Source Documents - iOS SDK - MaaS Configuration

iOS SDK Integration Guide - (Core) MaaS Configuration

Version 1.3.2

ON THIS PAGE

This is Phunware's iOS SDK for the Core module. Visit http://maas.phunware.com/ for more details and to sign up.

Requirements

- iOS 6.0 or greater
- Xcode 6 or greater

Documentation

MaaS Core documentation is included in the Documents folder in the repository as both HTML and as a .docset. You can also find the latest documentation here: Core API iOS Reference

Installation

MaaS Core is a required dependency for all MaaS modules.

It's recommended that you add MaaSCore.framework to the 'Vendor/Phunware' directory, then add it to your Xcode project.

The following frameworks are required:

```
SystemConfiguration.framework
MobileCoreServices.framework
QuartzCore.framework
CoreTelephony.framework
Security.framework
```

The following frameworks are optional:

```
CoreLocation.framework
```

> CoreLocation is used for comprehensive analytics. Apple mandates that your app have a good reason for enabling location services. Apple will deny your app if location is not a core feature for your app.

After specifying the frameworks, you will need to add a linker flag to your build target.

Alternatively, you can install MaaSCore using CocoaPods:

```
// Add this to your Podfile:
```

```
pod PWCore
```

To do this:

1. Navigate to your build target.
2. Navigate to the Build Settings tab.
3. Find the Linking Section > Other Linker Flags.
4. Add "-ObjC" to Other Linker Flags.

You can now install additional MaaS modules.

Application Setup

At the top of your application delegate implementation (.m) file, add the following:

```
#import <MaaSCore/MaaSCore.h>
```

Inside your application delegate, you will need to initialize MaaS Core in the application:didFinishLaunchingWithOptions: method:

```
-
(BOOL)application:(UIApplicati
on *)application
didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions
{
 // These values can be found
for your application in the
MaaS portal
(http://maas.phunware.com/clie
nts).
    [MaaSCore
setApplicationID:@"APPLICATION
_ID"

setAccessKey:@"ACCESS_KEY"

signatureKey:@"SIGNATURE_KEY"

encryptionKey:@"ENCRYPT_KEY"];
// Currently unused. You can
place any NSString value here.

    // OPTIONAL: If you want
to enable logging in MaaS
Core, call the following:
    [MaaSCore
setLoggingLevel:MaaSLogLevel_D
ebug forService:[MaaSCore
serviceName]];
    ...
}
```

iOS 13 or later must add the following key to the application's info.plist

NSBluetoothAlwaysUsageDescription. The application needs access to the bluetooth system to take hardware measurements and report status of bluetooth.

MaaS Core uses the following third-party components. All components are prefixed so you won't have to worry about namespace collisions.

Component	Description	License

AFNetworking	A delightful iOS and OS X networking framework.	MIT
RNCryptor	CCCryptor (AES encryption) wrappers for iOS and Mac.	MIT
NSOperationStack	A LIFO (Last-In, First-Out) queuing extension for NSOperationQueue.	MIT
TMCache	Fast parallel object cache for iOS and OS X.	Apache 2.0

iOS SDK Integration Guide - (Core) MaaS Configuration

Version 1.3.2

ON THIS PAGE

This is Phunware's iOS SDK for the Core module. Visit http://maas.phunware.com/ for more details and to sign up.

Requirements

- iOS 6.0 or greater
- Xcode 6 or greater

Documentation

MaaS Core documentation is included in the Documents folder in the repository as both HTML and as a .docset. You can also find the latest documentation here: Core API iOS Reference

Installation

MaaS Core is a required dependency for all MaaS modules.

It's recommended that you add MaaSCore.framework to the 'Vendor/Phunware' directory, then add it to your Xcode project.

The following frameworks are required:

```
SystemConfiguration.framework
MobileCoreServices.framework
QuartzCore.framework
CoreTelephony.framework
Security.framework
```

The following frameworks are optional:

```
CoreLocation.framework
```

> CoreLocation is used for comprehensive analytics. Apple mandates that your app have a good reason for enabling location services. Apple will deny your app if location is not a core feature for your app.

After specifying the frameworks, you will need to add a linker flag to your build target.

Alternatively, you can install MaaSCore using CocoaPods:

```
// Add this to your Podfile:
pod PWCore
```

To do this:

1. Navigate to your build target.
2. Navigate to the Build Settings tab.
3. Find the Linking Section > Other Linker Flags.
4. Add "-ObjC" to Other Linker Flags.

You can now install additional MaaS modules.

Application Setup

At the top of your application delegate implementation (.m) file, add the following:

```
#import <MaaSCore/MaaSCore.h>
```

Inside your application delegate, you will need to initialize MaaS Core in the application:didFinishLaunchingWithOptions: method:

```
-
(BOOL)application:(UIApplicati
on *)application
didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions
{
 // These values can be found
for your application in the
MaaS portal
(http://maas.phunware.com/clie
nts).
    [MaaSCore
setApplicationID:@"APPLICATION
_ID"

setAccessKey:@"ACCESS_KEY"

signatureKey:@"SIGNATURE_KEY"

encryptionKey:@"ENCRYPT_KEY"];
// Currently unused. You can
place any NSString value here.

    // OPTIONAL: If you want
to enable logging in MaaS
Core, call the following:
    [MaaSCore
setLoggingLevel:MaaSLogLevel_D
ebug forService:[MaaSCore
serviceName]];
    ...
}
```

iOS 13 or later must add the following key to the application's info.plist

NSBluetoothAlwaysUsageDescription

MaaS Core uses the following third-party components. All components are prefixed so you won't have to worry about namespace collisions.

Component	Description	License

AFNetworking	A delightful iOS and OS X networking framework.	MIT
RNCryptor	CCCryptor (AES encryption) wrappers for iOS and Mac.	MIT
NSOperationStack	A LIFO (Last-In, First-Out) queuing extension for NSOperationQueue.	MIT
TMCache	Fast parallel object cache for iOS and OS X.	Apache 2.0