

# Wiz-Card iOS SDK

Version 1.5.6

---



# Table of Contents

---

<b>1. Prerequisites</b>	<b>3</b>
<b>2. Sample App</b>	<b>4</b>
<b>3. Setup and installation</b>	<b>5</b>
3.1. Adding the framework	5
<b>4. Configurations</b>	<b>6</b>
4.1 Configure	6
4.1.1 The configurations object	6
4.1.2 Configure function	7
4.2 Other configuration options	8
4.2.1 Client ids	8
4.2.2 Client traffic sources	9
<b>5. API</b>	<b>11</b>
5.1 Track client journey	11
5.1.1 Page View	11
5.1.3 WizCart coupon copied	13
5.1.5 Client Cart updated	14
5.1.6 Purchase completed	14
5.1.7 Client Cart cleared	15
5.2 Get client incentive	16
<b>6. Models</b>	<b>17</b>
6.1 WizCartIncentiveResponse	17
6.2 Cart & Product	18
<b>7. Enums</b>	<b>20</b>
7.1 PageType	20

# 1. Prerequisites

Minimum iOS version: 9.0

Swift Version: 5

Supports Objective-C: Yes.

Please note, that this SDK was built for Swift, therefore some deviation from this guide is to be expected.

## 2. Sample App

---

Namogoo WizCart library is the communication platform which provides SDK, API to add targeted incentives to e-commerce consumers. This Quick Start guide will help you run a sample project powered by Namogoo as fast as possible. To run the sample project, you will need Xcode with swift 5 support installed.

1. Download the sample project from the following location:

<https://github.com/namogoo/wizcart-mobile-sdk/tree/master/iOS>

2. Open and run in Xcode

## 3. Setup and installation

---

To integrate Namogoo WizCart SDK to your application, you need a Namogoo WizCart account. If you do not have a WizCart account, contact your Customer Success Manager here: [killian.buffard@namogoo.com](mailto:killian.buffard@namogoo.com) . Namogoo provides you an account ID (account\_id) to activate the SDK.

WizCart SDK is built and designed to be used with Xcode. The following instructions will help you to integrate WizCart into your application:

### 3.1. Adding the framework

Copy WizCart-version.xcframework (download from- <https://github.com/namogoo/wizcart-mobile-sdk/tree/master/iOS>) to your local machine.

Using framework:

- Drag and drop into your project navigator inside Xcode
- Open your project file
- On the left pane select your target
- Click on General and scroll down to “Frameworks, Libraries and Embedded Content”
- Make sure the framework is marked as Embed & Sign
- If not, click on the drop menu and select Embed & Sign
- Done. Happy Coding!

Using cocoapods:

- Add `pod 'WizCart', '~> 1.5.6'` to your target app in the pod file and save changes.
- In the terminal app navigate to your project directory
- type `pod install` and press enter
- If cocoapods was unable to locate the podspec, run the pod install command with `--repo-update` arg like so: `pod install --repo-update`
- Done. Happy Coding!

## 4. Configurations

---

### 4.1 Configure

#### 4.1.1 The configurations object

Each application must register with Namogoo for a WizCart Account ID. For example:

```
let accountId = 25478
```

The WizCart SDK has a primary interface `WizCartClient` for interacting with the WizCart service. You have to configure the SDK before you can access its functions. A configuration object should be instantiated in your `AppDelegate` or when a user change occurs (i.e User is logging in / Logging out).

```
let wizCartConfigurations = WizCartConfigurations(withAccountId: accountId,
clientId: clientId,
tempClientId: tempClientId,
retailerVisitor: retailerVisitor,
isPreproduction: isPreProduction,
attributes: attributes,
trafficSources: trafficSources)
```

#### Parameters:

<u>accountId</u> : String	<b>*Must*</b> - The account you received from Namogoo Team.
---------------------------	--

<u>clientId</u> : String (OPTIONAL, DEFAULT VALUE: nil)	<b>*Must if the user is logged in*</b> WizCart will accept any unique String as a Client ID (UIDs, email addresses, phone numbers, usernames, etc), which we refer to as clientId. You can use any new or existing User Management system. Can be set later through WizCartSDK.setClientId(_isTempClientId:) - MUST be set prior to getIncentive call for logged in users.
<u>tempClientId</u> : String (OPTIONAL, DEFAULT VALUE: nil)	<b>*Optional*</b> WizCart will accept any unique String as a Client ID (UIDs, email addresses, phone numbers, usernames, etc), which we refer to as clientId. You can use any new or existing User Management system. tempClientId is an id that identifies the user when he is not logged in. If left empty, the SDK will create a temp id for you. Can be set later through WizCartSDK.setClientId(_isTempClientId:)
<u>retailerVisitor</u> : String (OPTIONAL, DEFAULT VALUE: nil)	<b>*Must*</b> Unique identifier that remains the same for the user regardless of his log in state. can be the same as clientId. If left empty the SDK will generate an id for you, and will preserve that id for subsequent app launches.
<u>isPreproduction</u> : Bool (OPTIONAL, DEFAULT VALUE: false)	<b>*Optional*</b> Should the sdk communicate with the development server or the production server.
trafficSources: Array<String> (OPTIONAL, DEFAULT VALUE: nil)	<b>*Must*</b> An array of strings that determine the user group and type, for example: "Personal", "User40"

#### 4.1.2 Configure function

Submit the configurations object to the SDK

### Parameters:

<u>wizCartConfigurations</u> : WizCartConfigurations	The configurations object we just created.
<u>delegate</u> : WizCartDelegate (OPTIONAL, DEFAULT VALUE: nil)	A class that conforms to the WizCartDelegate protocol. Inorder to get responses from the SDK. Can be set later through WizCartSDK.setDelegate(_:)

```
WizCartSDK.configure(with: wizCartConfigurations andDelegate: delegate)
```

## 4.2 Other configuration options

### 4.2.1 Client ids

**\*Must if not configured previously in the configurations object\***

In order to calculate incentives for a client, you must pass a unique client id.

WizCart will accept a unique String as a Client ID (UUID only), which we call as wizCartClientId. So you can use any new or existing User Management system.

### Parameters:

clientId: String	Unique identifier for a user.
isTempClientId: Bool	A boolean flag to indicate whether the aforementioned clientId belongs to a logged in user or not.

```
WizCartSDK.setClientId(clientId: String, isTempClientId: Bool)
```



## 4.2.2 Client traffic sources

**\*Must if not configured previously in the configurations object\***

In order to determine if the client is part of the WizCart incentive program and which “bucket” of incentives should be used for the end client, traffic sources should be passed to the `wizCartClient`.

### Parameters

<b>trafficSources:</b> <code>Array&lt;String&gt;</code>	An array of strings that determine the user group and type, for example: “Personal”, “User40”
--	---

```
WizCartSDK.setTrafficSources(trafficSources: Array<String>)
```

## 4.3 Delegate

### 4.3.1 Setting the delegate

The WizCart SDK uses the Delegate pattern to send responses to your application. If you wish to receive these responses, you need to set the sdk’s delegate. You can set the delegate either by the configuration object or with the `setDelegate` function. The delegate is a weak var, so there is no need to set the delegate to nil should you wish to set some other class as the delegate.

```
WizCartSDK.configure(with: andDelegate:)
```

### 4.3.2 Delegate functions

The `WizCartDelegate` is an Objective-C delegate, and the functions are optional should you wish to receive only specific events.

The result for any Track request or error if something happened along the way

```
trackResult(with trackResponse: WizCartTrackResponse?, and error: Error?)
```

The result for any Get Incentive request or error if something happened along the way

```
getIncentiveResult(with incentiveResponse: WizCartIncentiveResponse?, and  
error: Error?)
```

## 5. API

---

### 5.1 Track client journey

The WizCart ML requires information about the client journey in order to calculate the WizCart incentive. On application load at least one GetIncentive api call should be performed before calling the tracking apis.

The 'Cart' and 'Page type' information should be sent with each of the events below:

<a href="#">Cart</a>	The current client cart state.
<a href="#">Page type</a>	The current WizCartPageType the client is viewing.

#### 5.1.1 Page View

When to call:

Each time a client changes a PageType in his journey or a view is resumed.

params:

hasDisplayedWizCartIncentive	boolean if the Wizcart incentive was visible for the client
couponCode	What is the current wizCart coupon code that is eligible by the client. If the client is not eligible for a WizCart coupon set to nil
referrer	What application started the host app * First PageView only
pageReferrer	What was the Url that started the host application * First PageView only

```

WizCartSDK.trackPageView(pageView: WizCartPageType, cart: WizCartCart, referrer:
String?, pageReferer: String?, couponCode: String?, hasDisplayedWizCartIncentive:
Bool)

WizCartSDK.trackPageView(pageView: WizCartPageType, cart: WizCartCart, couponCode:
String?, hasDisplayedWizCartIncentive: Bool)

```

### 5.1.2 Incentive view Minimized/Maximized

When to call:

When the incentive is first seen and additionally every time a client engages with the incentive view (banner) minimize/maximize capability.

parameters:

couponCode	What is the current wizCart coupon code that is eligible by the client. If the client is not eligible for a WizCart coupon set to nil
------------	---

```

WizCartSDK.trackMinimizeView(pageView: WizCartPageType, cart: WizCartCart,
couponCode: String?)

WizCartSDK.trackMaximizeView(pageView: WizCartPageType, cart: WizCartCart,
couponCode: String?)

```

### 5.1.3 WizCart coupon copied

When to call:

When the coupon is copied to the clipboard by the client.

parameters:

couponCode	What is the current wizCart coupon code that is eligible by the client. If the client is not eligible for a WizCart coupon set to nil
------------	---

```
WizCartSDK.trackCopyCoupon(pageView: WizCartPageType, cart: WizCartCart, couponCode: String?)
```

### 5.1.4 Wizcart coupon added to cart

When to call:

When the coupon is used by the client during the purchase process.

parameters:

couponCode	What is the current wizCart coupon code that is eligible by the client. If the client is not eligible for a WizCart coupon set to nil
------------	---

```
WizCartSDK.trackInsertCouponToCart(pageView: WizCartPageType, cart: WizCartCart, couponCode: String?)
```

### 5.1.5 Client Cart updated

When to call:

Every time the client adds or removes items from the cart in the host application.

parameters:

product	The information about the product that was modified in the cart including the new quantity (0 if removed from cart)
---------	---

```
WizCartSDK.trackItemAddedToCart(pageView: WizCartPageType, cart: WizCartCart,
product: WizCartProduct)

WizCartSDK.trackItemRemovedFromCart(pageView: WizCartPageType, cart: WizCartCart,
product: WizCartProduct)
```

### 5.1.6 Purchase completed

When to call:

When the user completed a purchase.

parameters:

cartDiscountValueOfWizCart	If a WizCart incentive was used, insert the currency value of it
couponCode	What is the current wizCart coupon code that is eligible by the client. If the client is not eligible for a WizCart coupon set to null
addedOtherIncentivetypes	True if other incentives were used in this purchase, free shipping for example.
appliedCoupons	A list of coupons used in the purchase process. If Wizcart Coupon was used it should be in the list.

trackingId	The tracking id for this purchase
------------	-----------------------------------

```
WizCartSDK.trackCompletedPurchase(pageView: WizCartPageType, cart: WizCartCart,
trackingId: String?, couponCode: String?, appliedCoupons: [String]?,
cartDiscountValueOfWizCart: Double, hasAddedOtherIncentiveTypes: Bool)
```

### 5.1.7 Client Cart cleared

When to call:

Every time the client adds or removes items from the cart in the host application.

params:

wizCartCouponCode	What is the current wizCart coupon code that is eligible by the client. If the client is not eligible for a WizCart coupon set to null
-------------------	--

```
WizCartSDK.trackClearCart(pageView: WizCartPageType, cart: WizCartCart, couponCode:
String?)
```

## 5.2 Get client incentive

getIncentive is called in order to receive information if the client is eligible for an incentive at the current journey snapshot

The host application should call this method each time the client changes view or changes the content of his cart.

The getIncentive call is done asynchronously and the result will be returned via the Delegate.

params:

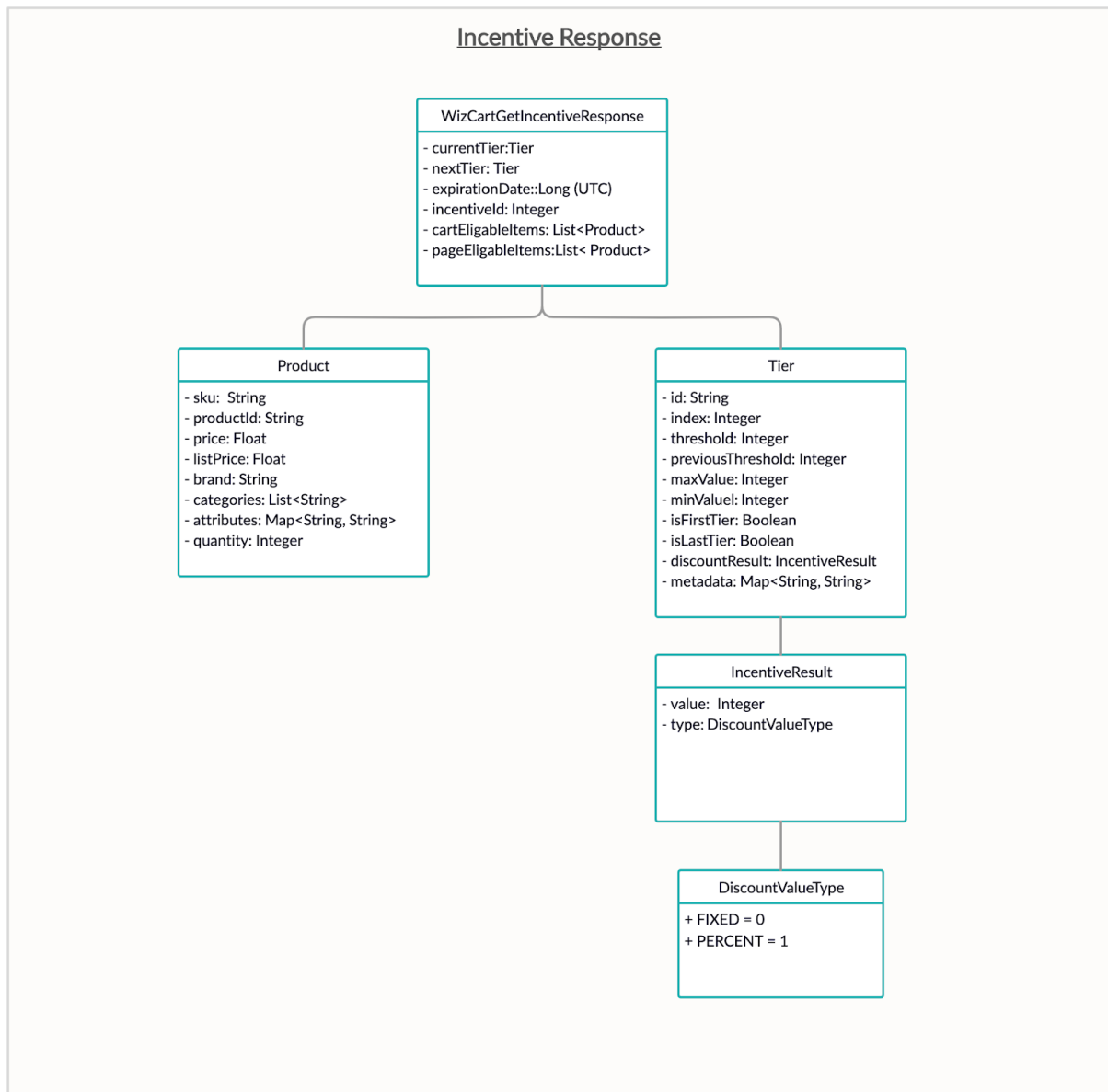
<a href="#">Cart</a>	The current client cart state.
<a href="#">Page type</a>	The current PageType the client is viewing.

```
WizCartSDK.getIncentive(Cart cart, PageType pageType);
```



## 6. Models

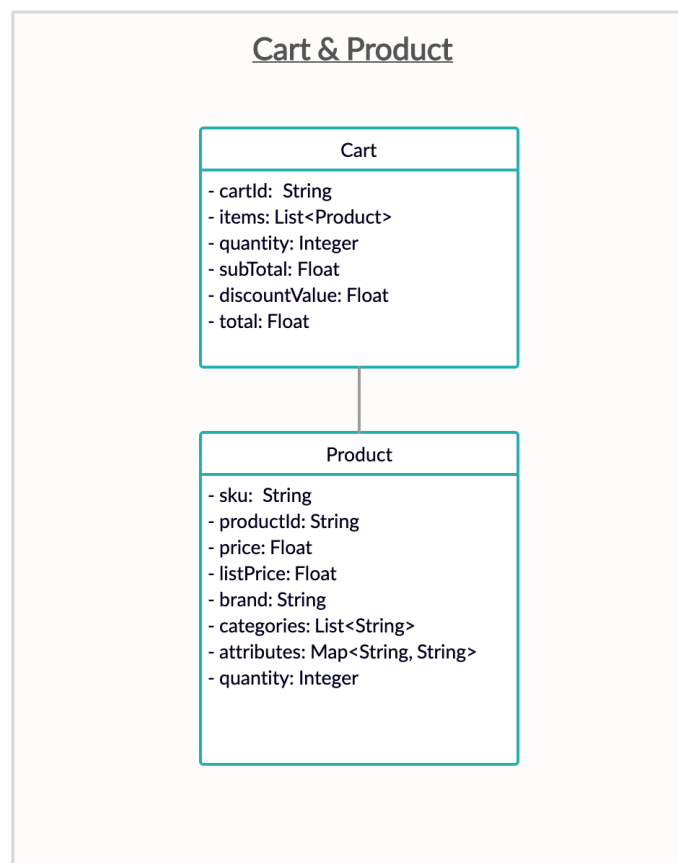
### 6.1 WizCartIncentiveResponse



## 6.2 Cart & Product

### Cart

1. cartId - The host app cart Id
2. items - list of all products in the cart
3. quantity - the total number quantities of all items
4. subtotal - the price of the cart before tax/discount/shipping...
5. discountValue - the total discount the client receives for this cart from all incentives
6. total - the final amount the client will spend on the cart



### Product

1. sku - sku of the product
2. productId - Product identifier of the product / can be the same as sku

3. price - The price of a single unit of the product after discounts (or same as list price)
4. listPrice - The price of a single unit of the product before discounts
5. brand - the brand name of the product
6. categories - A list of categories of the product (Baked goods / Meat / Clothing...)
7. attributes - A Map of descriptive attributes of the product (color / size ...)
8. quantity - The quantity of the product the client requests

## 7. Enums

---

### 7.1 PageType

The various pages that the WizCart should be called (Tracking & Incentive):

```
enum WizCartPageType
{
    case homePage
    case listingPage
    case productPage
    case cartPage
    case searchPage
    case registrationPage
    case paymentPage
    case confirmationPage
    case quickView
    case productSetPage
    case productSetQuickView
    case categoryPage
    case subcategoryPage
}
```