

Before you start using the plugin, you need make sure that you have a [PUSHY](#) login.

After logging in, you will have to access the applications tab, and **create a new app**, enter the details of the app and click **Create**.

The **PUSHY iOS SDK** is distributed through [Cocoapods](#).

The **Cocoapods** extension is now integrated with the latest cordova-ios version, so make sure you have the latest cordova-ios (version $\geq 4.3.0$).

If you are unsure of the cordova-ios version, navigate to the project folder and type the following command,

Cordova platform

If you have version lesser than 4.3.0, you might need to update the same, to do so type

Cordova platform update ios

Warning : If you update the cordova-ios version, some other plugins might face issues, so please update this in a separate environment rather than in production.

If you have updated the cordova, an error similar to the below might occur. Follow the steps mentioned in the error.

```
Updating ios project...
iOS project updated with cordova-ios@4.3.1
Error: The product name change (<name> tag) in config.xml is not supported dynamically.
To change your product name, you have to remove, then add your ios platform again.
Make sure you save your plugins beforehand using `cordova plugin save`.
    cordova plugin save
    cordova platform rm ios
    cordova platform add ios
```

Adding the plugin :

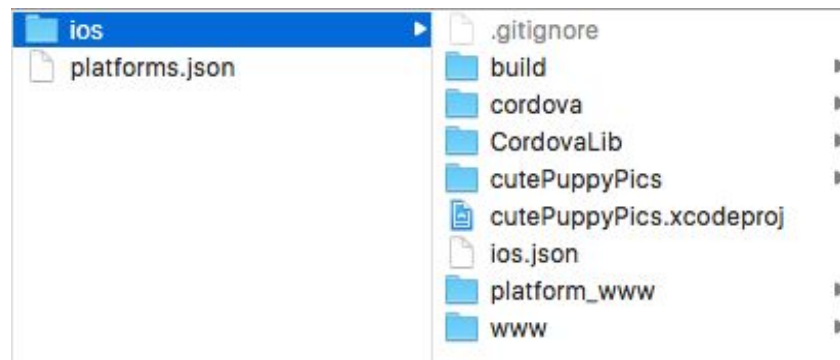
cordova plugin add <https://github.com/SreenivasR/cordova-pushy.git>

Manually Adding the plugin :

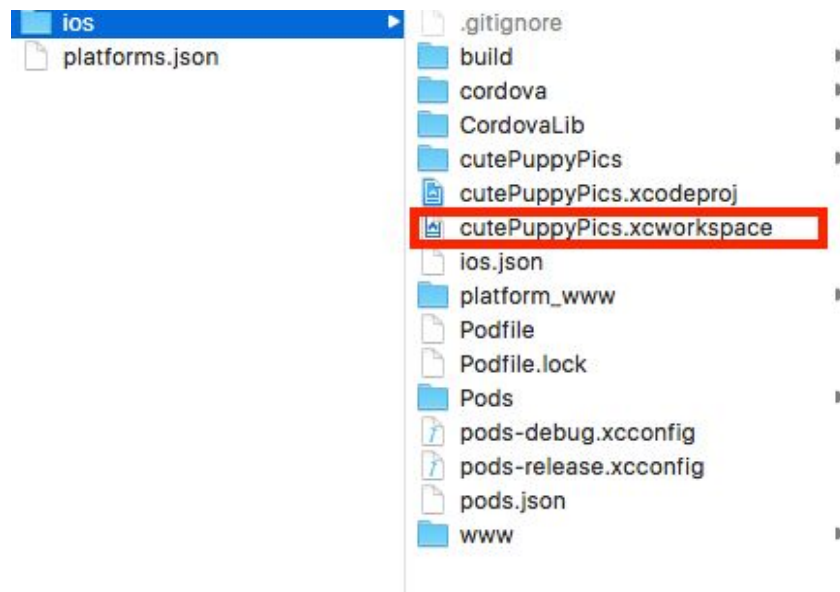
Once you have downloaded the plugin and unzipped it, you can install the plugin manually, using the below command,

cordova plugin add ~/Downloads/pushy.notification

Once you have added the plugin, build the project. Before building the iOS project you may have the folder structure similar to the below image



Once you have done building the project, you will have a new **.xcworkspace** project file. From here on we should be using this file to **run (or) archive** the project.



If build fails, make sure you have provisioning profiles and development team set appropriately.

If the app crashes during launch with the exception as, **"dyld: Library not loaded: @rpath/PushySDK.framework/PushySDK Referenced from:"**, then you might have to remove and add the plugin again.

Once you have verified, that you have required **cordova-ios** version, Open the Xcode project and do the following,

Go to the **project editor** for your **target**,

1. select the **"Build Settings"** tab and make sure you set the below property,

Always Embed Swift Standard Libraries to **Yes**

2. select the **"Capabilities"** tab
 1. turn on **Push Notifications** capability
 2. turn on **Background Modes** capability and check **Remote notifications**

For TypeScript:

Registering a device:

The following method will return you a push token from PUSHY. This token needs to be stored somewhere for sending targeted push notifications to user groups.

Inside of **app.component.ts**, add the following lines inside of **platform.ready()**.

```
if((<any>window).Pushy){  
  
    (<any>window).Pushy.register(  
    function(val) {  
        alert("Pushy device Token: "+val); // pushy device token  
    }, function() {  
        alert("ERROR");  
    });  
}  
else{  
  
    alert("no plugin")  
}
```

Notification Message received callback:

This callback function is invoked in two scenarios:

1. Whenever the app is in foreground and a notification is received
2. Whenever user taps on the notification and the app is in background or in the closed state.

Inside of *app.component.ts*, add the following lines above the **@Component** declaration.

```
interface Window {  
    pushJSON: Function;  
}  
  
(<any>window).pushJSON = (state:String, data:Object) => {  
    alert(data.toString());  
    alert(state);  
};
```

For JavaScript, same as above as you would do for all other plugins except for the type-casts.

Testing push notification:

To test the push notification, go to the URL.

Payload to should have “content_available” flag.

EXAMPLE PAYLOAD :

```
{  
    "message": "Hello World!", "content_available":1  
}
```