

# Cardinal Health - Mobile Change Jar iOS App Challenge - Deployment Guide

## Revision History

Author	Revision Number	Date
Alexander Volkov	1.0	June 25, 2015

## Deployment Instructions

1. Deployment Dependencies
2. Organization of Submission
3. 3rd party Libraries
4. Configuration
  - 4.1. CardinalHealth/Supporting Files/configurations.plist
  - 4.2. NodeJS application configuration
  - 4.3. iOS application configuration
5. Deployment Instructions
  - 5.1. Configure NodeJS and iOS application
  - 5.2. Launch NodeJS application
  - 5.3. Build and run the app in a simulator or on a real device
6. Starting
7. Verification
  - 7.1. Notes
8. Resource Contact List

# Deployment Instructions

## 1. Deployment Dependencies

Before performing a deployment, it is assumed that the following have been set up:

- Xcode 6.3.1+
- OS X 10.10 or above
- iOS SDK 8.3 or above
- iPhone device with iOS 8.0+

## 2. Organization of Submission

- *CardinalHealth/CardinalHealth.xcodeproj* – Xcode project to open.
- *CardinalHealth/external* – this directory contains external sources used within the application
- *test\_server* – this directory contains NodeJS application that emulates real servers
- *docs* – this directory contains the documents for this application, including this deployment guide

## 3. 3rd party Libraries

**SwiftJSON** - <https://github.com/SwiftyJSON/SwiftyJSON>

SwiftJSON makes it easy to deal with JSON data in Swift. Currently used version is “commit 321bedc299ad146be97b9b30fbcfebcf617b3672”.

**Alamofire** - <https://github.com/Alamofire/Alamofire>

Alamofire is an HTTP networking library written in Swift. Currently used version is “commit 333bf874b85a12c448f9d2419293e3d47e5416c2”.

## 4. Configuration

For this challenge you need to configure only the options described in [4.2. NodeJS application configuration](#) and [4.3. iOS application configuration](#) sections.

### 4.1. CardinalHealth/Supporting Files/configurations.plist

**configuration.plist** configuration file is located in “*CardinalHealth/Supporting Files*” group in Xcode project.

The file contains the following options:

- **apiBaseUrl** - the base URL to the server. Default: <http://localhost:8888/>
- **loggingLevel** (Number) - the logging level for Logger class. Default value: 1 (INFO)

### 4.2. NodeJS application configuration

You need to configure the port number for the local server that will be launched by the *test\_server/runLocalServer.sh* script. By default 8888 port is used, so if you have no services on this port you do NOT need to change it. The server will be binded to <http://localhost:8888> . If you

will verify the app on a real device you need to provide your computer real IP address. In the next section this IP will be referenced as `<ip_address>`.

### 4.3. iOS application configuration

You need to specify at least the URL of the local NodeJS application for REST API. By default it's <http://localhost:8888/>.

If you plan to test the app on a real device you need to change the value in ***apiBaseUrl*** property in ***configuration.plist*** (*CardinalHealth/Supporting Files/configuration.plist* in Xcode project) to `http://<ip_address>:8888/`, where `<ip_address>` is the IP address of your Mac where NodeJS application is launched.

## 5. Deployment Instructions

### 5.1. Configure NodeJS and iOS application

You should configure NodeJS and iOS applications. Please follow the instructions in [4. Configuration](#)

You must have NodeJS installed on your system. Follow the instructions on <https://nodejs.org> website to install command line nodejs utility if it's not yet installed.

### 5.2. Launch NodeJS application

Due to iOS application connects to local NodeJS application you need to launch the provided NodeJS application before launching iOS application.

Follow the instructions:

1. Open console on your Mac
2. Enter the following commands and execute:

```
$ cd test_server/  
$ ./runLocalServer.sh
```

You will see the next output that means that NodeJS application is launched successfully:

```
Starting local HTTP server...  
DONE
```

### 5.3. Build and run the app in a simulator or on a real device

To build and run the app in a simulator or on a real device you will need to do the following:

1. Open *CardinalHealth/CardinalHealth.xcodeproj* in Xcode
2. Select a real device (when connected) or a simulator from the top left dropdown list.
3. Click menu Product -> Run (Cmd+R)
4. Follow the verification steps in [7. Verification](#)

## 6. Starting

See [5. Deployment Instructions](#) on how to build and run the application.

## 7. Verification

Once you finished the [5. Deployment Instructions](#) and have NodeJS and iOS application launched follow the provided video (docs/Video.swf) and the rest of the section to verify the app. The provided video shows the main happy paths.

### 7.1. Notes

- When you reach the last or the first idea through swiping Idea Summary screen the screen will shake to indicate that there is no next/previous idea in the list.
- You can tap on the highlighted areas in Home Screen overlays to continue to next instructions (works the same as if you tap "Next/Done" button). Also you can swipe right to continue.
- Profile icons load is implemented. Once the json response will have correct URL for "submitterIconUrl" property the app will load and show the icons without any changes. Search test\_server/JSON\_responses/ideas.json for "submitterIconUrl" values. If the value not starts with "http://", then it is treated as an app embedded image name (for demonstration).
- There are two options in the code that can switch the app behaviour and help to verify:
  - OPTION\_MARK\_MY\_IDEAS\_AS\_NOT\_NEW in IdeasDataSource.swift
  - PTION\_MARK\_IDEAS\_AS\_NOT\_NEW\_WHEN\_GO\_BACK in IdeasListViewController.swift

Follow the related documentation in code.

## 8. Resource Contact List

Name	Resource Email
Alexander Volkov	Through TopCoder Member Contact