Google Places

Get data from the same database used by Google Maps and Google+ Local. Places features over 100 million businesses and points of interest that are updated frequently through owner-verified listings and user-moderated contributions.

Get started :

Step 1: Get the latest version of Xcode

To build a project using the Google Places API for iOS, you need **version 7.0** or later of Xcode.

Step 2: Install the API

Install manually

1. Download the SDK source files:

- GooglePlaces-2.0.1
- GoogleMaps-2.0.1
- GooglePlacePicker-2.0.1

GooglePlacePicker-2.0.1 is only needed if you are using the place picker.

2. Unpack the source files.
3. Launch Xcode and either open an existing project, or create a new project. If you're new to iOS, create a *Single View Application*, and disable *Use Storyboards* and enable *Use Automatic Reference Counting*.
4. Remove any Maps bundles from previous releases from your project.
5. Drag the following bundles into your project (when prompted, select **Copy items if needed**):

- GooglePlaces-2.0.1/Frameworks/GooglePlaces.framework
- GoogleMaps-2.0.1/Subspecs/Base/Frameworks/GoogleMapsBase.framework

If you are using the place picker, drag the following bundles into your project (in addition to the previously listed bundles):

- GooglePlacePicker-2.0.1/Frameworks/GooglePlacePicker.framework
- GoogleMaps-2.0.1/Subspecs/Maps/Frameworks/GoogleMapsCore.framework
- GoogleMaps-2.0.1/Subspecs/Maps/Frameworks/GoogleMaps.framework

6. Right-click ***GooglePlaces.framework*** in your project, and select **Show In Finder**.
7. Drag the ***GooglePlaces.bundle*** from the ***Resources*** folder into your project. When prompted, ensure *Copy items into destination group's folder* is not selected.
8. If you are using the place picker, repeat the previous step and drag the ***GoogleMaps.bundle*** and ***GooglePlacePicker.bundle*** files into your project. These can be found in the ***GoogleMaps.framework*** and ***GooglePlacePicker.framework*** respectively.
9. ***Select your project*** from the Project Navigator, and choose your ***application's target***.
10. Open the ***Build Phases*** **tab**, and within ***Link Binary with Libraries***, add the following frameworks:

- GooglePlaces.framework
- GoogleMapsBase.framework
- GooglePlacePicker.framework (only if using the place picker)
- GoogleMapsCore.framework (only if using the place picker)
- GoogleMaps.framework (only if using the place picker)
- Accelerate.framework
- CoreData.framework
- CoreGraphics.framework
- CoreLocation.framework
- CoreText.framework
- GLKit.framework
- ImageIO.framework
- libc++.tbd
- libz.tbd
- OpenGLES.framework
- QuartzCore.framework
- SystemConfiguration.framework
- UIKit.framework

11. Choose your project, rather than a specific target, and open the ***Build Settings*** tab.
12. In the ***Other Linker Flags*** section, add ***-ObjC***. If these settings are not visible, change the filter in the *Build Settings* bar from *Basic* to *All*.

Step 3: Get an API key (Different from G-Maps key)

13. To get started using the Google Places API for iOS, click the button below which guides you through the process of activating the Google Places API for iOS and getting an API key

Alternatively, follow these steps to get an API key:

1. Go to the Google API Console.
2. Create or select a project.
3. Click **Continue** to enable the Google Places API for iOS.
4. On the **Credentials** page, get an **iOS key** and enter your app's bundle identifier when prompted. For example: ***com.example.helloplaces***.

Note: If you have an existing **iOS key**, you may use that key. You can use the same key with any of your iOS applications within the same project.

5. Click **Create**.

Your new iOS key appears in the list of API keys for your project. An API key is a string of characters, something like this:

AizaSyBdVl-cTICSwYKrZ95SuvNw7dbMuDt1KG0

You can also look up an existing key in the Google API Console.

Step 4: Add the API key to your application

Swift

Add your API key to your ***AppDelegate.swift*** as follows:

•Add the following import statement:

```
import GooglePlaces
```

•Add the following to your ***application(_:didFinishLaunchingWithOptions:)*** method, replacing *YOUR_API_KEY* with your API key:

```
GMSPlacesClient.provideAPIKey("YOUR_API_KEY")
```

Objective C

Add your API key to your ***AppDelegate.m*** as follows:

•Add the following import statement:

```
@import GooglePlaces;
```

•Add the following to your ***application:didFinishLaunchingWithOptions:*** method, replacing *YOUR_API_KEY* with your API key:

```
[GMSPlacesClient provideAPIKey:@"YOUR_API_KEY"];
```

Note : Add your Google Maps API key also, Places work along with Maps. Refer Google maps document for steps

Place Picker

The place picker is a simple and yet flexible built-in UI widget, part of the Google Places API for iOS.

Introducing the place picker

The GMSPlacePicker provides a UI dialog that displays an interactive map and a list of nearby places, including places corresponding to geographical addresses and local businesses. Users can choose a place, and your app can then retrieve the details of the selected place.

The place picker provides the following advantages over developing your own UI widget:

1. The user experience is consistent with other apps using the place picker, including Google apps and third parties. This means users of your app already know how to interact with the place picker.
2. The map is integrated into the place picker.
3. Accessibility is built in.
4. It saves development time.

The place picker features autocomplete functionality, which displays place predictions based on user search input. This functionality is present in all place picker integrations, so you don't need to do anything extra to enable autocomplete. For more information about autocomplete.

Request location authorization

If your app uses the place picker, you must request permission to use location services. First add one or both of the following keys to your ***Info.plist*** file, to request 'when in use' or 'always' authorization:

- ***NSLocationWhenInUseUsageDescription***
- ***NSLocationAlwaysUsageDescription***

For the place picker, it's enough to request 'when in use' authorization, but you may want to request 'always' authorization for other functionality in your app. For each key, add a string informing the user why you need the location services. For example:

```
<key>NSLocationWhenInUseUsageDescription</key>
<string>Show your location on the map</string>
```

Reference :

https://developers.google.com/places/ios-api/start