好友清单

好友清单应该包括两个数组：一度好友数组，二度好友数据（已经查重）（肯定使用缓存）

疑问：怎么查有没有变化？还是不查，直接全部下载？（最好还是只进行修改，而不是重新下载）

0.1二度好友统计

疑问：以二度清单发起新的请求，还是直接将统计数据在下载清单的时候给带下来？

1.删除动作

2.添加好友

在通知中心生成一个通知中心的条目，包括host\_id, profile\_id，默认生成的notice\_content，然后条目又能显示在房间列表中；

检查共同好友

搜索好友

手机号查询，返回好友是否存在

配置邀请码，发送邀请码

设置页面

更改用户名和密码

更改个人资料

读取FAQ HTML页面

发送反馈意见

检查版本更新