

Integrating with the G6 Pay Per Install iOS SDK

SDK Setup

Download

To install the G6Pay SDK, first download the latest G6SDK package. After unzipping the download archive, you will see the following directories:

G6SDK/G6 Demo.xcodeproj/

G6SDK/G6Pay/

G6SDK/doc/

The G6Pay folder contains all the source files that will be needed to integrate into your Xcode project.

The G6 Demo.xcodeproj can be opened in Xcode. This is a sample project that demonstrates how to utilize the G6 SDK.

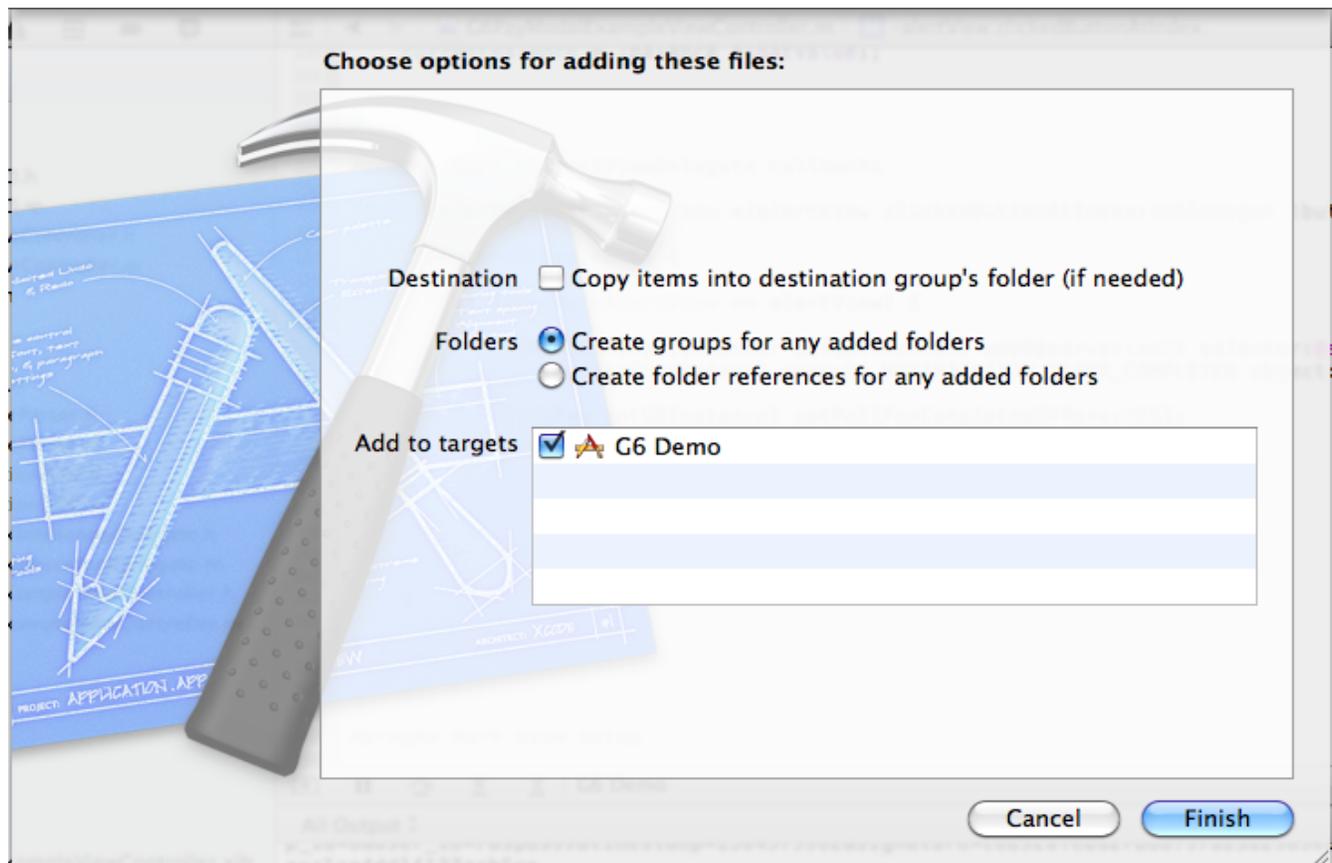
The doc directory contains additional documentation.

Application Setup

The G6SDK/G6Pay directory contains the source files that need to be copied into your application. You should copy this folder into your Xcode project. Simple drag the G6Pay folder into your Xcode project.

| Name | Date Modified | Size |
|---------------------------------------|-----------------|-------|
| ▶ Classes | Today, 11:37 PM | -- |
| h G6 Demo_Prefix.pch | Today, 11:37 PM | 4 KB |
| 📄 G6 Demo-Info.plist | Today, 11:37 PM | 4 KB |
| 📁 G6 Demo.xcodeproj | Today, 11:37 PM | -- |
| ▶ G6Pay | Today, 11:37 PM | -- |
| 📄 G6PayModalExampleViewController.xib | Today, 11:37 PM | 8 KB |
| ▶ images | Today, 11:37 PM | -- |
| 📄 main.m | Today, 11:37 PM | 4 KB |
| 📄 MainWindow.xib | Today, 11:37 PM | 20 KB |

When prompted, selected “create group...” and select “Finish”.



After you have copied the source files to your project, you can set up the SDK to utilize your G6 APP ID and Secret key.

Your APP ID and Secret key can be obtained from the G6 dashboard. You will need these values to configure the SDK.

Dashboard: My Apps: G6 SDK [edit](#)

App details

Analytics

Code integration

App details

App ID

329

Platform

iPhone

Currency name

gsixes

Exchange rate

5.00

Secret key

7594d9e[REDACTED].26202589 (required for signing any API call)

Total revenue

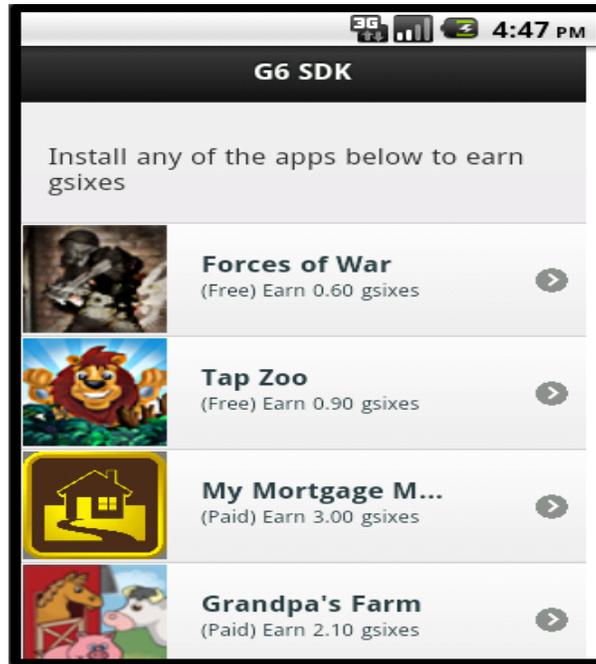
\$0.00

Monetization methods:



SDK Usage

Showing the Offer Wall



Showing the offer wall is made available easily with a single call.

If you would like to be notified of offer completion, add an observer for the `G6_NOTIFICATION_OFFER_COMPLETED` notification type. This method will automatically be called with a `G6OfferDTO` when an offer is completed. You will also need to enable polling for completed offers by calling `setPollForCompletedOffers`:

```
#import "G6Pay.h"
```

```
... snip ...
```

```
[G6Pay initWithAppId:@"YourAppId" andSecretKey:@"yourAppSecret"];
```

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(offerCompleted:)  
name:G6_NOTIFICATION_OFFER_COMPLETED object:nil];
```

```
[[G6Pay getG6Instance] setPollForCompletedOffers:YES];
```

```
[[G6Pay getG6Instance] showOffers:userId
```

```
delegate:nil
```

```
parent:self
```

```
showNavigationBar:YES
```

```
navigationBarOnTop:YES];
```

```
... snip ...
```

Crediting User

Crediting a user with currency on the G6 platform can be done with one line. If you'd like to receive a notification when the call has completed, you can add an observer for the G6_NOTIFICATION_CREDIT_USER_SUCCESS notification.

```
#import "G6Pay.h"
```

```
... snip ...
```

```
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(creditUserSuccessNotification:)  
name:G6_NOTIFICATION_CREDIT_USER_SUCCESS object:nil];
```

```
[[G6Pay getG6Instance] creditUser:@"uniqueTransactionId"  
                                  userId:@"uniqueUserId"  
                                  amount:amount  
                                  delegate:nil];
```

Debiting User

Debiting a user with currency on the G6 platform can be done with one line. If you'd like to receive a notification when the call has completed, you can add an observer for the G6_NOTIFICATION_DEBIT_USER_SUCCESS notification.

```
#import "G6Pay.h"
```

```
... snip ...
```

```
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(creditUserSuccessNotification:)  
name:G6_NOTIFICATION_DEBIT_USER_SUCCESS object:nil];
```

```
[[G6Pay getG6Instance] debitUser:@"uniqueTransactionId"  
                                  userId:@"uniqueUserId"  
                                  amount:amount  
                                  delegate:nil];
```

Getting User Balance

Debiting a user with currency on the G6 platform can be done with one line. If you'd like to receive a notification when the call has completed, you can add an observer for the G6_NOTIFICATION_USER_BALANCE_SUCCESS notification. The object associated with the notification object will be an NSNumber with the user's current balance

```
#import "G6Pay.h"

... snip ...

[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(getBalanceSuccessNotification:)
name:G6_NOTIFICATION_USER_BALANCE_SUCCESS object:nil];

[self.activity startAnimating];
[[G6Pay getG6Instance] getUserBalance:userId delegate:nil];
```

Getting User Transactions

Debiting a user with currency on the G6 platform can be done with one line. If you'd like to receive a notification when the call has completed, you can add an observer for the G6_NOTIFICATION_GET_TRANSACTIONS_SUCCESS notification. The object associated with the notification object in the call will be an NSArray of G6TransactionDTO objects.

```
#import "G6Pay.h"

... snip ...

[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(getTransactionsSuccessNotification:)
name:G6_NOTIFICATION_GET_TRANSACTIONS_SUCCESS object:nil];

[[G6Pay getG6Instance] getAllTransactions:userId delegate:nil];
```

Using Delegate Protocols

Instead of using the NSNotificationCenter, you can also utilize the G6 delegate protocols and specify a delegate for every API call. The protocols are described in G6Protocol.h. Delegates passed to the API methods must implement the appropriate protocol.

```
@protocol G6OffersDelegate <NSObject>
```

```
-(void) offerWasCompleted:(G6OfferDTO *)offer;
```

```
@end
```

```
@protocol G6UserAccountDelegate <NSObject>
```

```
-(void)creditUserSuccess:(NSString *)userId transactionId:(NSString *)transactionId amount:(float)amount;
```

```
-(void)creditUserFailure:(NSString *)userId transactionId:(NSString *)transactionId amount:(float)amount;
```

```
-(void)debitUserSuccess:(NSString *)userId transactionId:(NSString *)transactionId amount:(float)amount;
```

```
-(void)debitUserFailure:(NSString *)userId transactionId:(NSString *)transactionId amount:(float)amount;
```

```
-(void)getUserBalanceSuccess:(NSString *)userId balance:(float) balance;
```

```
-(void)getUserBalanceFail:(NSString *)userId;
```

```
@end
```

```
@protocol G6TransactionDelegate <NSObject>
```

```
-(void)getAllTransactionsSuccess:(NSString *)userId transactions:(NSArray *)transactions;
```

```
@optional
```

```
-(void)getAllTransactionsFail:(NSString *)userId;
```

```
@end
```