**Dosecast API Version 6.0.8**

**Documentation for**



**Overview**

The Dosecast API allows you to easily add supplement reminder functionality to your existing app for iPhone, iPod Touch, and iPad apps. It currently supports local notifications through the use of local iOS APIs and a server operated by Montuno Software.

The Dosecast API includes:

* Headers and static library (.a) files
* 3 types of resource files:
  + Reminder sound files (.caf)
  + A Dosecast-resources.bundle file containing all images, compiled NIBs, and strings used in the app

There are 4 build variants of the 2 static libraries: one for the iPhone Retina 4-inch simulator and one for devices, and each of these comes in a debug and release build. All variants were built:

* For a universal binary
* To support armv7, armv7s, and arm64 architectures
* With Xcode 5.1.1 and the iOS 7.1 SDK
* With Base SDK = 7.1 and iPhone OS Deployment Target = 6.0
* With ARC enabled

**Resources**

*Reminder sound files*

The reminder sound files must be located in the top-level of the main application bundle. This is to enable the OS to play them if a reminder is received while the app is not active.

*Dosecast-resources.bundle*

This bundle file must be located in the top-level of the main application bundle. It contains:

* All icons used in the app in both regular and high-resolution (@2x) sizes. In particular:
  + The Background images are used behind all view controllers
  + The Gear images are used in the toolbar for opening the settings view
  + The Mail images are used in the toolbar for emailing the supplement list
  + The Log images are used in the toolbar for viewing the history log
* Compiled NIBs for both iPhone/iPod and iPad devices. (iPad device NIB filenames are appended with ‘-iPad’ to identify them.)
* The Dosecast.strings file for the *en* locale

The Dosecast.strings file contains all the user-visible strings appearing in the app. Each is commented with a hint as to where the string appears. In particular:

* App colors are represented as strings. All customizable app colors may be set via strings whose key begins with ‘Color’. All colors are represented as ‘R G B A’ where each channel is a float between 0 and 1.
* URLs. There are 3 important URLs that must be set in order for Dosecast to communicate with the server:
  + ServerProtocolDebug/Release: the protocol to use
  + ServerHostDebug/Release: the server host to use
  + ServerPathDebug/Release: the server path to use

**Supplement Types**

The Dosecast API can be extended with new supplement types, each of which can have:

* A name
* A set of ordered inputs, each of which can be required or optional and can have:
  + A name
  + An input type: either a picklist or a numeric quantity
  + For a picklist:
    - A set of possible values
  + For a numeric quantity:
    - A number of total digits
    - A number of decimals
    - A set of possible units
* A dosage description that appears in reminders and summarizes the input values

The GNC-customizations.a static library has been compiled with the following supplement types built-in:

| **Supplement**  **Type Name** | **Required**  **Primary Input** | **Optional**  **Secondary Input** |
| --- | --- | --- |
| Pill | Name: Pills per Serving  Type: Numeric  Units: None  Total digits: 3  Decimals: 1 | Name: Pill Strength  Type: Numeric  Units: g, mg, mcg, units, IU  Total digits: 6  Decimals: 2 |
| Pak | Paks per Serving  Type: Numeric  Units: None  Total digits: 2  Decimals: 0 | Name: Pills per Pak  Type: Numeric  Units: None  Total digits: 2  Decimals: 0 |
| Drink | Drink Volume  Type: Numeric  Units: mL, tsp, tbsp, oz  Total digits: 4  Decimals: 1 | Name: Drink Strength  Type: Numeric  Units: g/mL, mg/mL, mg/tsp, mg/tbsp, mg/oz  Total digits: 6  Decimals: 2 |
| Shot | Number of Shots  Type: Numeric  Units: None  Total digits: 2  Decimals: 0 | Name: Shot Strength  Type: Numeric  Units: g/mL, mg/mL, mg/tsp, mg/tbsp, mg/oz  Total digits: 6  Decimals: 2 |
| Chew | Number of Chews  Type: Numeric  Units: None  Total digits: 2  Decimals: 0 |  |
| Gel | Number of Gels  Type: Numeric  Units: None  Total digits: 2  Decimals: 0 |  |
| Scoop | Scoops per Serving  Type: Numeric  Units: None  Total digits: 2  Decimals: 0 | Name: Scoop Volume  Type: Numeric  Units: mL, tsp, tbsp, oz, g  Total digits: 4  Decimals: 1 |
| Bar | Number of Bars  Type: Numeric  Units: None  Total digits: 2  Decimals: 0 | Name: Bar Volume  Type: Numeric  Units: mL, tsp, tbsp, oz  Total digits: 4  Decimals: 1 |
| Serving | Number of Servings  Type: Numeric  Units: None  Total digits: 2  Decimals: 0 | Name: Serving Strength  Type: Numeric  Units: mL, tsp, tbsp, oz  Total digits: 4  Decimals: 1 |

**User Registration**

The Dosecast API requires that each user be registered with the server for first-time use. This may be done asynchronously at any time as long as an internet connection is available. Once registered, the registration information is saved in a local file and automatically accessed on startup.

**Integrating the Dosecast API**

Follow these steps in order to integrate the Dosecast API:

1. Place the reminder sound files and Dosecast-resources.bundle file at the top-level of the main application bundle
2. For all xcode project configurations, add the following linker flags:

-all\_load –weak\_framework UIKit –ObjC

1. Link to the following frameworks:

* AddressBook
* AddressBookUI
* UIKit
* CoreData
* QuartzCore
* StoreKit
* AudioToolbox
* CoreGraphics
* Foundation
* MessageUI
* SystemConfiguration
* AdSupport
* Security

Also link to both static libraries (libDosecast\_API.a and libGNC\_customizations.a)

1. Decide where to embed the Dosecast main view controller. We recommend you add it as a tab in the tab bar controller.
2. Implement the DosecastAPIDelegate protocol. There are the following methods to implement:
   1. handleDosecastUIInitializationComplete

This is the callback for when UI initialization is complete after the Dosecast instance has been constructed. **No DosecastAPI calls may be made, and no DosecastAPI properties may be invoked, until after this delegate call.**

* 1. handleDosecastRegistrationComplete

This is the callback for when Dosecast registration completes.

* 1. displayDosecastComponent

This is the callback for when the Dosecast component must be made visible. If Dosecast is embedded in a UITabBarController or other UI component, this component must be made visible at the time of this call if it is not already.

* 1. isDosecastComponentVisible

This is the callback to find out if the Dosecast component is visible. If Dosecast is embedded in a UITabBarController or other UI component, return whether the component is active/selected.

* 1. disallowDosecastUserInteractionsWithMessage

This is the callback for when Dosecast user interactions should be disallowed and a message displayed to the user. When user interactions should be allowed again, allowDosecastUserInteractionsWithMessage will be called.

* 1. updateDosecastMessageWhileUserInteractionsDisallowed

This is the callback for when a message changes while user interactions are disallowed.

* 1. allowDosecastUserInteractionsWithMessage

This is the callback for when Dosecast user interactions should be allowed again.

* 1. disallowDosecastUserInteractionsWithMessageAndProgress

This is the callback for when Dosecast user interactions should be disallowed and a message displayed to the user with intermediate progress.

* 1. updateDosecastProgressWhileUserInteractionsDisallowed

This is the callback for when an intermediate progress update occurs.

* 1. updateDosecastProgressMessageWhileUserInteractionsDisallowed

This is the callback for when a message changes while progressing and user interactions are disallowed.

* 1. allowDosecastUserInteractionsWithMessageAndProgress

This is the callback for when Dosecast user interactions should be allowed again.

* 1. getUINavigationController

This is the callback to return the UINavigationController used for the Dosecast main view controller.

1. Construct an instance of the GNCDosecastAPI class on app startup using the initializer below:

-(id)initWithDelegate:(NSObject<DosecastAPIDelegate>\*)del

launchOptions:(NSDictionary\*)launchOptions

The meaning of the parameters are:

* launchOptions: these are the launchOptions passed in the Application Delegate’s didFinishLaunchingWithOptions method.

**Note: the GNCDosecastAPI must be instantiated before any other classes or methods provided in the Dosecast API are invoked.**

1. Pass-through each of the Application Delegate’s methods below to the corresponding GNCDosecastAPI method:
   1. didRegisterForRemoteNotificationsWithDeviceToken
   2. didReceiveLocalNotification
   3. didReceiveRemoteNotification
   4. didFailToRegisterForRemoteNotificationsWithError
   5. applicationDidEnterBackground
   6. applicationWillEnterForeground
   7. applicationWillResignActive
   8. applicationDidBecomeActive
   9. applicationWillTerminate
2. Once the DosecastAPIDelegate’s handleDosecastUIInitializationComplete method is called, register the user for first-time use (if not already done) using the GNCDosecastAPI instance. This should only be done once initialization is complete. To do this:
   1. Use the userRegistered property to determine if the user has been registered yet.
   2. If the user hasn’t been registered, use the internetConnection property to determine if an internet connection is available.
   3. If the user hasn’t been registered and an internet connection is available, call the asynchronous registerUser method. The DosecastAPIDelegate’s handleDosecastRegistrationComplete method will be called when registration completes.
3. Once the DosecastAPIDelegate’s handleDosecastUIInitializationComplete method is called, display the main view controller using the mainViewController property of the GNCDosecastAPI instance and start allowing UI interactions on it. This should only be done once the user has been successfully registered.
4. Once the DosecastAPIDelegate’s disallowDosecastUserInteractionsWithMessage or disallowDosecastUserInteractionsWithMessageAndProgress methods are called, stop user interactions from occurring with the Dosecast component while displaying a message to the user – until the corresponding method is called to allow user interactions again. If the disallowDosecastUserInteractionsWithMessageAndProgress is called, a progress bar should also be displayed. For convenience, these methods may use the SpinnerViewController and ProgressViewController classes provided to perform these functions. **If used, however, SpinnerViewController and ProgressViewController must be instantiated after the GNCDosecastAPI is instantiated.**