How to setup a development environment and create a project in NativeScript

1. Setup a development environment.

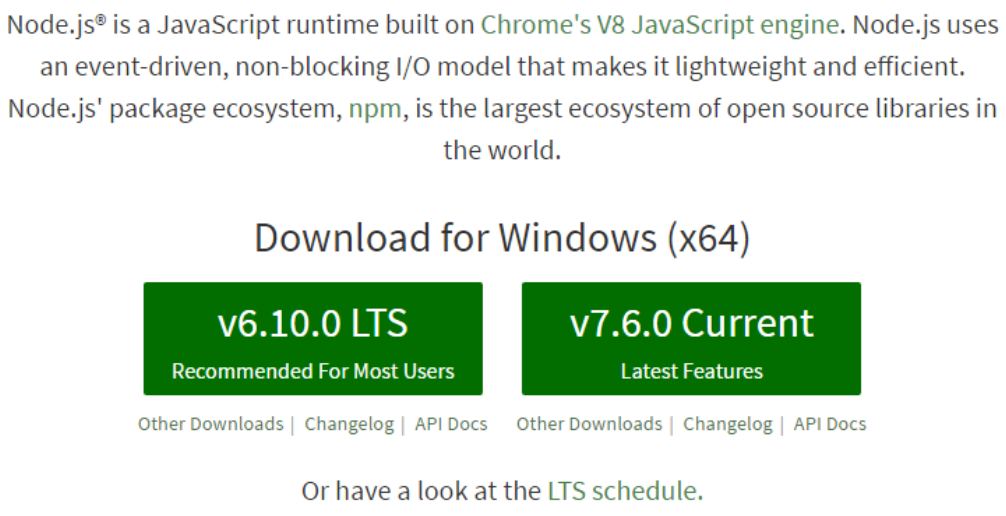
Step 1: Install Node.js

- What is Node.js?

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalablenetwork applications.

Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Please click this url: https://nodejs.org/ and set up this.



If you click this url, you can see this.

Please click v6.10.0 LTS and set up.

It takes a few minutes to finish.

Step 2: Install the NativeScript CLI

- What is NativeScript CLI?

CLI means command line interface.

- open your terminal and command prompt and execu te the following command to install the Nativescr ipt CLI from npm.

npm install -g nativescript

Step 3: Install iOS and Android

When you build with NativeScript you’re buildingt ruly native iOS and Android apps, and as such, yo u need to set up each platform you intend to buil d for on your development machine.

To ease the pain of installing all of these requi rements manually, the NativeScript CLI providesq uick-start scripts for Windows and macOS that han dle the necessary setup for you automatically.

Let’s look at how they work.

- Windows.

Please open the command prompt and execute the fo llowing command.

@powershell -NoProfile -ExecutionPolicy Bypass - Command "iex ((new-object net.webclient).Downloa

dString('https://www.nativescript.org/setup/win'))"

-MacOS (I recommend this system for iOS)

Copy and paste the script below into your terminal and press Enter:

sudo ruby -e "$(curl -fsSL https://www.nativescript.org/setup/mac)"

Much like the Windows script, the macOS script needs administrative access to run some commands using sudo; therefore, you may need to provide your password several times during execution.

The macOS script also may take some time to complete, as it’s installing the dependencies for both iOS and Android development.

When the script finishes, close and restart yourterminal.

Step 4: Verify the setup.

Once finished installing NativeScript and its dependencies, run the tns doctor command, which will check for any issues with installation.

tns doctor

2. Create simple project.

- create project.

  Open the terminal and run this command.

tns create Scichartdemo(project name)

cd Scichartdemo

- running apps

If you want Android emulator, run this command in terminal

tns run Android

For iOS emulator, run this

tns run iOS

3. Import plugin using pod file in this project

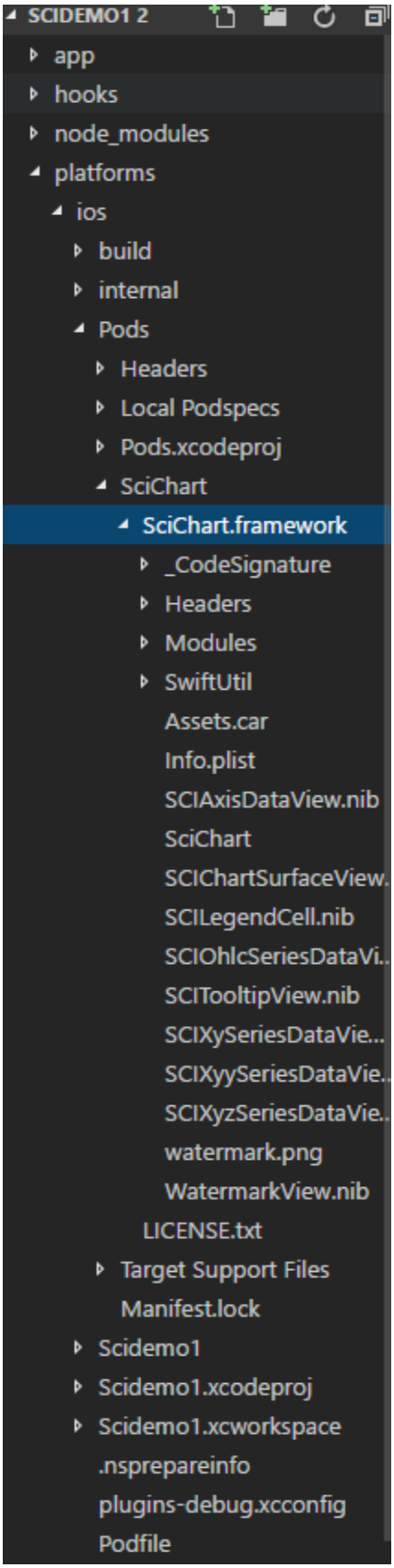
I have already named this plugin scichart-ui

tns add plugin <path to plugin> (for example /Users/Aloysha/Documents/work/scichart-ui)

Then, it will add in node module and platform/ios in my project.

Here is structure of this project importing

Scichart framework.


SCIDEMO1 2
app
hooks
node_modules
platforms
ios
build
internal
Pods
Headers
Local Podspecs
Pods.xcodeproj
SciChart
SciChart.framework
_CodeSignature
Headers
Modules
SwiftUtil
Assets.car
Info.plist
SCIAxisDataView.nib
SciChart
SCIChartSurfaceView.
SCILegendCell.nib
SCIOhlcSeriesDataVi..
SCITooltipView.nib
SCIXySeriesDataVie...
SCIXyySeriesDataVie..
SCIXyzSeriesDataVie..
watermark.png
WatermarkView.nib
LICENSE.txt
Target Support Files
Manifest.lock
Scidemo1
Scidemo1.xcodeproj
Scidemo1.xcworkspace
.nsprepareinfo
plugins-debug.xcconfig
Podfile