

Interactive Chart for iOS SDK

Installation & Package Overview / August 26, 2015

Package Contents

LiveCharts consists of the following files:

- Getting Started.pdf (This file)
- LiveCharts Overview.pdf A high-level overview of LiveCharts
- LiveCharts Example An Example application showing the use of the LiveCharts SDK
- LiveCharts Documentation An Xcode compatible documentation set for LiveCharts
- LiveCharts.framework LiveCharts framework (Includes header and library)

Installation

LiveCharts requires the following frameworks to be added to your project:

1. LiveCharts.framework
2. ShinobiCharts.framework
3. QuartzCore.framework
4. OpenGLES.framework
5. libc++.dylib (for Swift, this is libc++.tbd)

LiveCharts is ARC compliant.

The package has been tested on Xcode 6.4 and supports iOS 7.0 and up.

LiveCharts Documentation

The documentation contains auto generated SDK documentation from the headers of the framework. To view this documentation, open the HTML folder and then open index.html.

Getting Started

Be sure you have the frameworks above set up in your project prior to trying these steps. Make sure they are included in your target's Link Binary With Libraries build phase.

To get a basic chart up and running:

1. Make sure to add the `-all_load` flag to **Other Linker Flags** in the project's build settings.
2. Drag the Livecharts.bundle file from LiveCharts.framework/Resources into the project navigation pane.
3. Open your own ViewController where you want to add the chart view and import the framework.
4. Set your view controller to conform to protocols `NSURLSessionDelegate`, `LChartViewDataSource`, `LChartViewDelegate`, and `LChartLogReporterDelegate`.
5. Set your view controller as the delegate for `LChartViewDataSource`, `LChartViewDelegate`, and `LChartLogReporterDelegate`.
6. Establish the configuration.
7. Establish the chart and symbol information URLs. (This is a necessary step, failure to do so will cause the framework to stop with an assertion.)
8. Create the chart view.
9. Build the chart request.
10. Call the `drawCharts` method.

Example Code

This distribution comes with two starter view controller classes (one Swift, one Objective-C) named `SampleViewController`. Either `SampleViewController` can be used as the initial view controller for a project and it will display a three-month chart for Apple (AAPL).

The SDK comes complete with two example applications (one Swift, one Objective-C) that are completely functional where you can see the entire SDK in operation. Use either as a guide when you want to see how a specific design pattern works.