iPhone - in-App purchase consumable correct approach

Asked 13 years, 6 months ago Modified 7 years, 3 months ago

Viewed 17k times

29

I have this new app I am creating that will use consumable in-app purchases.

My question is this: how does that work? I mean, imagine the user buys a consumable stuff. So, after finalizing the sell I set a flag on the app's database authorizing the use of that object. I read somewhere that I have to provide the user with a button to restore old transactions in case of the user for some reason loses his device and has to restore everything.

Imagine the user has already used that purchase and after that he restores the old in-app purchases. What happens then? Will the user have the same resources again, so he can use a second time without paying? How it works and how should I approach that?

thanks

iphone in-app-purchase

Share

edited Jun 23, 2011 at 6:20

Improve this question

Follow

RK-

12.2k ● 23 ● 93 ● 157

asked Jun 23, 2011 at 4:23

Duck

35.9k ● 48 ● 259 ● 488

3 Answers

Sorted by: Highest score (default)

69

I wanted to share a somewhat unorthodox solution I found to this problem that has the HUGE advantage of not requiring a server. This method allows users to restore their consumable items if the app is deleted and reinstalled, but does not allow them to move the items to a new device (unless all their app data is copied over).

Data stored in the keychain persists when an app is deleted and reinstalled. The keychain is intended for storing usernames and passwords, but you can also store information about consumable purchases in there. I used the KeychainItemWrapper class, available here: https://developer.apple.com/library/content/samplecode/GenericKeychain/Introduction/Intro.html

Here is some sample code where I store and retrieve the number of paid hints that a user has remaining:

```
//Storing the consumable hint item count
int hintsLeft = 100;
KeychainItemWrapper *wrapper = [[KeychainItemWrapper a
initWithIdentifier:@"Hints" accessGroup:nil];
NSString *hintsString = [NSString stringWithFormat:@"%
[wrapper setObject:hintsString forKey:(id)kSecValueDat
```

```
[wrapper release];

//Retrieving it
KeychainItemWrapper *wrapper = [[KeychainItemWrapper a
initWithIdentifier:@"Hints" accessGroup:nil];
NSString *numHints = [wrapper objectForKey:(id)kSecVal
[wrapper release];
int retrievedHints = [numHints intValue];
```

Notes:

- the key (id)kSecValueData can't be an arbitrary string, there is a set list of constants that you can use as the key.
- You will need to add the security framework

Share Improve this answer

Follow

edited Jun 29, 2017 at 13:02

Cœur

38.6k ● 26 ● 202 ● 276

answered Feb 1, 2012 at 20:00

user1139733

996 ● 9 ● 14

6 This is the kind of post I wish I could upvote twice. Excellent, creative persistent solution to a perennial problem. I had this working in 15 minutes, and highly recommend the approach for tracking consumables. – SG1 Mar 7, 2012 at 0:38

3 But I assume, if the user restores his device as new, the IAP consumables would be lost. Wouldn't that be a problem and result in annoyed users? – Gottfried Apr 2, 2013 at 9:00

3 This is great solution!! But now you can get allow them to move a new device for iOS 7 too – use iCloud + Keychain

sync (kSecAttrSyncronizable) – alexmorhun Nov 14, 2013 at 12:50

So `kSecValueData` must be unique for each consumable purchase. How do we get a value for that? Are we storing the whole iap data in kSecValueData ? Or when you restore in app purchases does Apple provide a unique ID for each consumable? (In case they buy the consumable 13 times). How does this work? – Albert Renshaw Feb 6, 2014 at 17:18

1 Hi, I'm using this solution as well and it works nicely. However, I assume there will be problems when 1) Users log-in to their icloud keychain with a separate account from their appstore account (which is possible, see iOS Settings) and 2) By using consumables on two devices, how can device B (let's say it's entirely offline) know how many consumables (of the same account) were consumed on device A? (without restricting use of consumables when a device is offline) – pille Jan 6, 2015 at 21:27

29

StoreKit only provides recovery functionality for non-consumable items (and to some extend for subscriptions). So for consumable products, recovering using `restoreCompletedTransactions` will not deliver any transactions in your case. Any handling of restoring information about consumable products must be done within your app and/or server.

For reference of the various products' natures check the In App Purchase Programming Guide:Designing Your App's Products. There are **Consumable** products which must be purchased each time, **Non-consumable** products purchased only once and provided to all devices associated with that user's iTunes account, **Auto-**

Renewable subscriptions and **Non-Renewing** Subscriptions

Share Improve this answer Follow

edited Sep 23, 2017 at 7:05

answered Jun 23, 2011 at 5:23

marcus

2,531 ● 1 ● 23 ● 32

3 that what I was suspecting. Thanks. – Duck Jun 23, 2011 at 5:43

Marcus and Robot, What do u refer to as consumable and non-consumable in case in-app-purchase? – RK- Jun 23, 2011 at 6:19

4 @Krishnan The Apple documentation epxlains all these terms. Check the In App Purchase Programming Guide:Registering Products with the App Store. There are **Consumable** products which must be purchased each time, **Non-consumable** products purchased only once and provided to all devices associated with that user's iTunes account, **Auto-Renewable subscriptions** and **Non-Renewing Subscriptions** – marcus Jun 23, 2011 at 8:18

This is the correct answer. there is no "restore" feature for consumable in-app purchases. Apple will not reject your app for this reason. If they do then submit an appeal. It happened to me and they agreed that restore doesn't make sense for consumable in-app – Sam B Dec 23, 2016 at 1:32

1 @marcus your link is dead. This one seems better. developer.apple.com/library/content/documentation/… – Heath Borders Sep 20, 2017 at 15:47

5

For people who come hare searching for the way to store consumable items locally in iOS, have a look at PDKeychainBindingsController (https://github.com/carlbrown/PDKeychainBindingsController).

It works like NSUserDefaults and can be used to store the purchased consumable item count in iDevice's keychain (the items stored in keychain do not get removed while deleting the app.).

Use the code something like below to store and retrieve the value from keychain:

```
- (NSUInteger)hintCount {
    PDKeychainBindings *wrapper=[PDKeychainBindings sh
    NSString *valueString = [wrapper objectForKey:@"hi
    int value = [valueString intValue];
    return value;
}

- (void)setHintCount:(NSUInteger)starCount {
    PDKeychainBindings *wrapper=[PDKeychainBindings sh
    NSString *valueString = [NSString stringWithFormat
    [wrapper setObject:valueString forKey:@"hintCount"
}
```

Share Improve this answer Follow

edited Dec 3, 2015 at 22:44

answered Feb 13, 2013 at 3:32

Abduliam Rehmanius

928 ● 12 ● 23