iOS Developer Guide

In-App Payment

v.1.4.3

Table of Content

Upgrading from previous versions

Please note if you upgrade from 1.3.8 or older then we changed some of the API parameter names.
Update your code to the new names as detailed in this document.

Before Starting

ZooZ In-App checkout library is a complete and secure solution that enables you to start accepting payments in mobile applications. ZooZ SDK designed to make integration easy and seamless.
ZooZ handles the entire payment process for you – no merchant account or payment gateway is required.

If you do have a processing merchant account and a gateway, we most probably can link to it as well.
Your earnings will be transferred directly to a Prepaid Debit Card (by Payoneer), your PayPal account, your local bank account or to any major e-wallet.

Prior to initiating any forms of integration, it will be necessary for you to sign up to ZooZ www.ZooZ.com

The purpose of this guide is giving you all the information for seamless integration, however if you have any question you are welcome to contact us in support@ZooZ.com

Supporting Platforms

The library supports armv7 and armv7s architectures for iOS SDK 4.3 and above.
It’s recommended to use it with SDK5 .0 and above.

The library supports the following iOS devices – iPad all versions, iPhone 3Gs and above, iPod touch 3 and above.

The SDK does fully support iPhone 5.

Get Started with ZooZ

1. **Sign-Up** to ZooZ Developer Portal at www.ZooZ.com
2. **Register new App –** Go to "My Apps" fill in your app details.
   Make sure you register the bundle ID as defined for your project target setting "*Bundle Identifier*" property in your info.plist definition (visible in xCode under target summary view).
3. **Download SDK for your platform**
4. **Integrate code** – Example and details in this guide
5. **Verify success integration** by test your app to Sandbox environment. In order to test transaction on sandbox use one of the following test credit cards:
   a. Test Credit Cards (You can use any CVV and future expiry date)
      i. 4580-4580-4580-4580
      ii. 4111-1111-1111-1111
      iii. 5105-1051-0510-5100
   b. Test PayPal
      i. Username: user_1323072569_per@tactusmobile.com
      ii. Password: 12345678

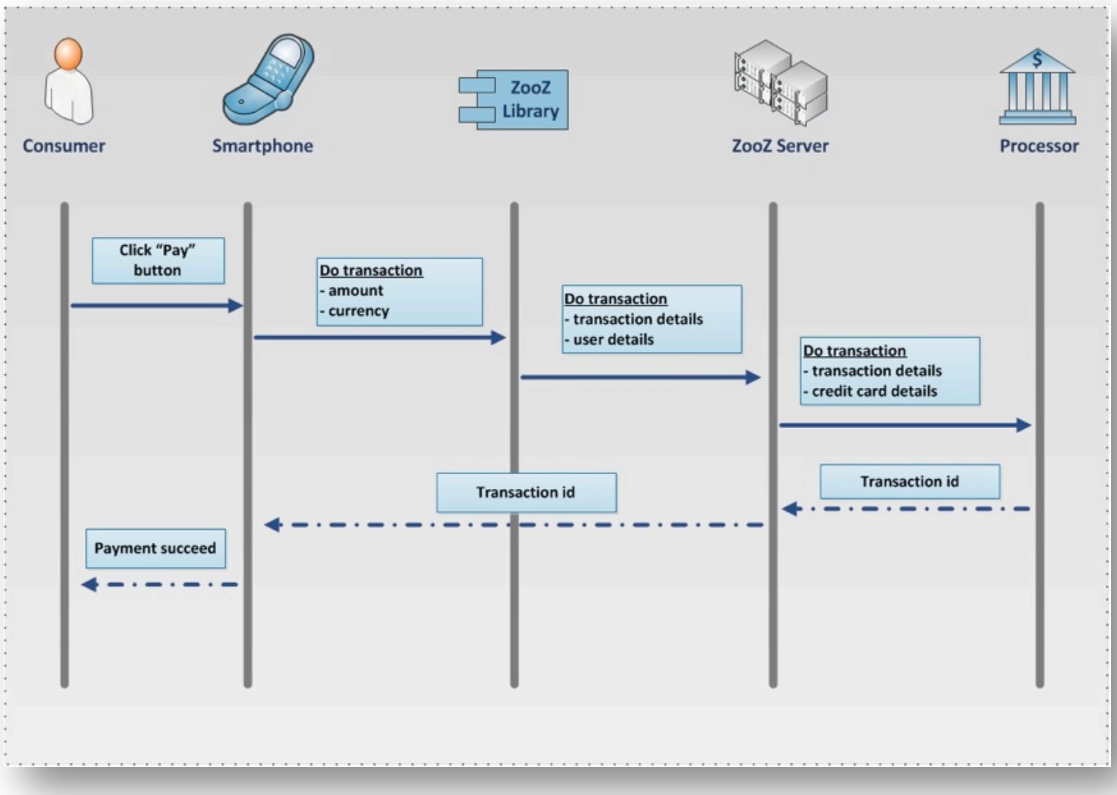
   Integrate code in your app with the given "App Key" + Bundle ID and get immediate access to the sandbox environment.
6. **Going live – Submit App for going live –** Go to Developer Portal and submit your app for going live.
7. **Go Live –** When approved (Don't forget to change your environment flag to "Production").
8. **Please start the go live process withing ZooZ before submitting to Apple.**

Congratulations!

You can start accepting payments in your application

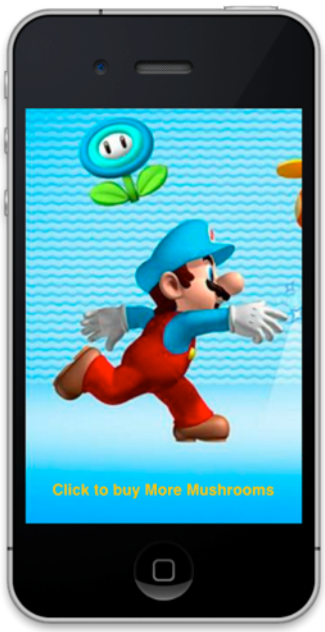
High Level Transaction Flow

ZooZ handle the entire finance transaction flow for you, below sequence diagram demonstrate transaction flow from the end-user clicked "Pay" to the time payment approved by the processor.

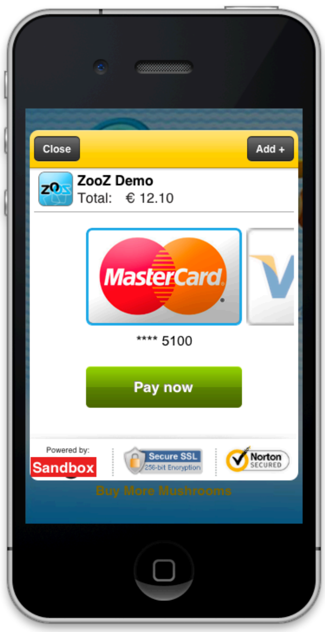


User Experience

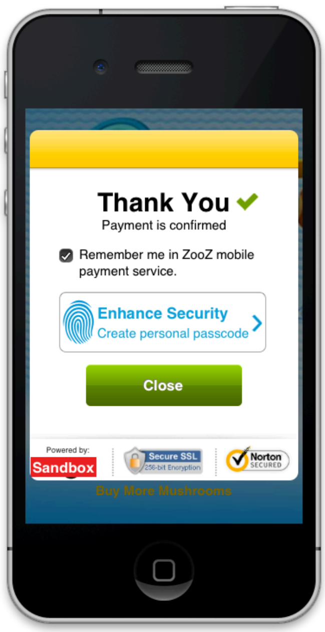
On going flow from user point of view – returning user



Consumer wants to purchase goods from app

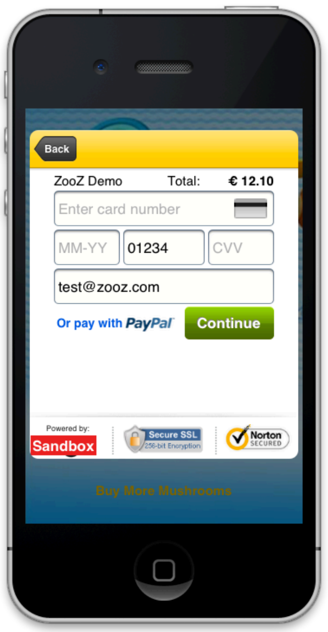


Consumer selects his preferred payment method

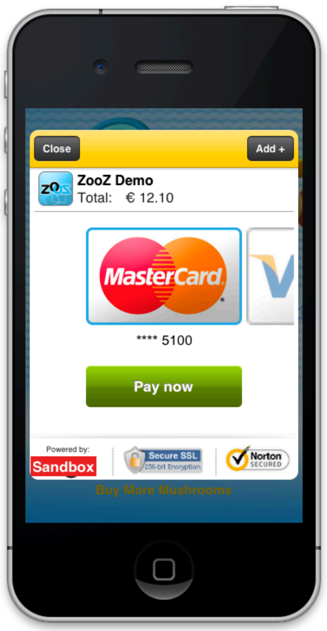


Purchase is confirmed. Option to enhance security by adding a passcode

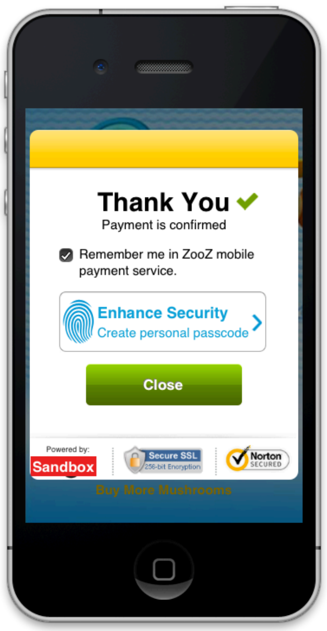
First time user



Consumer fills in his Credit Card details, 1st time only



Consumer approve the payment

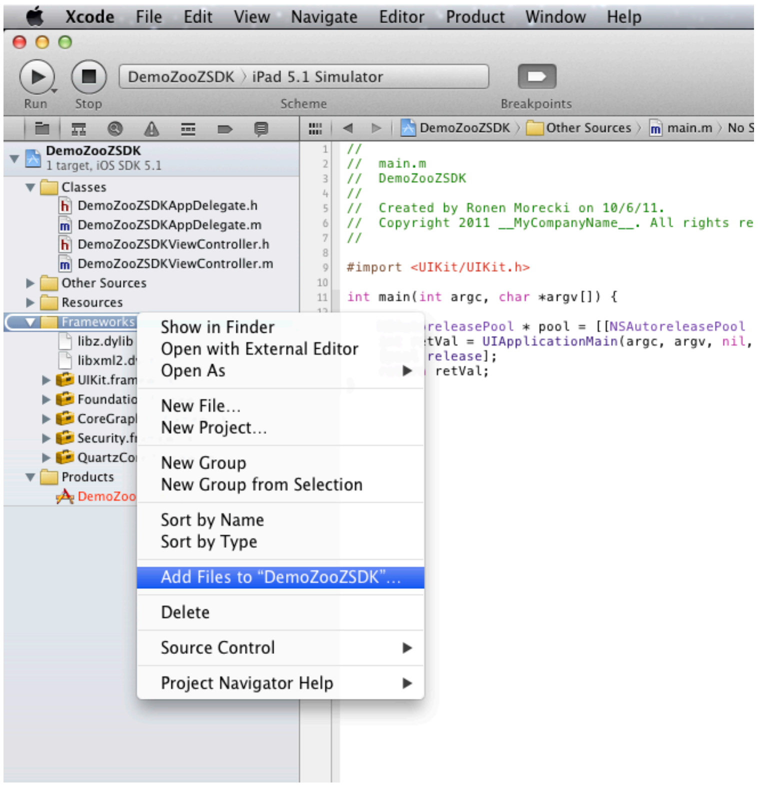


Purchase is confirmed. Consumer optionally can enhance security by adding a passcode

Integration

Setup

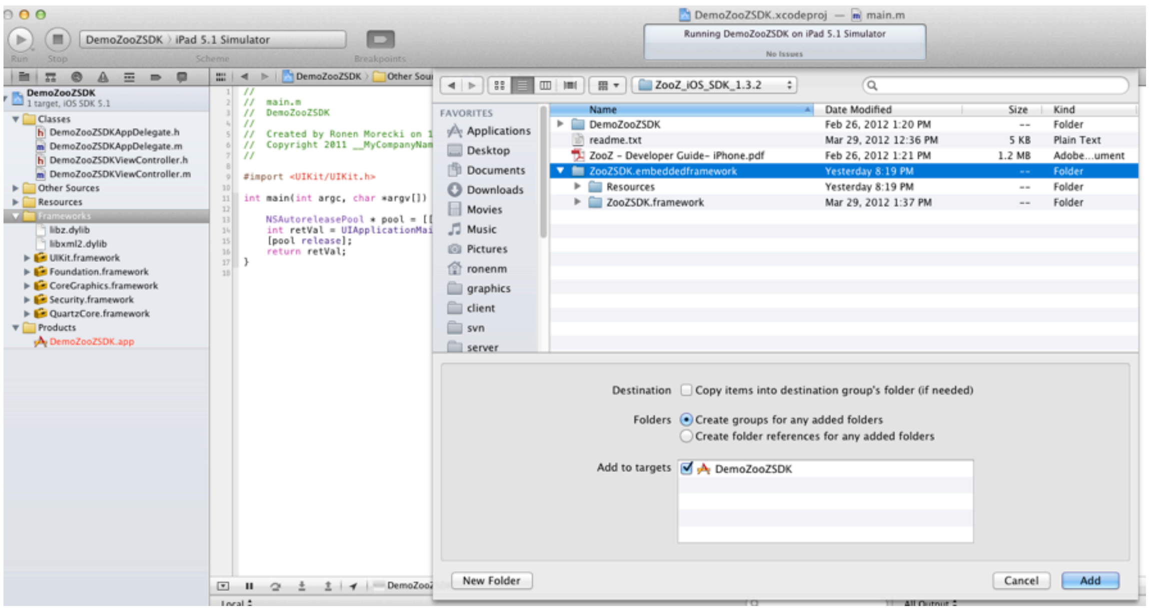
1. Add the ZooZ library to your project.
   a) Extract the downloaded zooz zip file to some folder in your computer. Please unzip only on **Mac** machine, Windows OS **does not** extract correctly the zip file.
   b) Right click on your project tree and choose to add new files to the project. Choose the ZooZSDK.embeddedframework Folder.
   Or Drag the extracted "ZooZSDK.embeddedframework" Folder (With all its subfolders) into your iPhone Xcode project framework section. Choose to add file, make sure you do not mark the "copy" checkbox.
   See screenshots:



```
//
//  main.m
//  DemoZooZSDK
//
//  Created by Ronen Morecki on 10/6/11.
//  Copyright 2011 __MyCompanyName__. All rights re
//

#import <UIKit/UIKit.h>

int main(int argc, char *argv[]) {
         releasePool * pool = [[NSAutoreleasePool
         tVal = UIApplicationMain(argc, argv, nil,
          release];
          retVal;
```

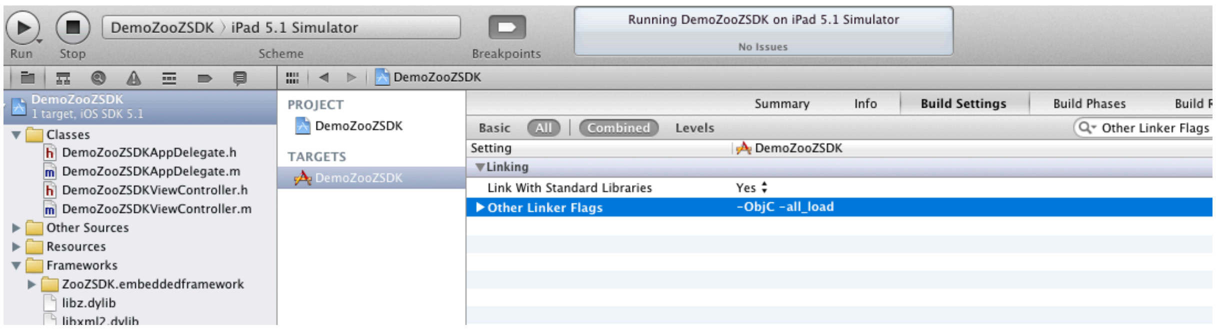


Name	Date Modified	Size	Kind
DemoZooZSDK	Feb 26, 2012 1:20 PM	--	Folder
readme.txt	Mar 29, 2012 12:36 PM	5 KB	Plain Text
ZooZ – Developer Guide– iPhone.pdf	Feb 26, 2012 1:21 PM	1.2 MB	Adobe...ument
ZooZSDK.embeddedframework	Yesterday 8:19 PM	--	Folder
Resources	Yesterday 8:19 PM	--	Folder
ZooZSDK.framework	Mar 29, 2012 1:37 PM	--	Folder

Destination: Copy items into destination group's folder (if needed)
Folders: [✓ Create groups for any added folders] Create folder references for any added folders
Add to targets: ✓ DemoZooZSDK

c) Add to your project `Security, Quartz, libxml2.dylib and libz.dylib` frameworks.
You can do that in the target setting window → Build Phases → Link binaries with Libraries.
If you had PayPal library, remove it from your project, as ZooZ already includes it.

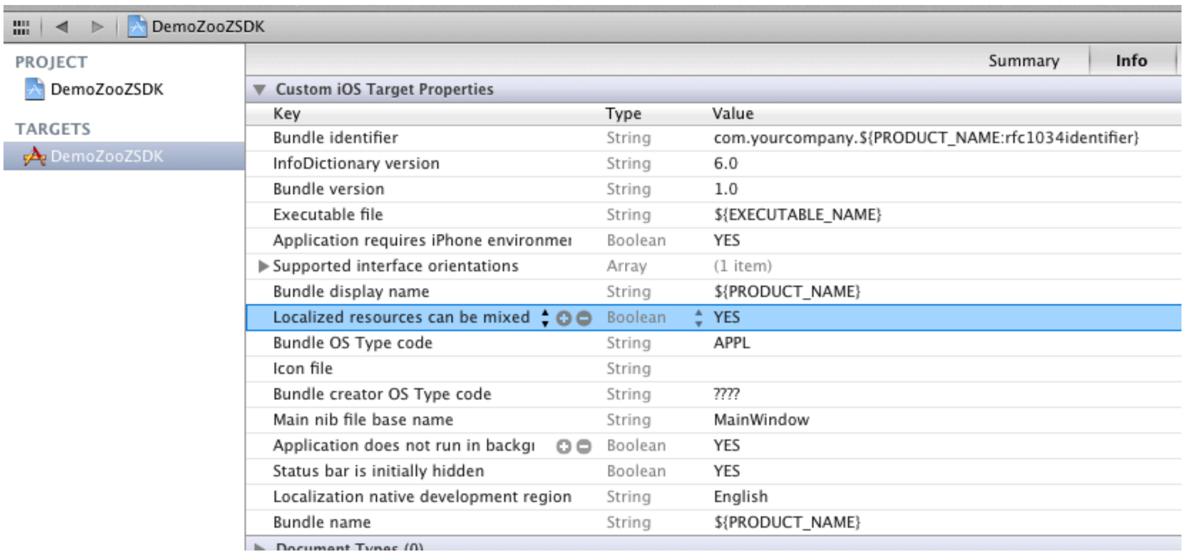
d) In your project target settings add "`-ObjC`" flags to the "`Other Linker Flags`" property



Setting	DemoZooZSDK
Linking	
Link With Standard Libraries	Yes
Other Linker Flags	-ObjC -all_load

e) To support localization languages other then English add to your `app-info.plist` definition the following flag:

"Localized resources can be mixed"=YES



Key	Type	Value
Bundle identifier	String	com.yourcompany.${PRODUCT_NAME:rfc1034identifier}
InfoDictionary version	String	6.0
Bundle version	String	1.0
Executable file	String	${EXECUTABLE_NAME}
Application requires iPhone environmer	Boolean	YES
Supported interface orientations	Array	(1 item)
Bundle display name	String	${PRODUCT_NAME}
Localized resources can be mixed	Boolean	YES
Bundle OS Type code	String	APPL
Icon file	String	
Bundle creator OS Type code	String	????
Main nib file base name	String	MainWindow
Application does not run in backgı	Boolean	YES
Status bar is initially hidden	Boolean	YES
Localization native development region	String	English
Bundle name	String	${PRODUCT_NAME}

f) For adding the SDK with the credit card camera scan feature, please see the relevant section at the end of this document with additional steps.

Coding

1. Create a payment request and open the payment dialog.

   In your checkout button action add the following code –

```
-(IBAction)pay{
     ZooZ *zooz = [ZooZ sharedInstance];
     zooz.sandbox = YES;//set this if working in Sandbox mode
     ZooZPaymentRequest *req = [zooz createPaymentRequestWithTotal:32.1
invoiceRefNumber:@"1234" delegate:self];
     req.currencyCode = @"USD";
     req.payerDetails.email = @"test@test.com";
     [zooz openPayment:req forAppKey:@"app_id"];

}
//There are more optional properties that can be added to this API - see in this
document
//invoiceRefNumber is your system accounting invoice reference, it is for tracking only
and not used by our system.
//appKey - is the app key you received from us upon registration
```

2. Implement the callbacks –
   Your ZooZ payment delegate should implement the following protocol (`ZooZPaymentCallbackDelegate`):

```
-(void)paymentSuccessWithResponse: (ZooZPaymentResponse *)response{
      /* CALLED BEFORE THE ZOOZ DIALOG IS CLOSED BY THE USER
the payment finished successfully call back to dialog is on background thread, no need
to auto release pool, as this been taken care of. You shouldn't update your UI on this,
just process the payment data */
}

-(void)paymentSuccessDialogClosed{

     /* Dialog is closed by the user after payment finished successfully (see
paymentSuccessWithResponse: - this is where you should update your UI on success
transaction */
}



-(void)paymentCanceled{

     //User decided to close the dialog and not to pay
}



-(void)openPaymentRequestFailed: (ZooZPaymentRequest *)request withErrorCode:
(int)errorCode andErrorMessage: (NSString *)errorMessage{

     //Some error occurred with opening the request to ZooZ servers, usually a network
issue or wrong credentials issue
}
```

ZooZ SDK Class API Reference

ZooZ	Data Type	Description
sandbox	BOOL	Flag for development times – work in sandbox mode
createPaymentRequestWithTotal: invoiceRefNumber: delegate:	ZooZPaymentRequest	Create the payment request object
createManageFundSourcesRequest WithDelegate:	ZooZPaymentRequest	Create the pre register cards request – similar to payment but users just authorize the cards and not paying with them.
openPayment:req forAppKey:	void	Opens the payment dialog and start the checkout flow
tintColor	UIColor	tintColor of the popup dialog navigation bar
barButtonTintColor	UIColor	tintColor of the popup dialog navigation bar buttons
barTitleImage	UIImage	Image for be shown on the popup dialog navigation bar
preInitialize:(NSString *)appKey isSandboxEnv:(BOOL)	BOOL	Optional Call to pre-init the app with ZooZ properties. This can be called on app load or page load of your app and it should run in the background thread. This will make the payment dialog popup immediately when you call the openPayment call. Returns YES if successful.

ZooZPaymentRequest	Data Type	Max Chars	Description
currency	NSString	3	3 letter currency code (ISO 4217)
amount	float		Requested pay amount
requireAddress	BOOL		If to ask from the user his Zip code or not (Default YES)
invoiceNumber	NSString	100	Your tracking system invoice reference used for reporting and tracking
addItem:	void		Adds an invoice item to the list of items purchased. This is an optional API.
Invoice.additionalDetails	NSString	200	Custom free text for invoice description – 200 chars

The response object that is returned on successful payment:

ZooZPaymentResponse	Data Type	Description
transactionDisplayID	NSString	A unique transaction ID, store that value for audit. An easy to view transaction identifier.
transactionID	NSString	Transaction ID to your record that the transaction completed successfully. You're advised to store the transaction ID for future tracking. Transaction ID can also be used with the ZooZ Extended Server API (For more details see the Download section of the ZooZ developer portal).
fundSourceType	NSString	Credit card brand or PayPal
lastFourDigits	NSString	Last 4 digits of credit card

The ZooZSUser object allows your app to send to ZooZ more details about the paying user.
This information is very important for your payment tracking, refunds, payment reports and reduce transaction risk. It will also save some steps for the user filling out details that he already gave in the app.
In future these details will also help in your portal analytics.

ZooZUser	Data Type	Max Chars	Description
firstName;	NSString	50	Payer First name
lastName	NSString	50	Payer Last name
phoneNumber	NSString	20	Payer phone number
phoneCountryCode	NSString	5	Payer phone number country code
email	NSString	50	Payer email - You can use this attribute to track the user's email. You will see the value on the transaction details in the developer portal, and you can filter by this value.
additionalDetails	NSString	400	Any additional information that could be interested to add about the user (Ex. Your own managed user ID)
billingAddress	ZooZUserAddress		Payer billing address
shippingAddress	ZooZUserAddress		Payer shipping address

ZooZUserAddress	Data Type	Max chars	Description
country	NSString	50	Address Country
state	NSString	50	Address state where applicable
city	NSString	50	Address city
streetAddress	NSString	50	Street Address
zipCode	NSString	50	Address zip / postal code

Invoice Items, represent your cart items and can be sent to ZooZ. These items will be displayed in ZooZ reports and invoices that are sent.

ZooZInvoiceItem	Data Type	Max chars	Description
name	NSString	200	Display name for the item
price	float		Price per item
quantity	float		Quantity of items
itemId	NSString	100	Item reference number (optional)
additionalDetails	NSString	200	Custom text for describing the specific item

Enabling the card scan by camera feature

General
This ZooZ version introduces a new SDK option:
Instead of typing in credit card details now the user can photo scan his card using the device's camera.
The feature is powered by Jumio SDK and it's fully integrated in ZooZ.

We separated the SDKs packages to packages that include / exclude this feature due considerations of package size and integration steps.

In order to include this feature you need to follow the same integration steps as described in this document above, plus the steps below:

This framework supports iPhone 3GS and higher, iPad 2 and higher, iPod Touch 4G and higher and iOS 5.0 and higher

Setup
1. Instead of adding the ZooZSDK.embeddedframework folder to your Xcode add the ZooZSDK_with_camera.embeddedframework folder.
2. Also add to your project the following frameworks:
   a. AudioToolbox
   b. AVFoundation
   c. CFNetwork
   d. CoreGraphics
   e. CoreLocation
   f. CoreMedia
   g. CoreVideo
   h. Foundation
   i. OpenGLES
   j. QuartzCore
   k. UIKit
3. Add the following flag to your Xcode Build Settings in section "Other Linker Flags": "-lstdc++"

Coding
Please note the new ZooZ property: zooz.autoOpenScanWithCamera.
This flag allows you to open the camera by default for first time card.
Also you can setup your Jumio credentials so you will have access to the Jumio scans dashboard:
Simply open a free account with Jumio and enter the relevant credentials in:
netSWipeMerchantToken, netSwipeMerchantApiSecret properties.