
# Mercury SDK for iOS
*Final Report*

Julien SALVI

September 19, 2011

***Abstract***

This document provides a short report about the work that has been done in the Mercury SDK for iOS project within Mogreet, Inc. company.

## Content


1. Introduction.....**3**
2. Prerequisites.....**3**
3. Timeline.....4
4. Features of Mercury
    1. Mercury
    2. System
    3. Transaction
    4. User
5. Presentation of the iOS Application
6. Difficulties encountered
7. Conclusion

# 1. Introduction

The Mercury Software Development Kit (SDK) provides the functions to implement the Mogreet API **Mogreet Messaging System** (MoMS). In order to be able to use Mercury on iPhone or iPad devices and applications, the SDK has to be ported using the iOS SDK provides by Mac and used on Xcode.
The main goal of this project was to implement Mercury on iPhone and build an application to test the functionalities of this SDK. This report will explain the porting on iPhone and the creation of the application.

# 2. Prerequisites

To carry out this project, the documentation of the API and my previous work on the Mercury SDK for Android have been used, and Xcode and the iOS SDK to implement the Mercury on iPhone. In order to download and use the iOS SDK you have to register as a Mac developer. As my computer version is 3.6.8 I cannot use the most recent version of Xcode and iOS, therefore I built Mercury thanks to iOS 4.3 and Xcode 3.2.6.
Obviously, as it is a iPhone application, the development has been done in Objective-C.

# 3. Timeline

The following section will describe the tasks I achieved during the development of this project:

- *Learning Objective-C* (**2 days**)
- *Handling of Xcode and iOS* (**2 days**)
- *Realization of the Graphical User Interface* (**1 day**)
- *Implementation of the API calls* (**3 days**)
- *Testing and polishing* (**1 day**)
- *Writing the documentation* (**1 day**)

# 4. Features of Mercury

The Mercury SDK is based on the Mogreet API MoMS. You can make different calls to MoMS using API functionalities. Three types of requests can be execute: System, User and Transaction.
This part of the report will detail the different calls which can be done.

## *3.1. Mercury*

You have to create a new mercury for calling requests. I had to implement the all-trusting trust manager in order to access to the HTTPS page. The creation of a new Mercury allows the internet connectivity. For every call, the user has to give his client ID and his token.

## *3.2. System*

The System request, through the Ping call, tests the connectivity to the MoMS API servers. All you need to send this request is your client ID and token. In case of success, you will receive a Pong message.

## 3.3. Transaction

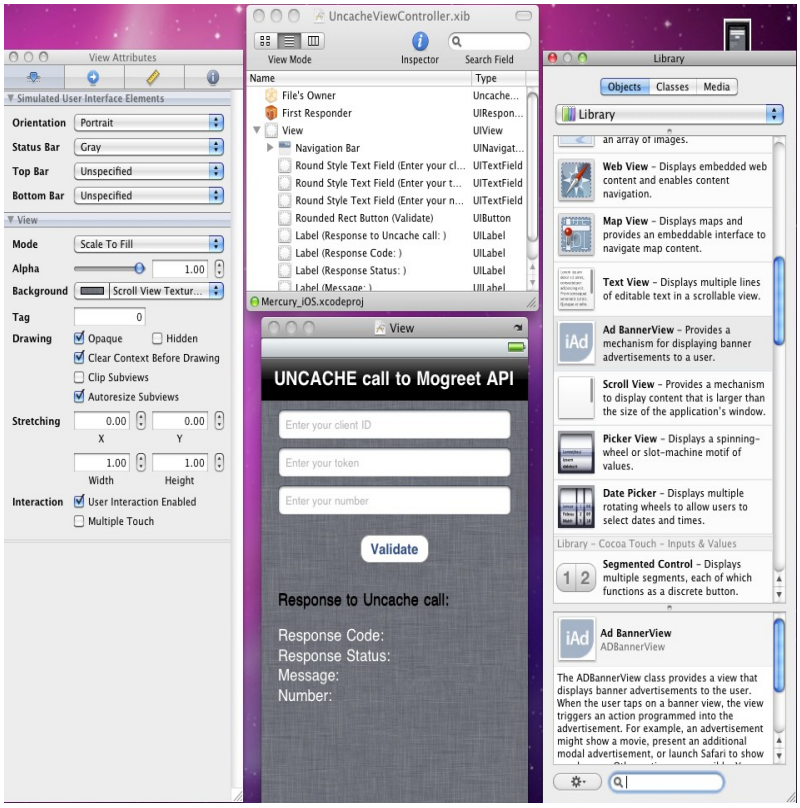
The Transaction request is building through two functions: Send and Lookup. The first one initiates a transaction and delivery of SMS or MMS. The user have to enter his client ID and token, the campaign ID, the mobile number you would like to send to and the mobile number you would like to send from, your message, the content ID. You may also add the sender name and the receiver name.
Concerning the second function, Lookup, it will give you the info, status and history of the requested transaction. The user has to enter the message ID of the requested message and the hash.

## 3.4. User

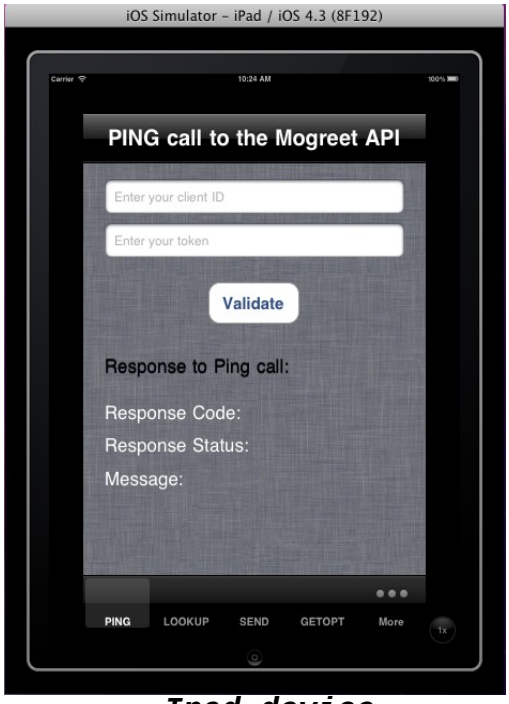
The user request is divided into five functions: Getopt, Info, Setopt, Transactions and Uncache.
The Getopt call will give you the opt in status of any mobile in entering your mobile number.
The Info call will give you the user carrier and the handset info if it available in giving your mobile number.
The Setopt call will set the opt is status of any mobile number in giving your number, the campaign ID and the status code.
The Transactions call will give you the user's transactions (open and closed) in giving your number.
The Uncache call will clear the user carrier and handset info from the Mogreet cache.

# 4. Presentation of the iOS Application

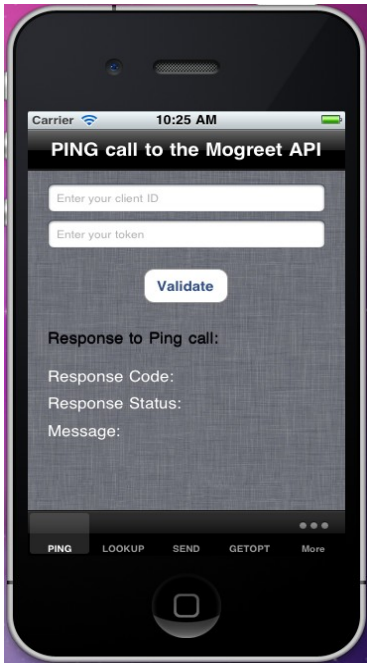
To develop iPhone applications with iOS SDK you have the use a MVC (Model View Controller) method. Indeed, first of all you have build the graphical interface thanks to Interface Builder. And then link with attributes of the classes.




Now let us talk about the application. It is similar to the Android version. I divided the application into eight tabs. Each tab represents an API call to the Mogreet servers. When you will compile the SDK, you can choose to have a virtual iPhone device or a virtual iPad device (pictures below). Therefore you could build an application for your iPad or iPhone using this SDK.

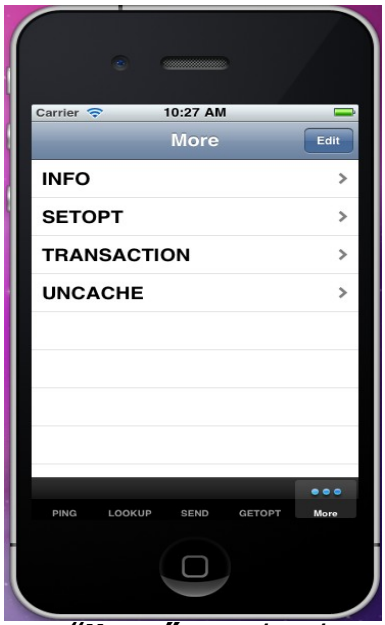



***Ipad device***




***iPhone device***

When you are in the application, you may choose the call you wish in clicking on the tab of your choice. Clicking the “More” tab, you will open the content for the INFO, SETOPT, TRANSACTION and UNCACHE calls.For example, let us do an INFO call to the Mogreet servers. You have to fill the three fields: enter your client ID, your token and your phone number. Then press the “Validate” button. If the the request is well structured the severs will return you a success response and the corresponding message plus the additional informations with respect to your call, otherwise the application will close.




***“More” content***


***INFO call***

## 5. Difficulties encountered

This project allows me to have my first experience with Objective-C, Xcode and iOS. At the beginning I has to familiarize myself in the Xcode IDE and iOS which were new for me.
Moreover, I had to find a good XML parser in order to process the data from the Mogreet API. This parser use the Xpath syntax so it is easy to build requests for the collecting data.
The major difficulty was to familiarize myself with the Objective-C language which was new for me. The language's syntax is a mix of Ruby, Java and C++ but after learning bases it becomes a very powerful language to build iPhone applications.

## 6. Conclusion

This project to port the Mogreet SDK to the iPhone/iPad platform has been a great way to improve my knowledge of the mobile Operating Systems. Moreover I learnt a new language and now, I am able to build iPhone applications.
The application is just an example to show how the MoMS API can be used so this can can be improve in many ways. And this SDK can be used for creating applications using the Mogreet API calls.