Ios

实现iOS应用内购买的三个核心步骤

时间 2014-02-23 16:40:09 [CSDN博客](http://www.tuicool.com/sites/vQZRB3)

原文

[http://blog.csdn.net/nimingzhe2008/article/details/19759589](http://blog.csdn.net/nimingzhe2008/article/details/19759589?utm_source=tuicool&utm_medium=referral)

主题 [C语言](http://www.tuicool.com/topics/11010000) [iOS开发](http://www.tuicool.com/topics/11080009)

要把大象装冰箱，总共分三步——宋丹丹

这篇文章不是一个应用内购买的完整教程，而是讲述了内购买的核心编程步骤，以一个清晰简洁的视角，帮助读者理解应用内购买。编写内购买代码之前，需要在iTunes Connect中创建商品，方法请自行google。现在假设我们在iTunes Connect中创建了一个商品。

**第一步，要从苹果服务器获得商品信息。**

你说“我知道自己的商品信息”，不好意思，此步骤是必须的，后面会用到代表商品的变量。

if ([SKPaymentQueue canMakePayments])

{

SKProductsRequest \*request = [[SKProductsRequest alloc]

initWithProductIdentifiers:

[NSSet setWithObject:productID]];//productID是一个NSString变量，可从iTunes connect查询商品来获得

request.delegate = self;

[request start];

}

上面的代码创建并发送了一个获得商品信息的请求。苹果的服务器返回商品信息时，会触发代理方法。

#pragma mark -

#pragma mark SKProductsRequestDelegate

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

{

NSArray \*products = response.products;

if (products.count != 0)

{

\_product = products[0];//\_product是一个SKProduct变量

} else {

NSLog(@"Product not found");

}

products = response.invalidProductIdentifiers;

for (SKProduct \*product in products)

{

NSLog(@"Product not found: %@", product);

}

我们在代理方法中将商品信息放入了一个SKProduct变量\_product。

**第二步，创建并发送支付请求。**

首先，要为交易队列找一个观察者。

[[SKPaymentQueue defaultQueue]

addTransactionObserver:self];

然后创建一个交易，并将其加入交易队列。

SKPayment \*payment = [SKPayment paymentWithProduct:\_product];

[[SKPaymentQueue defaultQueue] addPayment:payment];

**第三部，处理交易结果。**

#pragma mark -

#pragma mark SKPaymentTransactionObserver

-(void)paymentQueue:(SKPaymentQueue \*)queue updatedTransactions:(NSArray \*)transactions

{

for (SKPaymentTransaction \*transaction in transactions)

{

switch (transaction.transactionState) {

case SKPaymentTransactionStatePurchased:

//购买成功，在这写入后续代码

[[SKPaymentQueue defaultQueue]

finishTransaction:transaction];

break;

case SKPaymentTransactionStateFailed:

//购买失败，在这写入后续代码

[[SKPaymentQueue defaultQueue]

finishTransaction:transaction];

break;

case SKPaymentTransactionStateRestored:

{

//在这写入恢复内购买的代码

[[SKPaymentQueue defaultQueue]

finishTransaction:transaction];

break;

}

default:

break;

}

}

}

当苹果的服务器处理完交易请求后，就会将交易结果发回来，并调用上面的方法。

内购买的基本步骤到此结束。虽然还有许多事要做，比如保存购买结果，恢复购买结果等等。这些内容请查阅 [***苹果官方文档***](https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/StoreKitGuide/Introduction.html) 。当你理解了基本步骤后，就如同盖房子打好了地基。后面可能还会遇到各种问题，但你都会循序渐进的将它们解决。

#import <UIKit/UIKit.h>

#import <StoreKit/StoreKit.h>

@interface ViewController : UIViewController<

SKProductsRequestDelegate,SKPaymentTransactionObserver>

{

SKProductsRequest \*productsRequest;

NSArray \*validProducts;

UIActivityIndicatorView \*activityIndicatorView;

IBOutlet UILabel \*productTitleLabel;

IBOutlet UILabel \*productDescriptionLabel;

IBOutlet UILabel \*productPriceLabel;

IBOutlet UIButton \*purchaseButton;

}

- (void)fetchAvailableProducts;

- (BOOL)canMakePurchases;

- (void)purchaseMyProduct:(SKProduct\*)product;

- (IBAction)purchase:(id)sender;

@end

11.更新ViewController.m ，如下所示

#import "ViewController.h"

#define kTutorialPointProductID

@"com.tutorialPoints.testApp.testProduct"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad

{

[super viewDidLoad];

// Adding activity indicator

activityIndicatorView = [[UIActivityIndicatorView alloc]

initWithActivityIndicatorStyle:UIActivityIndicatorViewStyleWhiteLarge];

activityIndicatorView.center = self.view.center;

[activityIndicatorView hidesWhenStopped];

[self.view addSubview:activityIndicatorView];

[activityIndicatorView startAnimating];

//Hide purchase button initially

purchaseButton.hidden = YES;

[self fetchAvailableProducts];

}

- (void)didReceiveMemoryWarning

{

[super didReceiveMemoryWarning];

// Dispose of any resources that can be recreated.

}

-(void)fetchAvailableProducts{

NSSet \*productIdentifiers = [NSSet

setWithObjects:kTutorialPointProductID,nil];

productsRequest = [[SKProductsRequest alloc]

initWithProductIdentifiers:productIdentifiers];

productsRequest.delegate = self;

[productsRequest start];

}

- (BOOL)canMakePurchases

{

return [SKPaymentQueue canMakePayments];

}

- (void)purchaseMyProduct:(SKProduct\*)product{

if ([self canMakePurchases]) {

SKPayment \*payment = [SKPayment paymentWithProduct:product];

[[SKPaymentQueue defaultQueue] addTransactionObserver:self];

[[SKPaymentQueue defaultQueue] addPayment:payment];

}

else{

UIAlertView \*alertView = [[UIAlertView alloc]initWithTitle:

@"Purchases are disabled in your device" message:nil delegate:

self cancelButtonTitle:@"Ok" otherButtonTitles: nil];

[alertView show];

}

}

-(IBAction)purchase:(id)sender{

[self purchaseMyProduct:[validProducts objectAtIndex:0]];

purchaseButton.enabled = NO;

}

#pragma mark StoreKit Delegate

-(void)paymentQueue:(SKPaymentQueue \*)queue

updatedTransactions:(NSArray \*)transactions {

for (SKPaymentTransaction \*transaction in transactions) {

switch (transaction.transactionState) {

case SKPaymentTransactionStatePurchasing:

NSLog(@"Purchasing");

break;

case SKPaymentTransactionStatePurchased:

if ([transaction.payment.productIdentifier

isEqualToString:kTutorialPointProductID]) {

NSLog(@"Purchased ");

UIAlertView \*alertView = [[UIAlertView alloc]initWithTitle:

@"Purchase is completed succesfully" message:nil delegate:

self cancelButtonTitle:@"Ok" otherButtonTitles: nil];

[alertView show];

}

[[SKPaymentQueue defaultQueue] finishTransaction:transaction];

break;

case SKPaymentTransactionStateRestored:

NSLog(@"Restored ");

[[SKPaymentQueue defaultQueue] finishTransaction:transaction];

break;

case SKPaymentTransactionStateFailed:

NSLog(@"Purchase failed ");

break;

default:

break;

}

}

}

-(void)productsRequest:(SKProductsRequest \*)request

didReceiveResponse:(SKProductsResponse \*)response

{

SKProduct \*validProduct = nil;

int count = [response.products count];

if (count>0) {

validProducts = response.products;

validProduct = [response.products objectAtIndex:0];

if ([validProduct.productIdentifier

isEqualToString:kTutorialPointProductID]) {

[productTitleLabel setText:[NSString stringWithFormat:

@"Product Title: %@",validProduct.localizedTitle]];

[productDescriptionLabel setText:[NSString stringWithFormat:

@"Product Desc: %@",validProduct.localizedDescription]];

[productPriceLabel setText:[NSString stringWithFormat:

@"Product Price: %@",validProduct.price]];

}

} else {

UIAlertView \*tmp = [[UIAlertView alloc]

initWithTitle:@"Not Available"

message:@"No products to purchase"

delegate:self

cancelButtonTitle:nil

otherButtonTitles:@"Ok", nil];

[tmp show];

}

[activityIndicatorView stopAnimating];

purchaseButton.hidden = NO;

}

@end

开发者在提交应用进行审核后，总会遇到各种各样的被拒理由，大部分被拒理由可以从苹果的App Store审核指南中找到，不过也有一些比较奇葩的理由，偶尔也会有一些苹果审核人员的人为原因，你曾经被拒的理由，尤其是奇葩理由有哪些呢？欢迎参加我们当前正在进行中的活动，来和大家分享下你的经验和经历。本次【参与送书】活动截止日期为11月16日，活动结束后我们将抽取10名分享经验的用户，这10名用户可获得我们送出《方寸指间—移动设计实战手册》一本。

以下是会员列举出的审核被拒理由，按照回答顺序先后合并重复理由，一起来看看苹果最近有什么新的审核动向吧！如果你也有奇葩的被拒理由，可参与我们的活动进行分享：[【参与送书】App Store ——心酸的上线路，说说那些不可思议的被拒理由吧#涨姿势#](http://www.cocoachina.com/bbs/read.php?tid-238067-page-1.html)

**yoyeayoyea**

您的应用包括色情内容（色情交易，色情展示）。

原因是我们的销售人员，把几张艺术照放在个人相册里（头像），换成卡通头像，通过。

**颜小风**

被拒很正常 一次通过不正常。

之前上线过一些webapp版本，就是个web页面+一个壳，但是客户需要这东西呀，想法设法的上线，后来写了一个伪界面，弄出几个复杂原生页面来，然后增加一个时间判断，等到一个月以后再加载客户需要的界面，没想到这样做居然通过了。

**基调中国**

有微信分享功能，结果因为要强制用户安装微信，才能使用该功能，被拒！

**akring**

第一次：应用内用了圆角按钮，无情被拒。

第二次：你给我解释解释，应用里的积分从哪里来，到哪里去？

**acgf**

应用里面表示有广告，但是审核者玩了后说没广告，后来申述成功，因为第二次游戏的时候才会出现广告（虐心重复向游戏），你丫就不能玩两遍么，一遍就几十秒~~

**andebi**

有第三方支付（支付宝）果断被拒！！！

**desert\_fox**

因为隐私条款被拒了2次，第一次是以为要在程序里加隐私条款，被拒了，还给我详细说要在哪里改。

**铁蛋**

因为集成了友盟  友盟获取用户mac地址被拒了。

**诸神的黄昏曲**

您的内容因为没有举报功能、含有色情内容不能通过。

**sandpiper**

17.5  隐私政策被拒

我的应用没有登录，没有注册，没有个人信息相关，我的应用完全没有隐私可言，他竟然给我17.5 这隐私政策。

使用第三方的logo被拒。

由于涉及到联通运营方面 果断在关于里面加了个联通的logo 标识合作伙伴 果断被拒 要求提供联通的书面证明。

**没给测试帐号**

同17.5 应用没有登录，没有注册，完全面向公众，他给我的拒绝理由是没有提供测试帐号。

获取广告标识符 没有使用被拒绝

果断我应用中加了各种广告还是被拒，启动页广告，加底部广告都加了，他竟然告诉我我的应用没加广告而被拒，申诉后告诉我要用苹果的iAD XXXX,果断去掉广告标识符.

**mohao**

理由是QR CODE里边有android的下载地址（直接转跳iTunes后通过）

上周悲剧了，理由是10.1,不符合iOS UI审美观（改UI）

还有一次悲剧了,理由是只有第三方登陆，没有自己的注册登陆功能...（新增登陆注册功能通过）

**angellixf**

你的软件加了强制评论功能：就一个按钮点击进入评论，必须改成先提示框提醒用户是否进入评论.

**womenshihaic**

第三方支付被拒,要求使用苹果内购 30%啊30%,我的应用是现实物品交易,这30%卖家出还是买家出？

**sffofn**

存放文档的地方由于iCloud会自动备份而被拒绝，只有用户自己使用和创建出来的才允许放在Document文件夹下。

**花太香齐**

审核员打不开我的接口，换海外vps，过最蛋疼的。

**iampat**

因为截图里面放了iPhone的模型被拒。

**乔布斯2013**

1.游戏运行崩溃；

2.游戏截图中有“测试字样”；

3.游戏太简单；

4.一次提交多个相同的游戏；

5.游戏中使用其他版权图片（使用了flappybird原图）

6.加了广告框架，游戏中缺没有广告显示（其实是在game over时候显示，结果测试人员就一直不知道怎么能死）；

7.内购不能使用；

8.game center不能使用。

**createmiracle**

App Store显示名字和软件名称不符合。

支付时不得强制获取用户信息 （比如  登陆、注册），必须在看到价格之前让用户登陆、注册。

App内购，产品类型不对。

**lyywhg**

我遇到过

第一：和电信合作的应用（因为有电信Logo），苹果拒绝：需要提供电信合作协议书。

第二：电子书项目，前一个版本审核通过了，后一个被拒绝，苹果拒绝理由：不再支持未经版权授权的单本书籍类应用。

第三：包中含有Beta文件，苹果拒绝理由：不支持Beta版本上传苹果商店。

第四：服务器恰好那时间在维护，苹果拒绝理由：登录注册都不能用。

**下雨天、轮滑**

1.说app使用了广告标示符，但没发现有使用广告标示符的地方。

解决：移除了Adsupport框架。（不知啥时候加的）

2.内购没有Restore按钮；

3.有登陆注册功能，但没有提供测试账户（大意了）

4.有去除广告的按钮，但没发现有广告。

解决：后来截了个图，回复审核那边，两小时之内就通过审核了；

5.版权问题。没有提供相关的版权文件

6.评级问题：提交的时候是评级4星，后来被拒，改成12星，通过审核了。

**evilshot**

3次 20.5条款，每次都出新状况，我勒个去，最主要的是还没有上架。

**怀念你的温柔**

1.关键字不符合要求

2.提交时说有idfa，但苹果测试时没看到广告，⊙﹏⊙b汗

3.内容太简单，说是浪费用户时间

4.界面风格不符合iOS风格

5.应用评级，从4+改成12+，再改成16+最后说我不符合16+，我再改回4+，竟然过了...

**rainieljy**

就说个我觉得最搞的一次悲剧吧：Splash上放了个蝙蝠侠蜘蛛侠，版权问题未给通过，后让UI改画了一个猥琐男人，通过了。

**zkforever**

应用内有个保存图片至相册的按钮，然后plist里有个Application supports iTunes file sharing设为了yes，无情被拒。

**woodn\_z**

用户在应用里自己下载的文档都不能存放在Document文件夹下，被拒。

**a4110494**

就一条 : 为了获得审查应用程序的完整功能，我们需要相关的硬件。

我以前都是录制一个操作视频上传然后就通过了，这回我录了2次都是相同的回复，不知道抽什么风。

**zpjvip**

做了款社交的软件，上线的时候没有提供账号和密码，导致审核的时候进不去,reject.

ps:那么大个注册 看不到么？看不到么？

**computernot**

论坛模块里由用户发的活动贴提到安卓平台和WP平台被拒，然后我说我没问题，他说三个工作日之内电话联系，过去4-5天了，还没联系。

**moonfrey**

按钮位置不符合ios风格。

**huanghanwei818**

一个笑话APP，开机画面上有“逗比”这两个字，苹果审核说含有粗俗不文明的文字，然后被拒了。

**wengxianxun**

QQ登陆功能，但是没下载QQ就不行。

**sinyran**

第一次没有给用户举报的功能。加了只有新浪和qq登录没有自己的登录注册系统（QQ SDK的没装QQ提示的是下载）。好像还有没有用户协议巴拉巴拉，反正那次审核就耗一月的时间。

**米兰9号**

审核人员打开app无法加载内容，一般是因为国内服务器的问题。

解决方法是录个app的操作视频，放到youtube上，发给苹果，屡试不爽。

zhangyuchang1

界面太丑被拒，换了张背景图通过了。

**职业吐槽**

iPad 写成了IPad。

**bluesky0318**

没提供注册功能，被拒...开放注册通过；

按钮图片类似iPhone桌面图标被拒；

因为iCloud云备份的问题被拒绝，将备份功能关闭通过。

**royluoyao**

使用第三方SDK，有个提示信息遮挡了状态栏。

**graceydb**

图片上有链接。

**shuaiyinoo**

因为有android关键字被拒绝了，因为描述中有android关键字被拒绝了，因为有swfit关键字是个没发布的语言被拒绝了。因为按钮点击无效被拒绝了。

**泊雨**

说功能太简单了, 不让上。

**zhuang.haishao**

1.使用第三方登录

2.内容包含苹果产品iPad

3.IDFA

4.有竖中指的图片

5.IDFA展示广告没有提供视频

**chaobaji**

1.注册页未添加pravicy声明文件。

2.iTunes升级后，分级莫名奇妙由12+变成了4+。

3.用了显眼的词语,其实就是"Beautiful girl"之类的，说内容令人反感。

4.引导页文案与内容不符。

**solomon\_bobo**

因为应用截图被拒。

美术偷懒给了4张android的截图，虽然app内容是一样的，但是顶部的状态栏是Android的！

**野猪洋洋笨**

我能想到的被拒理由大概就这些吧：

1.名字不符合，包含不良信息，包含与当前app不符的内容，包含特殊含义的歧义字符---被拒

2.苹果审核人员测试过程中出现crash或黑屏---被拒

3.没有给审核人员测试账户---被拒

4.截图与app内容不符---被拒

5.app代码中含有读取设备udid的代码---被拒

6.app中有积分墙---被拒

7.app中有第三方支付(如短信支付，积分墙支付)

8.App的功能过于单一或仅仅是一个demo---被拒

9.在注册时强制获取用户信息---被拒

10.需要提供原作者的版权授权书---被拒

11.App中有IAP内购，但是没有恢复内购的按钮---被拒

12.强制玩家给app评5星好评---被拒

13.没有兼容iOS---无情被拒，无情被拒，无情被拒啊。