

```
默认 13:19:20.000000 +0800 PYGiOSReBookApp [from pygxpc]message:
<dictionary: 0x14d536ee0> { count = 2, contents =
    "ErrCode" => <int64: 0x14d535880>: 0
    "Status" => <string: 0x14d536fa0> { length = 7, contents =
"Success" }
}
```

XPC 的目的是提高应用的安全性和稳定性，它让进程通信变得更容易，但是细心的读者可能会发现，该方法仅限于系统级别的应用，如果用普通应用就是不可达的，这就引出了下面要讲的 **RocketBootstrap** 通信。

8.6.3 RocketBootstrap 通信

在私有框架 **AppSupport** 中存在一个 **CPDistributedMessagingCenter**（分布式消息传送中心），它使用简单的消息和字典来提供不同进程之间的通信。**RocketBootstrap** 就是在其上进行封装，越过沙盒限制，给越狱环境提供稳定的进程通信服务。在 Cydia 中搜索安装即可。

RocketBootstrap 要求在权限比较高的进程中启动一个服务，再通过发送通知的方式越过沙盒限制，因此先创建一个 **tweak** 并注入 **SpringBoard** 进程，代码如下：

```
#import <rocketbootstrap/rocketbootstrap.h>
#import <AppSupport/CPDistributedMessagingCenter.h>

%group HookSpringBoard
%hook SpringBoard

- (void)applicationDidFinishLaunching:(id)application {
    %orig;
    CPDistributedMessagingCenter *c = [%c(CPDistributedMessagingCenter)
centerNamed:@"com.chinapyg.pygipc"];
    rocketbootstrap_distributedmessagingcenter_apply(c);
    [c runServerOnCurrentThread];
    [c registerForMessageName:@"PYGiOSReBookMessage" target:self
selector:@selector(handleMessage:withUserInfo:)];
    NSLog(@"注册监听");
}

%new

- (void)handleMessage:(NSString *)name withUserInfo:(NSDictionary
*)userInfo {
    NSLog(@"handleMessage withUserInfo:%@", userInfo);

    NSString *action = userInfo[@"action"];
```