

无线 iphone 客户端测试白皮书

版本：V0.1

作者：冰心

时间：2012 年 6 月

目录

做无线客户端测试时，项目流程有区别吗？	2
服务端测试	2
客户端数据库介绍	2
客户端测试	3
1、 网络	4
2、 应用的前后台切换	4
3、 数据更新	4
4、 免登录	5
5、 离线浏览	5
6、 时间测试	5
7、 PUSH 测试	5
8、 定位、照相机服务	6
9、 客户端更新	6
10、 CHECK CODE	6
11、 其它比较常见的通用内容	6
问题排查	7

做无线客户端测试时，项目流程有区别吗？

无线客户端的项目测试流程与 PC 测试基本一样的，只是在对 UI 这块，要求更高，需要更加注重用户体验。。对于一个小小的屏幕，如何让用户使用更加轻便、简洁、易用，包括整个 app 的色调搭配，是否会让用户看了有疲劳感，都是需要我们考虑的。所以，UI 的评审相当地重要。

在进行视觉稿的评审时，需要注意：请将图片放到手机上进行评审，效果最佳，因为手机上的效果和 PC 上显示的效果是有很大区别的。往往 PC 端上看上去的效果很好，但是放到手机上后可能就不太好了，这时候如果再去调整视觉设计，对整个项目的进度会有非常大的影响。

服务端测试

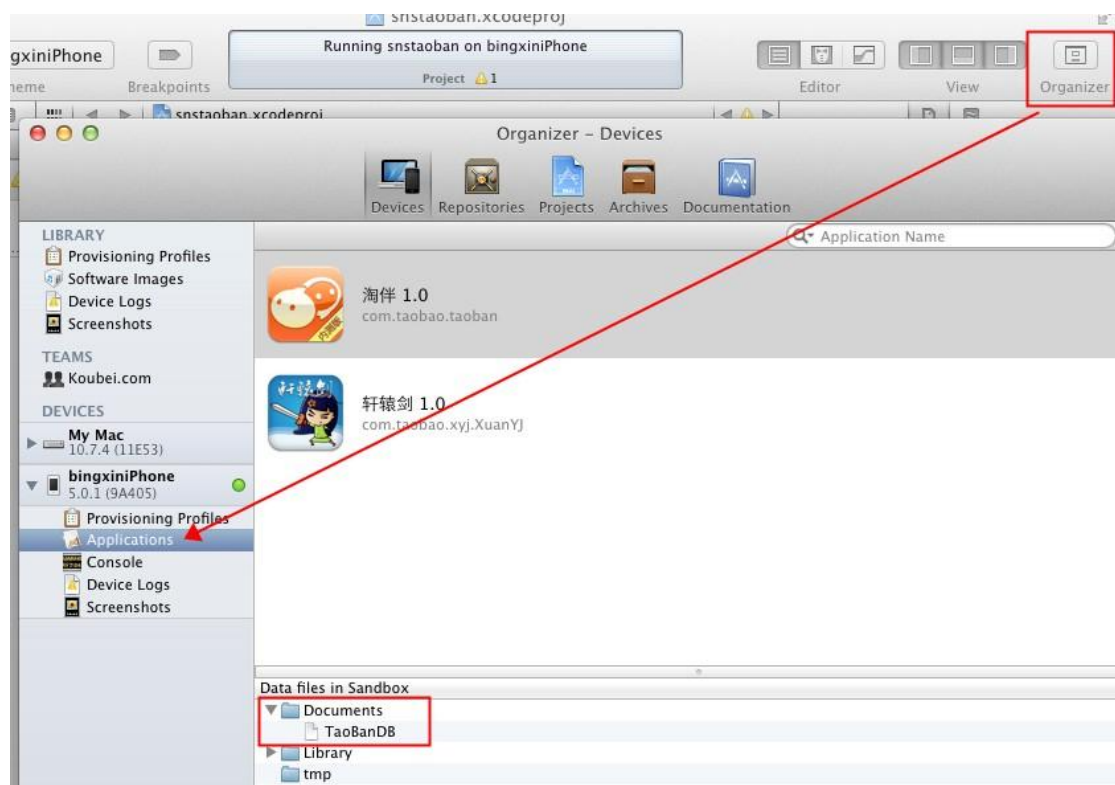
服务端一般会提供 JSON 格式的数据给客户端，所以我们在服务端需要进行接口测试，确保服务端提供的接口并转换的 JSON 内容正确，对分支、异常流有相应的返回值。此块测试可以采用 itest 框架进行测试。最方便的是采用 httpclient 进行接口测试。进行服务端测试时，需要开发提供一份接口文档。

客户端数据库介绍

客户端目前采用 sqlite 数据库，当 app 需要在客户端本地保存数据时，app 会创建相应的数据库表。

模拟器的数据库地址：<target_app>/Documents/下会有个 db 文件，此为客户端的数据库。

真机的数据库地址，可以通过 XCODE，点击 organizer，选择相应的手机-application 中，Documents 下会有个 DB 文件，如下图。



客户端数据库连接方式，可以安装 navicat 的 sqlite 客户端，也可以安装 Firefox 插件：sqlite manager 来查询、修改数据。

注意：真机在未越狱情况下，需要将数据文件导出后查看。

PC 端的数据库与客户端数据库有一个很明显的区别是：PC 端的数据库一般都是先建好的，而客户端是在需要用到相应的表时才会去创建的。所以我们需要除了和测试 PC 一样的数据增、删、改、查是否正确之外，还需要测试：

- 1) 当表不存在时是否能自动创建，当数据库表被删除后能否再自建，数据是否还能自动从服务端中获取回来并保存。
- 2) 在业务需要从服务端取回数据保存到客户端的时候，客户端能否将数据保存到本地。
- 3) 当业务需要从客户端取数据时，检查客户端数据存在时，app 数据是否能自动从客户端数据中取出，还是仍然会从服务器端获取？检查客户端数据不存在时，app 数据能否自动从服务器端获取到并保存到客户端
- 4) 当业务对数据进行了修改、删除后，客户端和服务端是否会有相应的更新。

客户端测试

在测试时，最好手机连上 xcode 进行测试，方便查看日志，程序崩溃等原因。下面整理了一些客户端方的测试总结。

1、 网络

手机的网络目前主要分为 2G、3G、wifi。目前 2G 的网络相对于比较慢，测试时尤其要注意此块的测试。

- 1) 无网络时，执行需要网络的操作，给予友好提示，确保程序不出现 crash。
- 2) 内网测试时，要注意选择到外网操作时的异常情况处理。
- 3) 在网络信号不好时，检查功能状态是否正常，确保不因提交数据失败而造成 crash。
- 4) 在网络信号不好时，检查数据是否会一直处于提交中的状态，有无超时限制。如遇数据交换失败时要给予提示。
- 5) 在网络信号不好时，执行操作后，在回调没有完成的情况下，退出本页面或者执行其他操作的情况，有无异常情况。此问题也会经常出现程序 crash。

2、 应用的前后台切换

- 1) APP 切换到后台，再回到 app，检查是否停留在上一次操作界面。
- 2) APP 切换到后台，再回到 app，检查功能及应用状态是否正常，IOS4 和 IOS5 的版本的处理机制有的不一样。
- 3) app 切换到后台，再回到前台时，注意程序是否崩溃，功能状态是否正常，尤其是对于从后台切换回前台数据有自动更新的时候。
- 4) 手机锁屏解屏后进入 app 注意是否会崩溃，功能状态是否正常，尤其是对于从后台切换回前台数据有自动更新的时候。
- 5) 当 App 使用过程中有电话进来中断后再切换到 app，功能状态是否正常
- 6) 当杀掉 app 进程后，再开启 app，app 能否正常启动。
- 7) 出现必须处理的提示框后，切换到后台，再切换回来，检查提示框是否还存在，有时候会出现应用自动跳过提示框的缺陷。
- 8) 对于有数据交换的页面，每个页面都必需要进行前后台切换、锁屏的测试，这种页面最容易出现崩溃。

3、 数据更新

根据应用的业务规则，以及数据更新量的情况，来确定最优的数据更新方案。

- 1) 需要确定哪些地方需要提供手动刷新，哪些地方需要自动刷新，哪些地方需要手动+自动刷新。
- 2) 确定哪些地方从后台切换回前台时需要进行数据更新。
- 3) 根据业务、速度及流量的合理分配，确定哪些内容需要实时更新，哪些需要定时更新。
- 4) 确定数据展示部分的处理逻辑，是每次从服务端请求，还是有缓存到本地，这样才能有针对性的进行相应测试。
- 5) 检查有数据交换的地方，均有相应的异常处理。

4、 免登录

很多应用提供免登录功能，当应用开启时自动以上一次登录的用户身份来使用 app.

- 1) app 有免登录功能时，需要考虑 IOS 版本差异。
- 2) 考虑无网络情况时能否正常进入免登录状态。
- 3) 切换用户登录后，要校验用户登录信息及数据内容是否相应更新，确保原用户退出。
- 4) 根据 MTOP 的现有规则，一个帐户只允许登录一台机器。所以，需要检查一个帐户登录多台手机的情况。原手机里的用户需要被踢出，给出友好提示。
- 5) app 切换到后台，再切回前台的校验
- 6) 切换到后台，再切换回前台的测试
- 7) 密码更换后，检查有数据交换时是否进行了有效身份的校验
- 8) 支持自动登录的应用在进行数据交换时，检查系统是否能自动登录成功并且数据操作无误。
- 9) 检查用户主动退出登录后，下次启动 app，应停留在登录界面。

5、 离线浏览

很多应用会支持离线浏览，即在本地客户端会缓存一部分数据供用户查看。

- 1) 在无网络情况可以浏览本地数据
- 2) 退出 app 再开启 app 时能正常浏览
- 3) 切换到后台再切回前台可以正常浏览
- 4) 锁屏后再解屏回到应用前台可以正常浏览
- 5) 在对服务端的数据有更新时会给予离线的相应提示

6、 时间测试

客户端可以自行设置手机的时区、时间，因此需要校验该设置对 app 的影响。

iphone 的时间设置:设置——通用——日期与时间，将自动设置关闭，修改时区、时间。

中国为东 8 区，所以当手机设置的时间非东 8 区时，查看需要显示时间的地方，时间是否展示正确，应用功能是否正常。时间一般需要根据服务器时间再转换成客户端对应的时区来展示，这样的用户体验比较好。比如发表一篇微博在服务端记录的是 10:00，此时，华盛顿时间为 22:00，客户端去浏览时，如果设置的是华盛顿时间,则显示的发表时间即为 22:00,当时间设回东 8 区时间时，再查看则显示为 10:00。

7、 PUSH 测试

- 1) 检查 push 消息是否按照指定的业务规则发送
- 2) 检查不接受推送消息时，检查用户不会再接收到 push.
- 3) 如果用户设置了免打扰的时间段，检查在免打扰时间段内，用户接收不到 PUSH。

在非免打扰时间段，用户能正常收到 push.

- 4) 当 push 消息是针对登录用户的时候，需要检查收到的 push 与用户身份是否相符，没有错误地将其它人的消息推送过来。一般情况下，只对手机上最后一个登录用户进行消息推送。
- 5) 测试 push 时，需要采用真机进行测试。

8、 定位、照相机服务

- 1) App 有用到相机，定位服务时，需要注意 IOS 版本差异
- 2) 有用到定位服务、照相机服务的地方，需要进行前后台的切换测试，检查应用是否正常。
- 3) 当定位服务没有开启时，使用定位服务，会友好性弹出是否允许设置定位提示。当确定允许开启定位时，能自动跳转到定位设置中开启定位服务。
- 4) 测试定位、照相机服务时，需要采用真机进行测试。

9、 客户端更新

- 1) 当客户端有新版本时，有更新提示。
- 2) 当版本为非强制升级版时，用户可以取消更新，老版本能正常使用。用户在下次启动 app 时，仍能出现更新提示。
- 3) 当版本为强制升级版时，当给出强制更新后用户没有做更新时，退出客户端。下次启动 app 时，仍出现强制升级提示。
- 4) 当客户端有新版本时，在本地不删除客户端的情况下，直接更新检查是否能正常更新。
- 5) 当客户端有新版本时，在本地不删除客户端的情况下，检查更新后的客户端功能是否是新版本。
- 6) 当客户端有新版本时，在本地不删除客户端的情况下，检查资源同名文件如图片是否能正常更新成最新版本。如果以上无法更新成功的，也都属于缺陷。

10、 CHECK CODE

客户端在拿图片的时候，一般会将其缓存到本地，减少每次请求服务时所耗的流量。但是有一种图片是不能带缓存的方式存取的，这个就是checkcode。如果开发使用了带缓存方式的方法去取checkcode的图片,就会造成验证码错误的情况。因为check code图片的链接是一样的，但是每次这个链接请求到的数据是不一样的，所以这块测试要额外注意。

11、 其它比较常见的通用内容

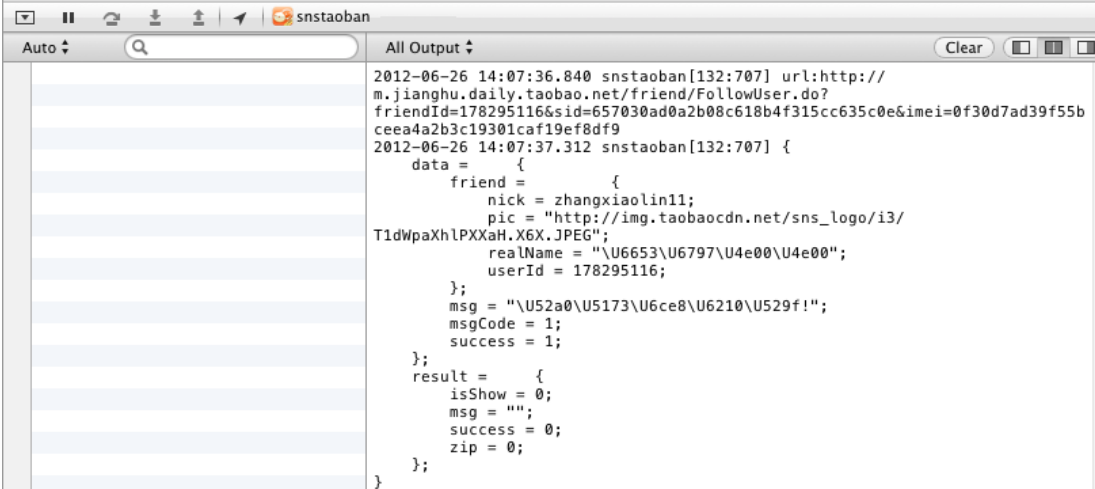
- 1) iphone3 比 iphone4 像素低一半，Iphone3 为 320*480,iphon4 为 640*960。图片需要分成 2 套来开发，检查应用资源图片展示有无模糊。
- 2) 当 app 支持某版本及以上时,需要对相应的版本进行兼容性测试

- 3) 对模拟键盘的处理，例如键盘展开后，点击其他位置能否正常收起，键盘使用完成后能否正常收起。
- 4) 同时或者快速点击不同的两个按键，检查程序是否正常，此问题经常会出现 crash，或者出现两个功能界面并存的情况。
- 5) 较快速度点击同一按钮多次，检查程序是否正常，一般情况下需要对按钮做置灰处理，只允许操作一次，否则可能会产生重复数据。
- 6) 文字含特殊符号的展示显示能正常输出，不转义显示，如<>不会显示成<>
- 7) 考虑界面的完整性，在 ios 界面数据显示宽度上，我们要考虑是自适应，还是自动换行，当自适应的时候，ios 程序会在显示不全的时候自动显示...，此时，就要考虑哪些内容是可以...，哪些内容是必须要完整显示的。
- 8) 应用在有数据自动更新时，必须要有超时时间，如果超过指定时间还没有更新成功，要给予提示，而不能一直进行数据交换，更不能影响 用户的正常使用。
- 9) 多线程情况处理。多线程同时操作一个数据库的问题，经常会引起 app 崩溃。因此，当数据交换存在多线程情况处理时，此块要进行反复测试。
- 10) 对于用户的密码、通讯录等涉及隐私的数据，需要进行加密处理。
- 11) 手机端和 PC 端业务关联的问题，如果在手机端上特有功能注意在 PC 端上的处理，如果在 PC 端上特有的功能，在手机端上也要做相应的容错处理。
- 12) 字体，颜色，视觉搭配的感观测试也是很重要的一点，如果你感觉看上去很模糊，或者说看着很累就说明设计上肯定是存在一定问题的，快点找视觉大师们整改吧。

问题排查

我们在客户端测试时，经常会碰到程序 crash,有的是可以重现出现，有的莫名其妙的闪退可能我们不知道原因无法重现。这时候我们需要利用日志这个武器来帮助分析问题原因的所在。

1、连上 XCODE 运行，打开 debug 区查看日志输出情况。



```
2012-06-26 14:07:36.840 snstaoban[132:707] url:http://m.jianghu.daily.taobao.net/friend/FollowUser.do?friendId=178295116&sid=657030ad0a2b08c618b4f315cc635c0e&imei=0f30d7ad39f55bceea4a2b3c19301caf19ef8df9
2012-06-26 14:07:37.312 snstaoban[132:707] {
  data = {
    friend = {
      nick = zhangxiaolin11;
      pic = "http://img.taobaocdn.net/sns_logo/i3/T1dWpaXh1PXXaH.X6X.JPEG";
      realName = "\U6653\U6797\U4e00\U4e00";
      userId = 178295116;
    };
    msg = "\U52a0\U5173\U6ce8\U6210\U529f!";
    msgCode = 1;
    success = 1;
  };
  result = {
    isShow = 0;
    msg = "";
    success = 0;
    zip = 0;
  };
}
```

当测试发现 bug 时，我们可以自己先做一个简单的排查。反复操作出现 BUG 的功能，查看日志输出内容：

- 1) 看 debug 输出区里的日志输出内容中是否有错误
 - 2) 将 debug 中请求服务器的 URL 拷出来在浏览器上访问一下看服务端返回的数据是否正确
- 2、如果在真机上测试时遇到有 crash，可以将真机的 crash 文件导出来提供给开发，帮助开发排查问题。

