

# iOS内购编程指南



little\_ma (/u/fbaf32529c7a) ✓ 已关注

2018.01.07 21:05\* 字数 2595 阅读 842 评论 2 喜欢 1

(/u/fbaf32529c7a)

一般来说，开发人员刚接触内购，都会遇到流程不清楚、千头万绪。  
如何一次性搞定内购问题？

## 一、掌握内购流程：

### 1、完成前期准备工作

1)、接手内购，一定要阅读苹果的《APP内购买项目》(<https://link.jianshu.com?t=https%3A%2F%2Fhelp.apple.com%2Fitunes-connect%2Fdeveloper%2F%23%2Fdevb57be10e7>)文档

2)、iTunesconnect后台配置内购项目

a、未配置过内购的新项目，签署 Paid Applications agreement（《付费应用程序协议》）





424da0e1-d0e4-4c92-9525-e853a243fa77.png

b、配置内购项目，“App -->功能-->APP内购买项目”  
参考：创建及发布说明 (<https://link.jianshu.com?t=https%3A%2F%2Fhelp.apple.com%2Fitunes-connect%2Fdeveloper%2F%23%2Fdevae49fb316>)

3)、Xcode工程配置



da468add-ce28-4140-86bd-6aa27c3d40be.png

开启此选项App Store中APP的介绍界面显示内购的相关项目，关闭则不显示

4)、开发实现流程：





296873ed-be51-4435-8ddc-d8391fd9b56f.jpg

## 2、iTunesconnect创建产品

苹果的内购分以下四类商品：

### 1、消耗型项目

只可使用一次的产品，使用之后即失效，必须再次购买。

示例：钓鱼 App 中的鱼食。

### 2、非消耗型项目

只需购买一次，不会过期或随着使用而减少的产品。

示例：游戏 App 的赛道。

### 3、自动续期订阅

允许用户在固定时间段内购买动态内容的产品。除非用户选择取消，否则此类订阅会自动续期。

示例：每月订阅提供流媒体服务的 App。

### 4、非续期订阅

允许用户购买有时限性服务的产品。此 App 内购买项目的内容可以是静态的。此类订阅不会自动续期。

示例：为期一年的已归档文章目录订阅。

首先，在“我的APP”——“功能”——“App内购买项目”添加适合自己的商品：



3CAFC87E-7FFA-43C3-8370-0E3A28329B14.png

## 二、开发实现：

每个开发人员帐户可在该帐户的所有 App 中创建最多 10,000 个 App 内购买项目产品。App 内购买项目共有四种类型：消耗型、非消耗型、自动续期订阅和非续期订阅。



开源库 (XYIAPKit (<https://link.jianshu.com?t=https%3A%2F%2Fgithub.com%2Fmqc123450%2FXYIAPKit>)) , 使用此开源库直接

可省去“开发实现”这一步骤。

## 实现步骤主要包括三步：

- 1、首先在项目工程中加入StoreKit.framework
- 2、加入头文件#import <StoreKit/StoreKit.h>
- 3、遵守代理SKPaymentTransactionObserver,SKProductsRequestDelegate

### 步骤一：App Store请求内购项

注意：此步骤建议在开始创建购买订单前完成，这样可以减少购买时查询订单的时间

- 1)、判断用户是否具备支付权限

```
if ([SKPaymentQueue canMakePayments]) {  
    //允许应用内付费购买  
}else {  
    //用户禁止应用内付费购买。  
}
```

Indicates whether the user is allowed to make payments.  
An iPhone can be restricted from accessing the Apple App Store. For example, parents can restrict their children's ability to purchase additional content. Your application should confirm that the user is allowed to authorize payments before adding a payment to the queue. Your application may also want to alter its behavior or appearance when the user is not allowed to authorize payments.

- 2、创建一个商品查询的请求， `productIdentifiers` 指需要查询的“产品ID”的数组



```
NSSet * set = [NSSet arrayWithArray:productIdentifiers];
SKProductsRequest * request = [[SKProductsRequest alloc] initWithProductIdentifiers:set];
request.delegate = self;
[request start];
```

查询的结果将通过 SKProductsRequestDelegate 得到查询的结果  
获取商品的查询结果

```
- (void)productsRequest:(SKProductsRequest *)request didReceiveResponse:(SKProductsResponse *)response
```

在此回调中我们可以拿到所查询产品的信息，保存所查询的信息，以供构建购买时使用。

```
SKProductsResponse

// Array of SKProduct instances.
@property(nonatomic, readonly) NSArray<SKProduct *> *products NS_AVAILABLE_IOS(3_0);

// Array of invalid product identifiers.
@property(nonatomic, readonly) NSArray<NSString *> *invalidProductIdentifiers NS_AVAILABLE_IOS(3_0);
```

获取请求完成和失败的结果

```
- (void)requestDidFinish:(SKRequest *)request NS_AVAILABLE_IOS(3_0);
- (void)request:(SKRequest *)request didFailWithError:(NSError *)error NS_AVAILABLE_IOS(3_0);
```

## 步骤二：开始构建购买请求

### 1)、判断请求的产品ID是否已获取到它的产品信息 SKProduct，且是否可用。

若商品可用创建支付；

若商品不可用提示用户；

若商品尚未获取它的产品信息，进行“步骤一”操作

### 2)、创建支付



```
SKPayment * payment = [SKPayment paymentWithProduct:product];  
[[SKPaymentQueue defaultQueue] addPayment:payment];
```

### 3)、添加支付交易的Observer

```
[[SKPaymentQueue defaultQueue] addTransactionObserver:self];
```

注意在适当的时候移除Observer

```
[[SKPaymentQueue defaultQueue] removeTransactionObserver:self];
```

可以通过遵循 SKPaymentTransactionObserver 协议来监听整个交易的过程

交易状态发生改变时，包括状态的改变，交易的结束

```
- (void)paymentQueue:(SKPaymentQueue *)queue updatedTransactions:(NSArray<SKPaymentTransaction> *)transactions {  
    switch (transaction.transactionState) {  
        case SKPaymentTransactionStatePurchased://交易完成  
            [self completeTransaction:transaction];  
            break;  
        case SKPaymentTransactionStateFailed://交易失败  
            [self failedTransaction:transaction];  
            break;  
        case SKPaymentTransactionStateRestored://已经购买过该商品  
            [self restoreTransaction:transaction];  
            break;  
        case SKPaymentTransactionStatePurchasing: //商品添加进列表  
            NSLog(@"商品添加进列表");  
            break;  
        default:  
            break;  
    }  
}
```

### 4)、校验票据 ([https://link.jianshu.com?](https://link.jianshu.com?t=https%3A%2F%2Fdeveloper.apple.com%2Flibrary%2Fcontent%2Fgeneral%2Fvalidate-app-store-receipt%2Fchapters%2Freceipt-validation)

[t=https%3A%2F%2Fdeveloper.apple.com%2Flibrary%2Fcontent%2Fgeneral%2Fvalidate-app-store-receipt%2Fchapters%2Freceipt-validation](https://link.jianshu.com?t=https%3A%2F%2Fdeveloper.apple.com%2Flibrary%2Fcontent%2Fgeneral%2Fvalidate-app-store-receipt%2Fchapters%2Freceipt-validation))





## eiptFields.html%23%2F%2Fapple\_ref%2Fdoc%2Fuid%2FTP40010573-CH106-SW1)

票据的校验是保证内购安全完成的非常关键的一步，一般有三种方式：

- 1、服务器验证，获取票据信息后上传至信任的服务器，由服务器完成与App Store的验证（提倡使用此方法，比较安全）
- 2、本地票据校验
- 3、本地App Store请求验证

### a)、本地票据校验

本地票据校验的一般步骤

要验证收据，请按顺序执行以下测试：

1.找到收据。

如果没有收据，则验证失败。

2.验证收据是否由Apple 正确签署

如果收据不是由 Apple 签署，则验证失败。

3.验证收据中的Bundle Identifier（数据包标识符）与在Info.plist文件中含有您要的CFBundleIdentifier值的硬编码常量相匹配。

如果两者不匹配，则验证失败。

4.验证收据中的版本标识符字符串与在Info.plist文件中含有您要的CFBundleShortVersionString值（macOS）或CFBundleVersion 值（iOS）的硬编码常量相匹配。

如果两者不匹配，则验证失败。

5.按照“计算 GUID 的哈希（Hash）（第 8 页）”所述计算GUID 的哈希（Hash）。

如果结果与收据中的哈希（Hash）不匹配，则验证失败。

如果通过所有测试，则验证成功。

注意: Bundle Identifier（数据包标识符）和版本标识符字符串是UTF-8 字符串，而不仅仅是一系列字节。确保您相应地编写比较逻辑代码。



如果您的 App 支持“批量购买计划”，请检查收据的有效日期。

## b)、App Store验证：

### 1)、获取票据

```
- (NSString *)iapReceipt
{
    NSString *receiptString = nil;
    NSURL *rereceiptURL = [[NSBundle mainBundle] appStoreReceiptURL];
    if ([[NSFileManager defaultManager] fileExistsAtPath:[rereceiptURL path]]) {
        NSData *receiptData = [NSData dataWithContentsOfURL:rereceiptURL];
        receiptString = [receiptData base64EncodedStringWithOptions:0];
    }

    return receiptString;
}
```

如果此时未获取到票据的信息，使用 `SKReceiptRefreshRequest` 来刷新票据结果。

```
SKReceiptRefreshRequest *refreshReceiptRequest = [[SKReceiptRefreshRequest alloc] init];
refreshReceiptRequest.delegate = self;
[refreshReceiptRequest start];
```

```
- (void)requestDidFinish:(SKRequest *)request {
}
```

### 2)、请求验证

获取到票据以后我们通过App Store来验证票据是否真实

沙盒状态下使用：<https://sandbox.itunes.apple.com/verifyReceipt>

(<https://link.jianshu.com?t=https%3A%2F%2Fsandbox.itunes.apple.com%2FverifyReceipt>)来验证

生产环境下使用：<https://buy.itunes.apple.com/verifyReceipt> (<https://link.jianshu.com?t=https%3A%2F%2Fbuy.itunes.apple.com%2FverifyReceipt>)

生产环境下使用：<https://buy.itunes.apple.com/verifyReceipt> (<https://link.jianshu.com?t=https%3A%2F%2Fbuy.itunes.apple.com%2FverifyReceipt>)

生产环境下使用：<https://buy.itunes.apple.com/verifyReceipt> (<https://link.jianshu.com?t=https%3A%2F%2Fbuy.itunes.apple.com%2FverifyReceipt>)

常见的验证状态代码：



6861C64F-2A45-4EF5-8D42-6A07C6B174A8.png

---



由于审核时，审核人员为沙盒状态，注意状态码“21007”

```

- (void)verifyRequestData:(NSString *)base64Data
    url:(NSString *)url
    transaction:(SKPaymentTransaction *)transaction
    success:(void (^)(void))successBlock
    failure:(void (^)(NSError *error))failureBlock
{
    NSMutableDictionary *params = [NSMutableDictionary dictionary];
    [params setValue:base64Data forKey:@"receipt-data"];
    [params setValue:self.sharedSecretKey forKey:@"password"];

    NSError *jsonError;
    NSData *jsonData = [NSJSONSerialization dataWithJSONObject:params
                                                    options:NSJSONWritingPrettyPrinted
                                                    error:&jsonError];

    if (jsonError) {
        NSLog(@"verifyRequestData failed: error = %@", jsonError);
    }

    NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:[NSURL URLWithString:url]
    request.HTTPBody = jsonData;
    static NSString *requestMethod = @"POST";
    request.HTTPMethod = requestMethod;

    __weak typeof(self) weakSelf = self;
    dispatch_async(dispatch_get_global_queue( DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
        NSError *error;
        NSData *data = [NSURLConnection sendSynchronousRequest:request returningResponse:&error
        dispatch_async(dispatch_get_main_queue(), ^{

            if (!data) {
                NSError *wrapperError = [weakSelf unableVerifyReceiptError:error];
                if (failureBlock != nil) failureBlock(wrapperError);
                return;
            }

            NSError *jsonError;
            NSDictionary *responseJSON = [NSJSONSerialization JSONObjectWithData:data
            if (!responseJSON) {
                NSLog(@"Failed To Parse Server Response");
                if (failureBlock != nil) failureBlock(jsonError);
            }

            static NSString *statusKey = @"status";
            NSInteger statusCode = [responseJSON[statusKey] integerValue];

            static NSInteger successCode = 0;
            static NSInteger sandboxCode = 21007;

```

```
        if (statusCode == successCode) {
            [weakSelf saveVerifiedReceipts:transaction response:responseJSON];
            if (successBlock != nil) successBlock();
        } else if (statusCode == sandboxCode) {
            [weakSelf sandboxVerify:base64Data
                                transaction:transaction
                                success:successBlock
                                failure:failureBlock];
        } else {
            NSLog(@"Verification Failed With Code %ld", (long)statusCode);
            NSError *serverError = [NSError errorWithDomain:XYStoreErrorDomain code:statusCode userInfo:nil];
            if (failureBlock != nil) failureBlock(serverError);
        }
    }
});
}
```

### 3)、核实票据信息

- a、非订阅产品，若返回结果正确，验证通过
- b、订阅产品需要验证订阅是否过期



```
- (BOOL)checkIsSubscribed:(XYiTunesResponse *)iTunesResponse productId:(NSString *)productId {
{
    NSDate *expires_date;
    for (XYInAppReceipt *appReceipt in iTunesResponse.latest_receipt_info) {

        if ([appReceipt.product_id isEqualToString:productId] == NO) {
            continue;
        }

        if (expires_date) {
            expires_date = [expires_date laterDate:appReceipt.expires_date];
        } else {
            expires_date = appReceipt.expires_date;
        }
    }

    // 不包含过期信息表示无自动续期交易
    if (!expires_date) {
        return NO;
    }

    if ([expires_date timeIntervalSinceDate:iTunesResponse.receipt.request_date] > 0) {
        // 1、对比请求时间
        // 针对SKPaymentTransactionObserver的监听，当交易信息发生更新时，苹果会自动推送当前的
        // 缓存票据更新时的请求时间，通过与过期时间对比来确定用户的订阅是否过期
        // 此方式可以避免用户修改系统时间造成的问题，也能保证及时的更新用户订阅状况
        // 也可使用获取当前外部服务器的时间，当然需要异步操作，时间成本比较高

        // 2、对比系统时间
        // 防止订阅后，用户强制断网

        return YES;
    }

    return NO;
}
}
```



```

@interface XYInAppReceipt : NSObject

/**
 The default value is 1, the minimum value is 1, and the maximum value is 10.
 */
@property (nonatomic, assign) NSInteger quantity;

@property (nonatomic, copy) NSString *product_id;

@property (nonatomic, copy) NSString *transaction_id;

@property (nonatomic, copy) NSString *original_transaction_id;

@property (nonatomic, strong) NSDate *purchase_date;

@property (nonatomic, strong) NSDate *original_purchase_date;

/**
 仅用于，自动续费订阅
 true，表示处于 免费试用 时期
 如果已有票据中含有is_trial_period或者is_in_intro_offer_period为true，用户不再具备此项资格

 For a subscription, whether or not it is in the free trial period.
 This key is only present for auto-renewable subscription receipts. The value for th

 Note: If a previous subscription period in the receipt has the value “true” for eit
 */
@property (nonatomic, assign) BOOL is_trial_period;

//*****以上为四种内购类型公共字段，下面字段为自动续期订阅独有字段*****

/**
 This key is only present for auto-renewable subscription receipts.
 Use this value to identify the date when the subscription will renew or expire, to
 After validating the latest receipt, if the subscription expiration date for the la

 */
@property (nonatomic, strong) NSDate *expires_date;

/**
 “1” - Customer canceled their subscription.
 “2” - Billing error; for example customer’s payment information was no longer valid
 “3” - Customer did not agree to a recent price increase.
 “4” - Product was not available for purchase at the time of renewal.
 “5” - Unknown error
 */
@property (nonatomic, assign) NSInteger expiration_intent;

/**

```



```

对于订阅过期的自动续费产品，苹果是否会尝试自动续费
For an expired subscription, whether or not Apple is still attempting to automatically renew the subscription.
“1” – App Store is still attempting to renew the subscription.
“0” – App Store has stopped attempting to renew the subscription.

This key is only present for auto-renewable subscription receipts. If the customer's subscription has expired, this key is not present.
*/
@property (nonatomic, assign) BOOL is_in_billing_retry_period;

/**
仅用于，自动续费订阅
true, 表示处于 引导价格 时期
如果已有票据中含有is_trial_period或者is_in_intro_offer_period为true, 用户不再具备有此项资格

For an auto-renewable subscription, whether or not it is in the introductory price period.
This key is only present for auto-renewable subscription receipts. The value for this key is “true” if the subscription is in the introductory price period.

Note: If a previous subscription period in the receipt has the value “true” for either is_trial_period or is_in_intro_offer_period, then this key is not present.
*/
@property (nonatomic, assign) BOOL is_in_intro_offer_period;

/**
退款操作时间
For a transaction that was canceled by Apple customer support, the time and date of the cancellation.
Treat a canceled receipt the same as if no purchase had ever been made.
A canceled in-app purchase remains in the receipt indefinitely. Only applicable if the transaction is in the receipt.
*/
@property (nonatomic, strong) NSDate *cancellation_date;

/**
内购取消的原因
“1” – Customer canceled their transaction due to an actual or perceived issue with the product.
“0” – Transaction was canceled for another reason, for example, if the customer made a purchase error.

Use this value along with the cancellation date to identify possible issues in your app.
*/
@property (nonatomic, copy) NSString *cancellation_reason;

/**
APP唯一标识符
*/
@property (nonatomic, copy) NSString *app_item_id;

/**
This key is not present for receipts created in the test environment. Use this value to identify receipts created in the test environment.
*/
@property (nonatomic, copy) NSString *version_external_identifier;

```





```
/**
 This value is a unique ID that identifies purchase events across devices, including
 */
@property (nonatomic, copy) NSString *web_order_line_item_id;

/**
 The current renewal status for the auto-renewable subscription.
 "1" - Subscription will renew at the end of the current subscription period.

 "0" - Customer has turned off automatic renewal for their subscription.

 This key is only present for auto-renewable subscription receipts, for active or expiring
 */
@property (nonatomic, assign) NSInteger auto_renew_status;

/**
 The current renewal preference for the auto-renewable subscription.
 This key is only present for auto-renewable subscription receipts. The value for the
 */
@property (nonatomic, copy) NSString *auto_renew_product_id;

/**
 The current price consent status for a subscription price increase.
 "1" - Customer has agreed to the price increase. Subscription will renew at the higher price.
 "0" - Customer has not taken action regarding the increased price. Subscription expires at the end of the current period.

 This key is only present for auto-renewable subscription receipts if the subscription price has increased.
 */
@property (nonatomic, assign) NSInteger price_consent_status;
```

## \*自动续期订阅

### 1)、自动续费

购买流程上，自动续费订阅与普通购买没有区别；

主要的区别在于：除了第一次购买行为是用户主动触发的。后续续费都是Apple自动完成的，一般要在要过期的前24小时开始，苹果会尝试扣费，扣费成功的话 会在APP下次启动的时候主动推送给APP。

```
[[SKPaymentQueue defaultQueue] addTransactionObserver:self];
```

添加上面的监听就是非常必要的了。

如何处理首次订阅以及续费订阅？



```
- (void)paymentQueue:(SKPaymentQueue *)queue updatedTransactions:(NSArray *)transactions
{
    for (SKPaymentTransaction *transaction in transactions)
    {
        switch (transaction.transactionState)
        {
            case SKPaymentTransactionStatePurchased://交易完成

                // 订阅特殊处理
                if(transaction.originalTransaction){
                    // 如果是自动续费的订单originalTransaction会有内容
                }else{
                    // 普通购买, 以及 第一次购买 自动订阅
                }
                break;
            case SKPaymentTransactionStateFailed: //交易失败

                break;
            case SKPaymentTransactionStateRestored: //已经购买过该商品

                break;
            case SKPaymentTransactionStatePurchasing: //商品添加进列表

                break;
            default:
                break;
        }
    }
}
```

**关键点：**上面代码可以看到，通过 `transaction.originalTransaction` 来判断是否为续费订单，通过这个知识，对之后的支付统计及付费跟踪非常关键。

2)、推介促销价（目前比较流行的套路）



f415317a-a6f4-4cc4-8c27-7031fdb78e86.png

主要分：随用随付、随用随付、免费试用

#### a、随用随付

如果您选择“随用随付”，则顾客将为选定时限的每个结算周期支付推介促销价（例如，订阅的标准价格为 9.99 美元，推介促销价为前 3 个月每月 1.99 美元）。可设定以下时限：

1 周订阅，1 至 12 周

1 个月订阅，1 至 12 个月

2 个月订阅，2、4、6、8、10 和 12 个月

3 个月订阅，3、6、9 和 12 个月

6 个月订阅，6 和 12 个月

1 年订阅，1 年

#### b、提前支付

如果您选择“提前支付”，顾客将一次性支付选定时限的推介促销价（例如，订阅的标准价格为 9.99 美元，推介促销价为前 2 个月 1.99 美元）。可设定以下时限：1 个月、2 个月、3 个月、6 个月或 1 年。

#### c、免费试用



如果您选择“免费试用”，则顾客在选定的时限内免费访问订阅。时限可以是 3 天、1 周、2 周、1 个月、2 个月、3 个月、6 个月或 1 年。

推介促销价是为回馈用户的一个促销方式，Apple推出后很受APP开发的欢迎。

当然也是非常流行的套路，我们了解一下这个地方套路的核心问题：

拿免费试用来说，用户一旦点击了免费试用后，免费试用结束后，苹果将自动进行扣款；

那么如果试用后不想购买而想退款的话，找到取消的地方就需要靠智商了，取消免费试用的路径隐藏的非常深。





5322eec9-ff3e-4962-a908-51e56fce0a42.png

---









0b6e907e-a101-4290-9d5a-430883d75601.png

---



0b6e907e-a101-4290-9d5a-430883d75601.png

---

+

🔖

🔗



39605e26-7ca0-471d-9d4e-85524a638996.png

---



cc407dec-efdd-4bde-878e-795e3eceabd3.png

---

## 四、内购的测试问题

1) 、沙盒测试：非正式环境下测试需要添加沙盒测试员



您可以使用沙箱技术测试您的 App 和 App 内购买项目，而无需创建财务交易。沙箱技术是一个使用 App Store 基础架构但不处理实际付款的测试环境。它会返回交易，付款被视为已成功处理。

添加方式：

64e162fc-cb43-4c7c-84e5-32bef7c7b489.png

自动续费测试时需注意：

11436ef0-9cfe-40c4-bd2c-fe4aa2d1e5fd.png

意思就是，沙箱环境 自动续费时间缩短了，一周 对应 三分钟，一月 对应 五分钟。。。购买完一个一周 类型订阅，就不要在APP不退出的情况等待了，必须3分钟 或是 10分钟后重新登录，Apple才会主动告知你结果，也就是第一点提到的。



沙箱环境自动续费是一定会自动续费的吗？

不一定的，有时候会，有时候不会。所以要多测测，多建几个测试账号。

## 五、审核问题

添加自动订阅，必须在订阅路径内需包含以下信息：

- 1)、隐私协议及关于App的内容
- 2)、暴露自动续费的信息

- Information about the auto-renewable nature of the subscription
- Links to the privacy policy and terms of use

小礼物走一走，来简书关注我

赞赏支持

📖 已发布 (/nb/19973454)

举报文章 © 著作权归作者所有



little\_ma (/u/fbaf32529c7a)

写了 7745 字，被 4 人关注，获得了 2 个喜欢

(/u/fbaf32529c7a)

✓ 已关注

喜欢 | 1



更多分享


(http://cwb.assets.jianshu.io/notes/images/22085996/weibo/image\_!

+



被以下专题收入，发现更多相似内容

+ 收入我的专题

 ios基础 (/c/32649906136a?utm\_source=desktop&utm\_medium=notes-included-collection)


(/p/cd3582e28ddb?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

**iOS 内购最新讲解 (/p/cd3582e28ddb?utm\_campaign=maleskine&utm\_c...**

一.总说内购的内容 协议、税务和银行业务 信息填写 内购商品的添加 添加沙盒测试账号 内购代码的具体实现  
内购的注意事项 二.协议、税务和银行业务 信息填写 2.1、协议、税务和银行业务 信息填写 的入口 协议、...

 默默\_David (/u/9ff76bcf7db5?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/4f8a854ff427?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

**iOS 内购的最新讲解(审核上线的坑) (/p/4f8a854ff427?utm\_campaign=mal...**

一.总说内购的内容 1.协议、税务和银行业务 信息填写 2.内购商品的添加 3.添加沙盒测试账号 4.内购代码的具体实现 5.内购的注意事项 6.掉单的处理(这个是一个蛋疼的事情) 二.协议、税务和银行业务 信息填写 2.1...

 IronMan (/u/8fed18ed70c9?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/a607f051b3b6?


utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

**iOS 内购最新讲解 (/p/a607f051b3b6?utm\_campaign=maleskine&utm\_c...**

+  
🔖  
🔗



一.总说内购的内容 协议、税务和银行业务 信息填写 内购商品的添加 添加沙盒测试账号 内购代码的具体实现 内购的注意事项 二.协议、税务和银行业务 信...


 任如运 (/u/58a31b6bca47?)



utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

iOS应用内付费(IAP)开发步骤 (/p/e9e4dc3dc9ee?utm\_campaign=malesk...


iOS应用内付费(IAP)开发步骤 1.苹果iTunes Connect内购产品信息录入。 1) 创建app内购买项目 (Create New) , 选择类型: 1.消耗型项目 对于消耗型 App 内购买项目, 用户每次下载时都必须进行购买。一次...

 MillerWang (/u/a40fecbc4982?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

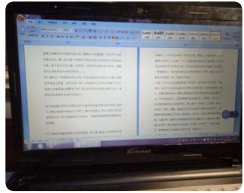
说说IOS内购那些事 (/p/6477e9f6ac43?utm\_campaign=maleskine&utm\_...

1.支付类和协议 Store Kit框架下提供用来处理应用内购大致的类可以分为两部分: 1.请求商品, 2.购买商品。 具体如下: SKMutablePayment: 该类为App Store定义了用户的付费请求功能, 通过封装特定产品名称和...

 萧子然 (/u/4473b5adb922?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/b7309d89e6ea?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

第一次论坛, 第一次导演 (/p/b7309d89e6ea?utm\_campaign=maleskine&...

我是一名来自芜湖荃萃中学初二七班的学生, 前几天在老师和同学的推举下我迎来了人生的第一次论坛导演机会 其实说真的, 刚刚听到记者论坛, 这个, 我的表情是这样的 但是在同学们的鼓励之下我还是硬着头皮...

 乐如鱼 (/u/9b92725a2770?)

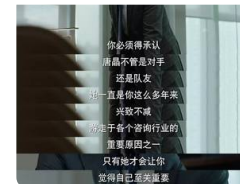
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/02b84adfb875?

+








utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 《我的前半生》讲了哪些人艰不拆的真相 (/p/02b84adfb875?utm\_campai...


在上海开往北京的夜班火车上，听着同车厢大叔此起彼伏的呼噜声，看完了《我的前半生》，泪流满面。这部剧确实讨论了很多关于“人”不可回避的矛盾：譬如用什么来维系婚姻，为难以长久的爱情保鲜；譬如如...

 小煦说 (/u/3fd903aa2ec3?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 171130-大交易-可儿朵朵-任务一 (/p/cf60d2e44747?utm\_campaign=male...


早知道该好好取个名字的，不过人生哪有那么多早知道哈 1、请做简要自我介绍 外企Marketing。2、上一本阅读的书、下一本想要阅读的书以及理由 上一本阅读的书《全新思维》，正在阅读的是《影响力》，下一...

 可儿朵朵 (/u/3ab38bdecbca?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 原来你是这样的东野圭吾 (/p/683933f027d8?utm\_campaign=maleskine&...

这些天很忙，但断断续续的也看完了日本著名推理小说家东野圭吾的半自传《我的晃荡的青春》。之所以说是半自传，是因为这本书只着重讲述了他从小学到大学这段时间的经历。有人说传记对人的最大吸引在于...

 fan林凡 (/u/8fecfcef6ba9?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/88b094bc99df?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 这样说话的人，不要深交！ (/p/88b094bc99df?utm\_campaign=maleskin...

这样说话的人，不要深交..... 话语展现一个人的性情，言谈展现一个人的品格。你身边有那些说话不顾及人感受的人吗？很多时候，我们会善良地以为，一个人说话难听只是因为他心直口快，一个人满嘴负能量只...





巴蜀云雨 (/u/1531459e354a?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

