Steps to Integrate Neura SDK into your App

Neura Description

Neura is an AI Engine that allows your product to

Engaging at the Perfect Moment

A moment is a change in the “situation” of the user. In other words, it could be defined as when one thing stops and another begins. For example, when a user stops sleeping and wakes up. It is knowing and enabling a product to act at this exact moment that makes Neura so powerful.

Adapting to Each User

There’s a clear difference between knowing a user’s intentions and knowing a user. Purchasing a fitness tracking device shows intent, but it’s only when that device can adapt to a user’s true persona that engage is sustained. In the world’s crowded marketplace, it’s the difference between becoming an integral part of the user’s life or the flavor of the month.

Anticipating Users’ Needs

It’s when knowing a user and recognizing key moments is combined with being able to accurately predict their behavior or intent, that a product becomes most valuable. Neura Prediction Services are based on the Neura AI engine’s ability to recognize user behavior, learn and analyze it – then make informed predictions about the user’s future actions.

For more info Please visit https://www.theneura.com/

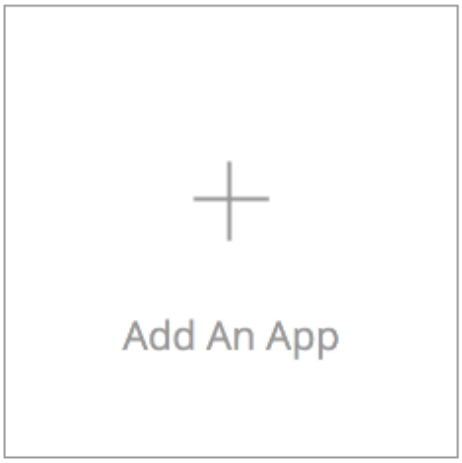
Registration of app on Neura Console

1) Go to Neura Website : https://dev.theneura.com/

2) Create Account : https://dev.theneura.com/signup/ *Accounts management for Philips Yet to be decided

3) Go to My Console : https://dev.theneura.com/console/

4) Click on Add an app icon :



5) You will be redirected to https://dev.theneura.com/app/new

6) Enter the details of the App in which you want to integrate Neura SDK

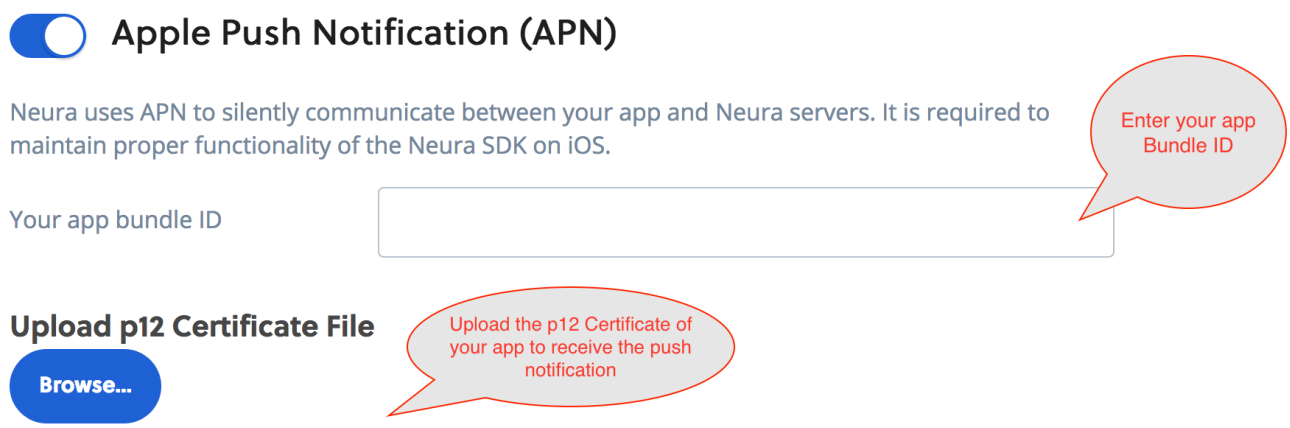
Key Points to remember while filling details ..

- Subscribe to the permission which your app wants to track in tab "Neura Permission"

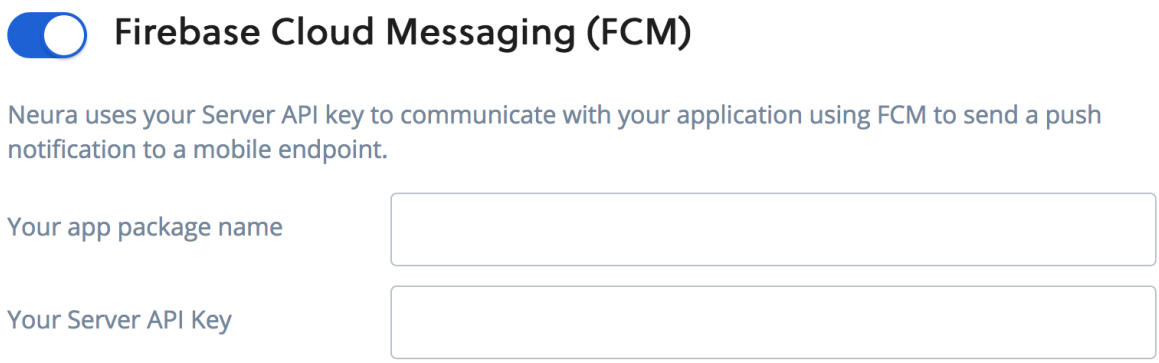
- Webhook is backend you are using. Give the identifier and URL of your backend
- In case you don't want to have webhook .. Please leave it blank (Switch off)

IOS

- Enable the Apple Push Notification .. This is mandatory to receive Push Notification



Android



The server API key is obtained from Firebase Application: https://dev.theneura.com/pages/push-notifications-guide. As this includes google play services in your application integrating Neura need some attention.

Check https://dev.theneura.com/pages/google-play-services-version for play services dependencies.

Neura Integration on the App Side

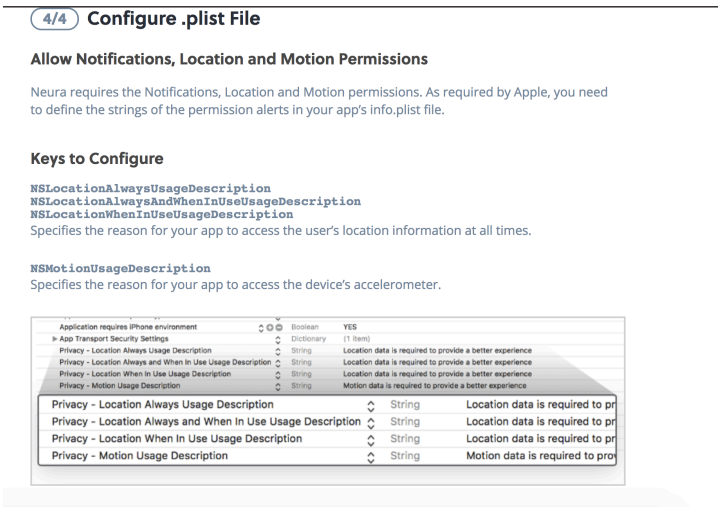
Configuration on App Side

7)

IOS

On the App Side based on the user data you Subscribed to Neura

Configure your info.plist :



The above value in plist is important to get the user consent so that neura can track there location and there fitness activity through Device accelerometer

Android

Add neura dependency in your application's gradle file

```
dependencies {
    compile 'com.theneura:android-sdk:+'
}
```

Precondition :

8)

Initiate the Neura SDK connection

The APP_ID and APP_SECRET Key will be given once you are able to register your app on NeuraConsole (Step 1)

IOS

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject:
AnyObject]?) -> Bool{
  // Override point for customization after application launch.
  NeuraSDK.shared.appUID = "[APP_ID]";
  NeuraSDK.shared.appSecret = "[APP_SECRET]";
  return true
}
```

Android

```
private NeuraApiClient mNeuraApiClient;

mNeuraApiClient = NeuraApiClient.getClient(getApplicationContext(), "[APP_ID]", "[APP_SECRET]")
```

Neura needs location to work. Hence ask for location permission.

```
private void requestLocationPermission(){

    if (ActivityCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_FINE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this,
                new String[]{ android.Manifest.permission.ACCESS_FINE_LOCATION}, 1111);
        return;
    }
    else{
        // Make sure the user enables location,
 // this is needed to for Neura to work, and is not automatic when using anonymous authentication.
 // Phone based auth asks for it automatically.
 Log.d("Neura","Neura Works only for location enabled");
    }
}
```

Code Snippet for iOS and Android to login,Subscribe and Unsubscribe :

9)

Below is the code snippet (swift) to login into Neura

IOS

```
let request = NeuraAnonymousAuthenticationRequest()

        NeuraSDK.shared.authenticate(with: request) {[weak self] result in

            if result.error != nil {

// Not able to login Properly ..

            }else{}
```

Android

```
//Get the FireBase Instance ID, we will use it to instantiate AnonymousAuthenticationRequest
String pushToken = FirebaseInstanceId.getInstance().getToken();

//Instantiate AnonymousAuthenticationRequest instance.
AnonymousAuthenticationRequest request = new AnonymousAuthenticationRequest(pushToken);

//Pass the AnonymousAuthenticationRequest instance and register a call back for success and failure events.
mNeuraApiClient.authenticate(request, new AnonymousAuthenticateCallBack() {
        @Override
        public void onSuccess(AnonymousAuthenticateData authenticateData) {
        Log.i(getClass().getSimpleName(), "Successfully requested authentication with neura. ");
        }

        @Override
        public void onFailure(int errorCode) {
        Log.e(getClass().getSimpleName(), "Failed to authenticate with neura. "
                                                                        + "Reason : " + SDKUtils.
errorCodeToString(errorCode));
        }
});
```

10)

Below is the code snippet (Swift) for subscribing to the moment

IOS

```
var userId = NeuraSDK.shared.neuraUserId()
var identifier = "userStartedRunning" + "_" + userId
subscriptionMoment = NSSubscription(eventName: "userStartedRunning", identifier: identifier, webhookId:"", method: NSSubscriptionMethod.push
NeuraSDK.shared.add(subscriptionMoment, callback: { response in
    if response.error == nil {
      // Do you stuff
    } else {
  // Do you stuff
    }
})
```

This is the moment subscribed while uploading the details of your app in NeuraWebsite

This is the respone we get from call back of neura if it contains error then moment has not been successfully subscribed and vice versa

This is the userID given by Neura Once You successfully able to login into Neura

Android

```
//Define moments you would like to subscribe to.
        List<String> moments = Arrays.asList("userStartedWalking", "userFinishedWalking",
                        "userStartedDriving", "userFinishedDriving", "userWokeUp", "userGotUp", "userIsIdle
For2Hours",
                        "userIsAboutToGoToSleep", "userArrivedHome", "userLeftHome",
                        "userArrivedToWork", "userLeftWork");

//Subscribe to the moments you wish Neura to alert you :
for (int i = 0; i < moments.size(); i++) {
// YourMomentIdentifier_ is recommended to be the NeuraID of the user for follow up with customer suppport
mNeuraApiClient.subscribeToEvent(moments.get(i).getName(),
        "YourMomentIdentifier_" + moments.get(i).getName(),
        new SubscriptionRequestCallbacks() {
                @Override
                public void onSuccess(String eventName, Bundle bundle, String s1) {
                        Log.i(getClass().getSimpleName(), "Successfully subscribed to event " + eventName);
                }

                @Override
                public void onFailure(String eventName, Bundle bundle, int i) {
                        Log.e(getClass().getSimpleName(), "Failed to subscribe to event " + eventName);
                }
        });
}
```

NOTE: You should have subscribed to the moments while app creation as well to get subscriptions for the requestinf moments (Step 6)

11)

Below is code snippet(Swift) for unsubscribing the moment

IOS

NeuraSDK.shared.remove(subscriptionMoment, callback: { (response) in

```
        if response.error == nil {

         // If there is error in the response it means that moment is not successfully unsubcribed

        } else  {



        }

      })
```

Android

```
mNeuraApiClient.removeSubscription(moments.get(i).getName(),
        "YourMomentIdentifier_" + moments.get(i).getName(),
        new SubscriptionRequestCallbacks() {
                @Override
                public void onSuccess(String eventName, Bundle bundle, String s1) {
                        Log.i(getClass().getSimpleName(), "Successfully unsubscribed to event " +
eventName);
                }

                @Override
                public void onFailure(String eventName, Bundle bundle, int i) {
                        Log.e(getClass().getSimpleName(), "Failed to unsubscribe to event " + eventName);
                }
        });
}
```

For More Info how to integrate NeuraSDK into your app .. Please go through the below link :

iOS - https://dev.theneura.com/tutorials/ios

Android - https://dev.theneura.com/tutorials/android

Size of Neura aar - 3.5 MB

How to handle Neura Consent

Reference App Screen Design :

IOS

Once the User gives the permission It can be seen in the privacy settings :

IOS

For More Info on Neura Flow please go through the below link

Neura Flow :
https://confluence.atlas.philips.com/display/UC/ON+%7C+Screen+flow
Neura Consent Specs
https://confluence.atlas.philips.com/display/UC/ON+%7C+Neura+consent+18.2.S2

If You want to integrate without webhookID and allow user consent before he login into UserRegistration .. You can only use **DeviceStoredConsentHandler**

If You want to integrate with webhookID and allow user consent after he login into UserRegistration .. You can use **DeviceStoredConsentHandler, ConsentAccessToolKit(OneBackend) and Cloud**

For More info visit : GDPR

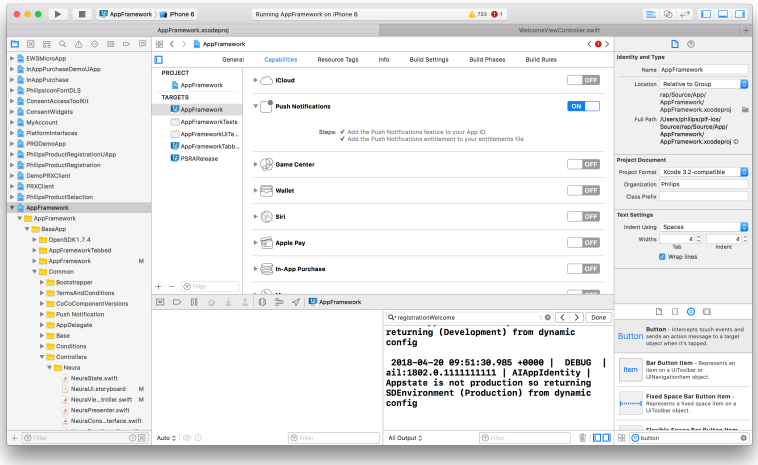
How to handle Push Notification For Neura Consent:

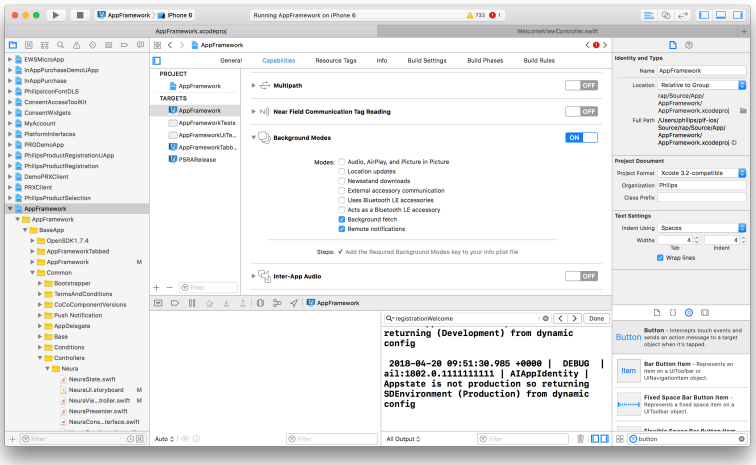
IOS

Below is the code Snippet

```
 func application(_ application: UIApplication, didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data) {

      NeuraSDKPushNotification.registerDeviceToken(deviceToken)

   }

func application(_ application: UIApplication, performFetchWithCompletionHandler completionHandler: @escaping (UIBackgroundFetchResult) -> Void) {

      NeuraSDK.shared.collectDataForBGFetch { result in

         completionHandler(result)

      }

   }

func application(_ application: UIApplication, didReceiveRemoteNotification userInfo: [AnyHashable : Any], fetchCompletionHandler completionHandler: @escaping (UIBackgroundFetchResult) -> ()) {

// Handle the Notification from Neura

}
```

Allow Push Notification in the Project and Background Notification in the Project :





NFR

	Moto G3	MI note 3	Iphone X	iPhone 8	IPAD
Battery Usage 24 hours	11 %	10%	12%	12%	18%
Network Usage	273 KB (1 day)	2.2 MB (1 day)	401 MB (7 days)	100MB(7 days)	85MB(7days)
App size with Neura	Neura aar Size 3.5 MB	same	142.8 MB with Neura (139 MB without Neura)	same	same

Please find the video for Neura :

Your browser does not support the HTML5 video element