**Documentation**

This provides a multiple on demand service based on nearby locations, by using this app simply you view and book a service or provide a service to the customers.

This is an on demand nearby service finder and booking system, it’s very powerful, flexible and most affordable that can be used for multiple type business. Truly sell Service offers a bouquet of services like pest control, carpentry, housekeeping, kitchens repairing, lighting, painting, wooden work, plumbing, refrigeration and washing machine repair and cleaning, air-conditioning repair and maintenance, and electrical services. It’s fully responsive, easy to use and user friendly applications.

This can be used in multiple ways, it has a multiple category and subcategories list. Customers can book a service within a minute based on your convenient time. Payment can be collected through stripe. If you are looking for any other payment gateways it can be under customizations.

**Product Features**

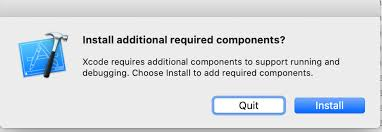
* Package contains Web, Android, and iOS platforms
* Professional Design on Web and Mobile Apps
* Subscription based App usage option
* OTP Based Login
* Ingrate with Firebase for push notification
* Simple and Powerful Chat Option
* Fully Responsive
* Powerful Admin Panel
* Add Multiple Service
* Add/Edit/delete Service details
* Dynamic Pricing
* Simple Calendar and Time view
* Powerful Location based Service
* Easy and user friendly Admin panel
* History view option
* Multiple Payment gateways
* Wallet concept
* And Much More....

**Installation Procedure for iOS App**

**Setup Project**

**iOS (Xcode 10.2)**

1. Download Xcode 10.2 if you still don't have follow the below link for downloading: <https://stackoverflow.com/questions/10335747/how-to-download-xcode-dmg-or-xip-file>
2. After downloading Xcode, double click the downloaded dmg file. It will take a few minutes for extract.
3. After this extraction, It will ask for installing components. Please click Install. It will download the components and it will install the Xcode tool.

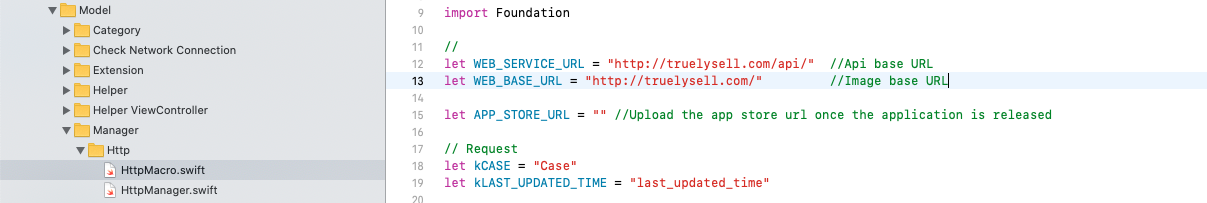


**Configuration:**

**Change API and Image Base URL:**

Open iOS project by clicking the file with extension .xcworkspace from the source file.

Then Click MVC > Model > Manager > Http > HttpMacro.swift > Update the API base url and Image base url.



**Change Websocket url:**

To change the websocket url of the app, MVC > Model > Manager > Http > HttpMacro.swift > Update the websocket url here.

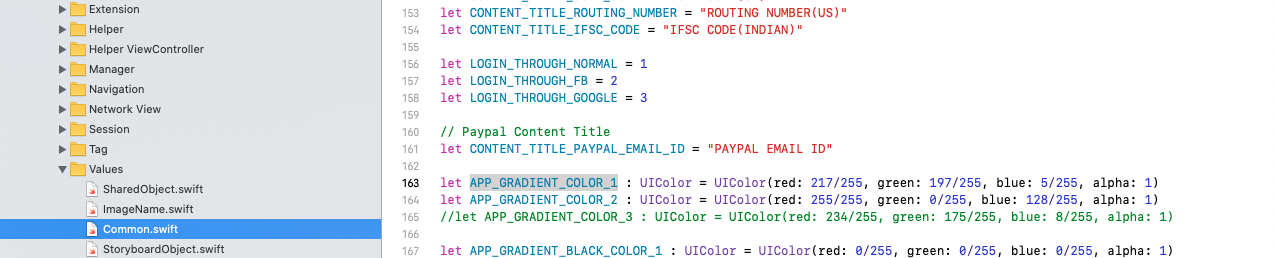


**Change App Color:**

To change the color of the app, Click MVC > Model > Values > Common.swift.If you change the Color here, the whole application will be reflected.

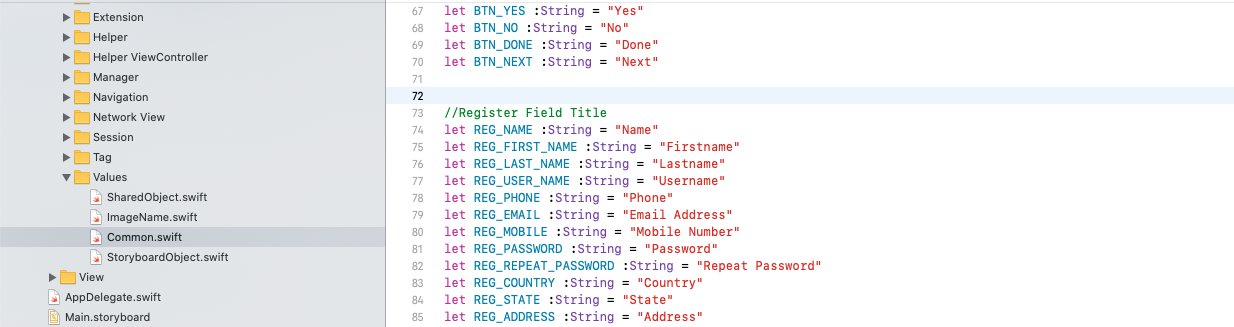


To change the Gradient color in app, Click MVC > Model > Values > Common.swift. For gradient color kindly convert the hex color code to RGB and add it. If you change the Gradient color here, the whole application will be reflected.



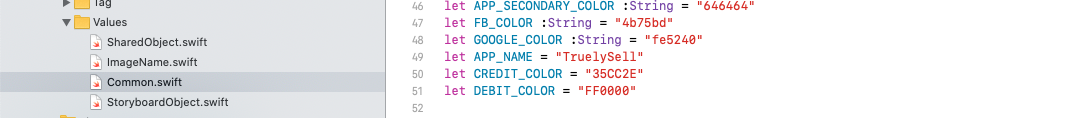
**Change Text Values:**

To change Text values in the application, Click MVC > Model > Values > Common.swift. From this file, you can change the common texts. If you change the text here, the whole application text will be changed.



**Change App Common Name:**

To change the common app name in the application, Click MVC > Model > Values > Common.swift. From this file, you can change the common app name. If you change the app name here, the whole application text will be changed.

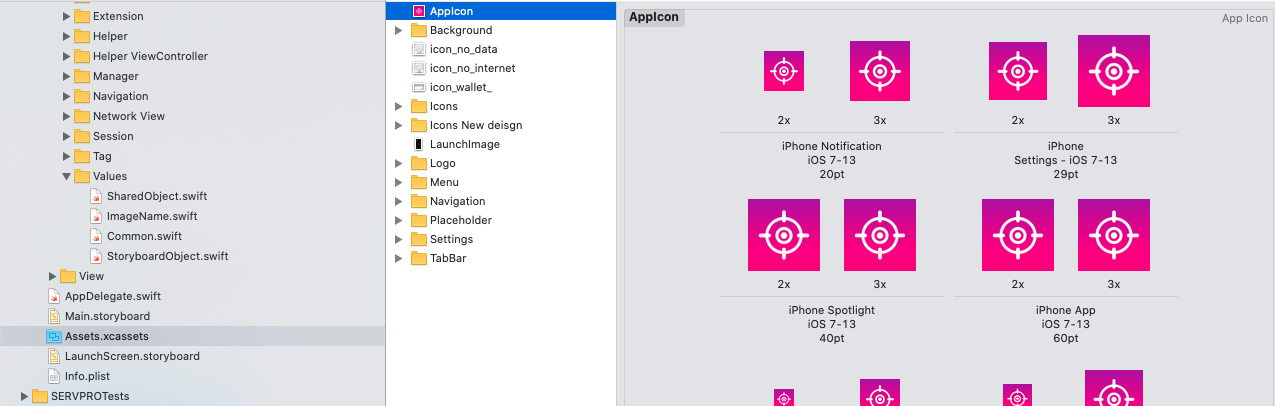


**Change App Icons:**

First design the App Icon in 1024\*1024 size without corner radius.

Then resize the app using this website <https://makeappicon.com/>.

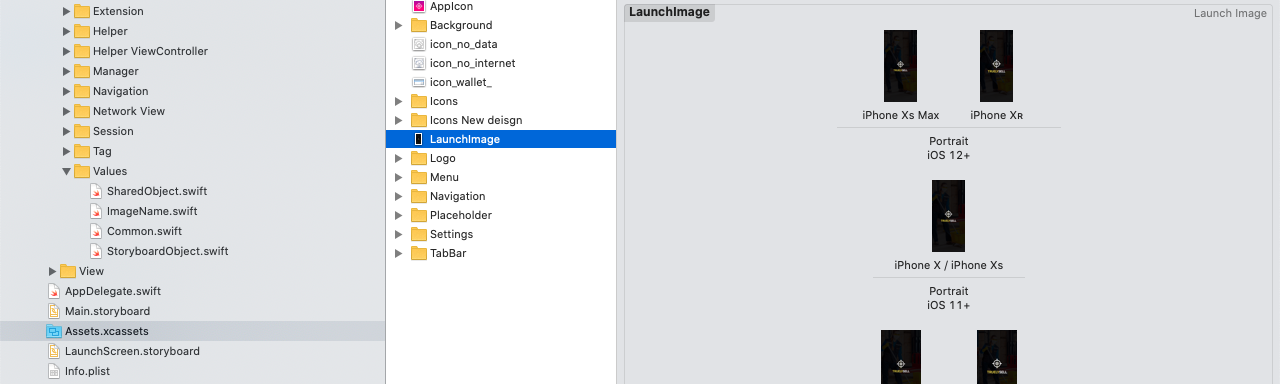
Upload the resized App Icons with its corresponding sizes, Open project file, Click Assets.xcassets > AppIcon

****

# Change Splash Screen:

First design the Splash screen according to the sizes required in the xcode.

Upload the resized Splash screen with its corresponding sizes, Open project file, Click Assets.xcassets > LaunchImage

****

# Configure APNs Push Notification:

For creating the APNs certificate first you need to purchase an Apple developer account.

Once the account is subscribed then for creating APNs(.p12 file) kindly follow this link <https://stackoverflow.com/questions/21250510/generate-pem-file-used-to-set-up-apple-push-notifications>

Please follow the steps in the above attached link. First create the APNs certificate(.p12). Then convert the .p12 certificate file to .pem file. This certificate conversion is also available in the above link. Once the .pem certificate is created then configure the certificate in the backend for receiving push notification in mobile application.

# Configure Google map:

First login to Google API console and create a project in the dashboard.

After creating the project kindly click Credentials from the left menu.

# 

From this option Credentials, Create the API keys for the project. For creating the API key kindly follow this link <https://developers.google.com/maps/documentation/ios-sdk/get-api-key> . Just create the API key alone from these steps from the link. Don’t install pods as per the steps given in the link.

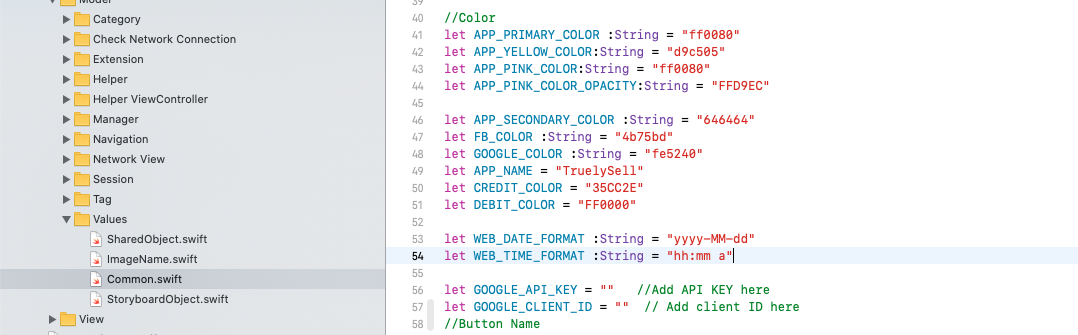
Then likewise create OAuth 2.0 Client IDs from the same Credentials menu. For creating the OAuth 2.0 Client IDs kindly follow this link <https://developers.google.com/identity/protocols/oauth2/native-app> .

After creating the OAuth 2.0 Client IDs, Kindly download the json file and replace it in the source code. You can see the download option as in the image below.



You can find the file with this format client\_xxxxxxxxxxxxxx.apps.googleusercontent.com.plist. And this file needs to be replaced.

Once the OAuth 2.0 Client IDs and API keys are created then kindly add those keys in the file. For adding these keys Open source code, Click MVC > Model > Common.swift.



# Create Create certificate for development and distribution:

Certificate creation is of two types, One will be a Development certificate which will be used while running the project in physical devices. And another one will be a Distribution certificate which will be used while releasing the application in the App Store.

For creating these two certificates first you need to create a CSR file in the Keychain tool from your mac machine. For creating this CSR certificate follow this link <https://stackoverflow.com/a/45504269>

After creating the CSR file,

For Development certificate follow this link, <https://docs.nativescript.org/sidekick/user-guide/code-signing/code-signing-for-ios/create-development-certificate>

For Distribution certificate follow this link, <https://docs.nativescript.org/sidekick/user-guide/code-signing/code-signing-for-ios/create-distribution-certificate>

# System requirements:

To run the product, You need the followings

|  |  |
| --- | --- |
| **Operating Systems** | Mac OS 10.14 |
| **Software** | Xcode 10.2 |

# 

# Prepare to Deploy

And yes finally, To deploy your app to Apple App Store you need to generate the .ipa file and you need to upload it in iTunes connect. For this big process do follow the below video carefully and upload it in App Store

<https://www.youtube.com/watch?v=YPLs3xrDcm0>