

Flutter Tutorial Part 3: Push notification with firebase cloud messaging(FCM)

 medium.com/@jun.chenying/flutter-tutorial-part3-push-notification-with-firebase-cloud-messaging-fcm-2fbdd84d3a5e

April 20, 2019



In this tutorial, I will show you how to integrate Flutter with Firebase Cloud Messaging.

Create a new flutter project

```
flutter create flutter_firebase_push_notification
```

Install the flutter firebase cloud messaging plugin **firebase_messaging 4.0.0+3**

firebase_messaging | Flutter Package

Flutter plugin for Firebase Cloud Messaging, a cross-platform messaging solution that lets you reliably deliver...

pub.dartlang.org

Add this to your package's **pubspec.yaml** file:

dependencies:

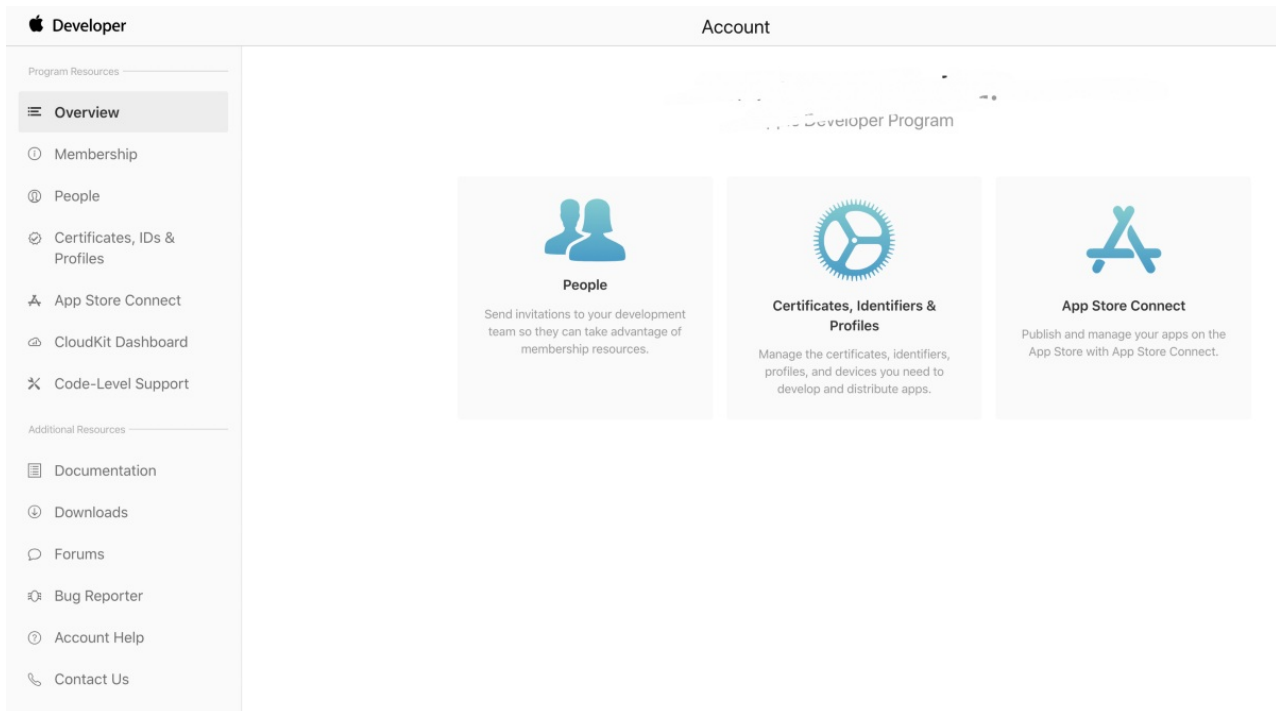
Then get the package

iOS Integration

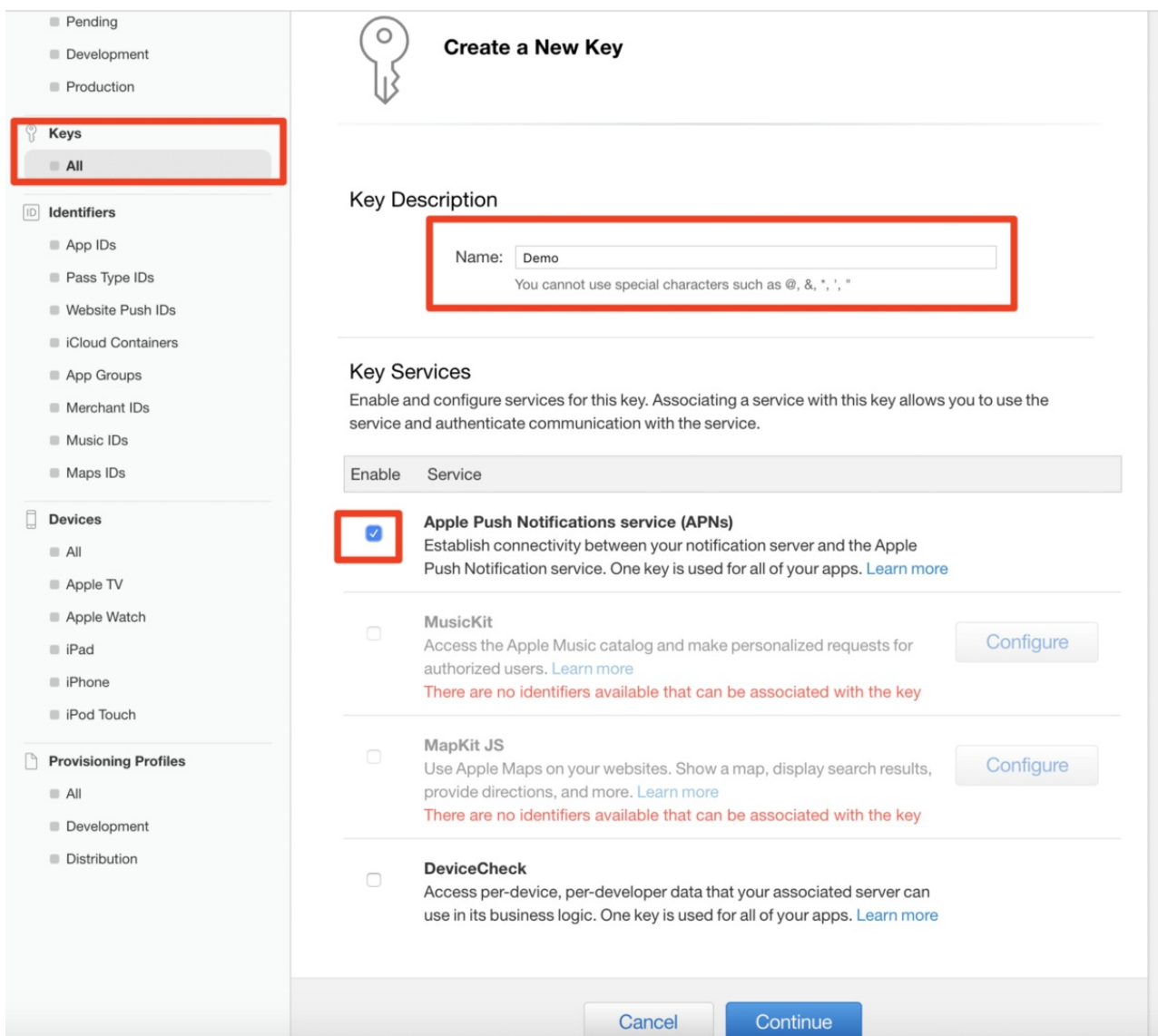
For iOS integration, we need to create a certificate required by Apple.

Configuring APNs with FCM

Login into your Apple developer account, go to Certificates, Identifiers & Profiles to **create the authentication key.**



Click continue and confirm button.



Create a new key

Download your key.

Create and configure services for this key.



Your key is ready.

Download and Back Up

After downloading your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key at this time, click Done and download it at a later time. Be sure to save a backup of your key in a secure place.



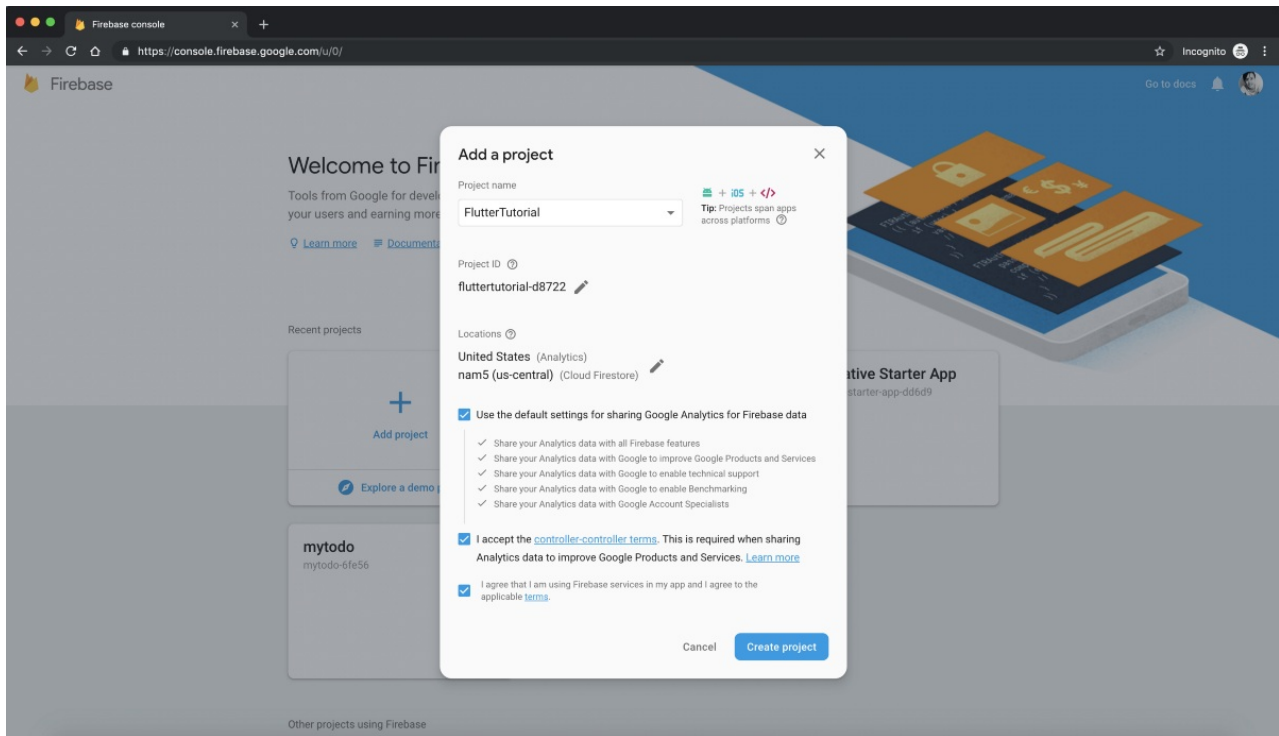
Name:	Demo
Key ID:	[REDACTED]
Services	Apple Push Notifications service (APNs)

Download

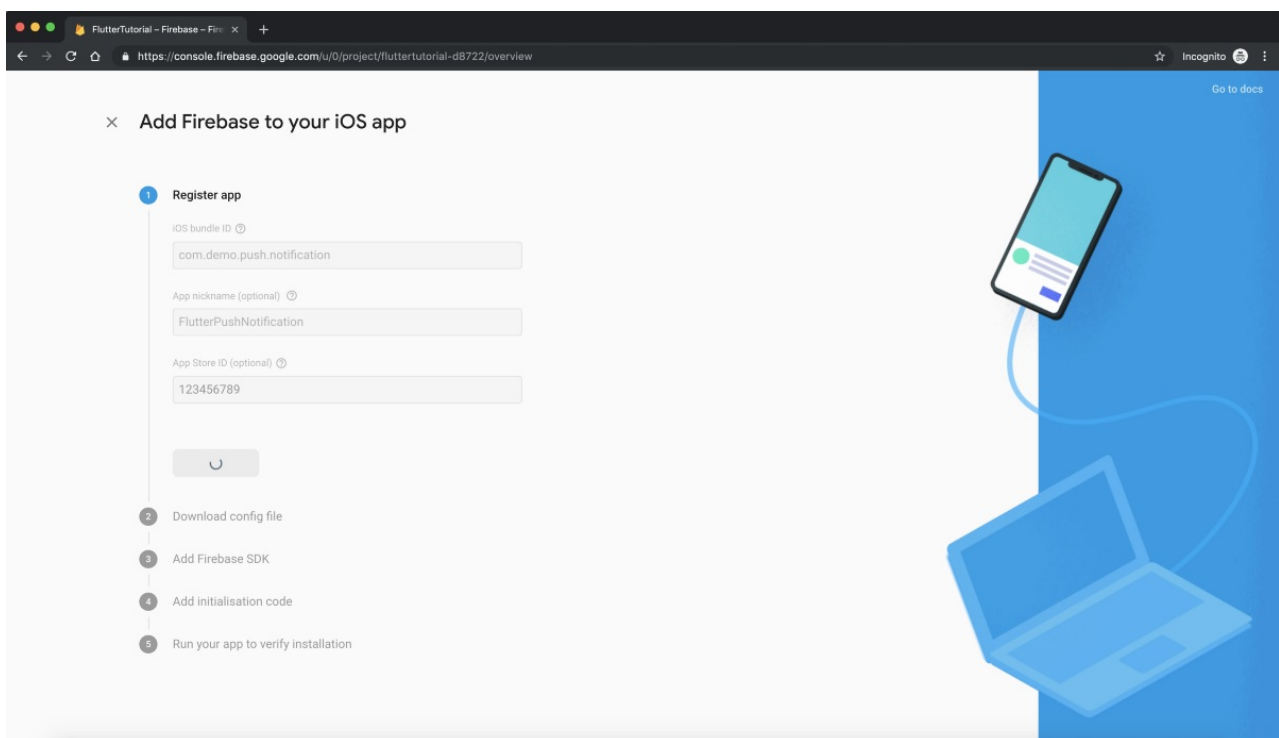
Download your APN key.

Create an App ID

Create a new Firebase app and add an iOS app.

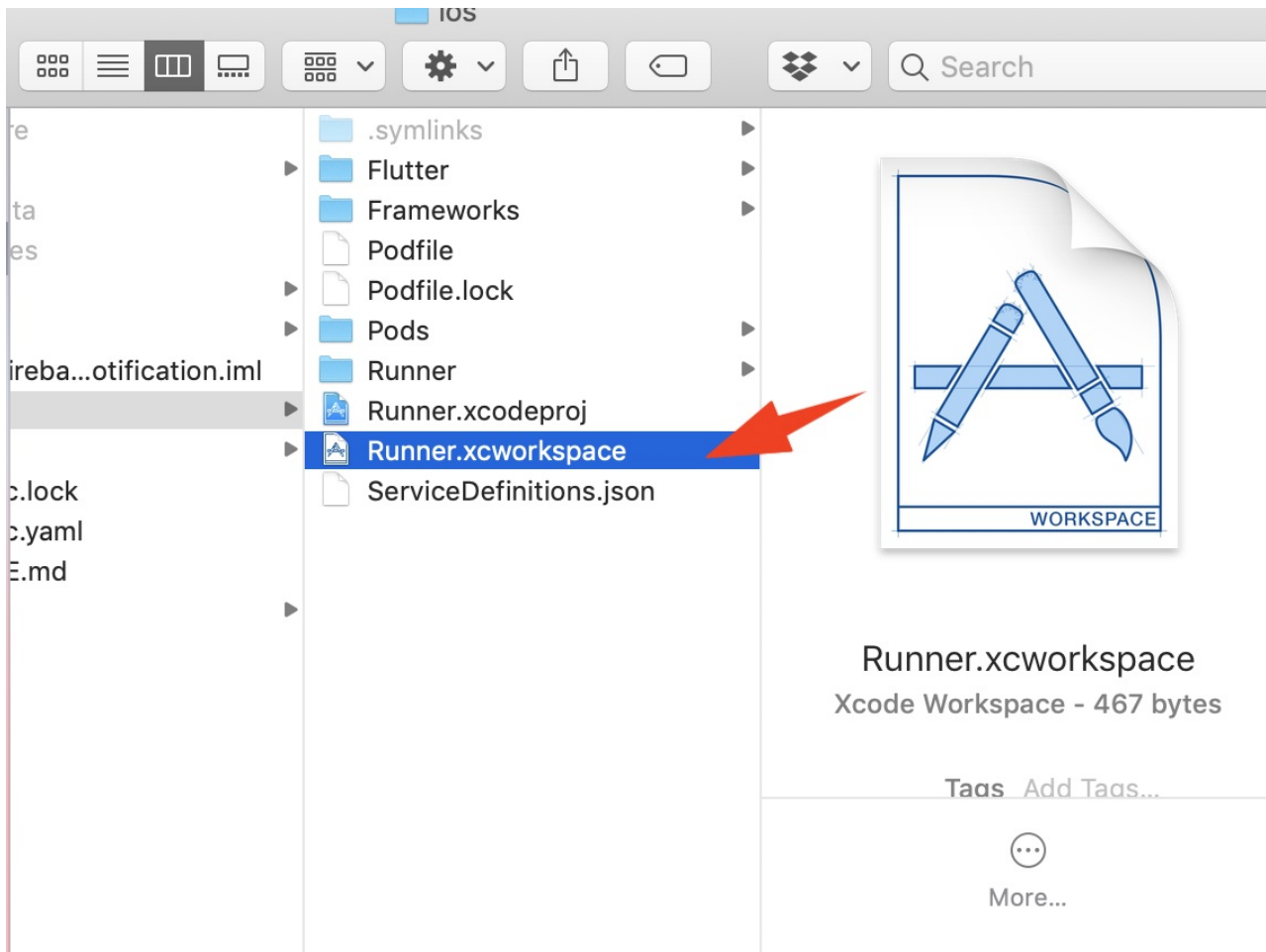


create a firebase project

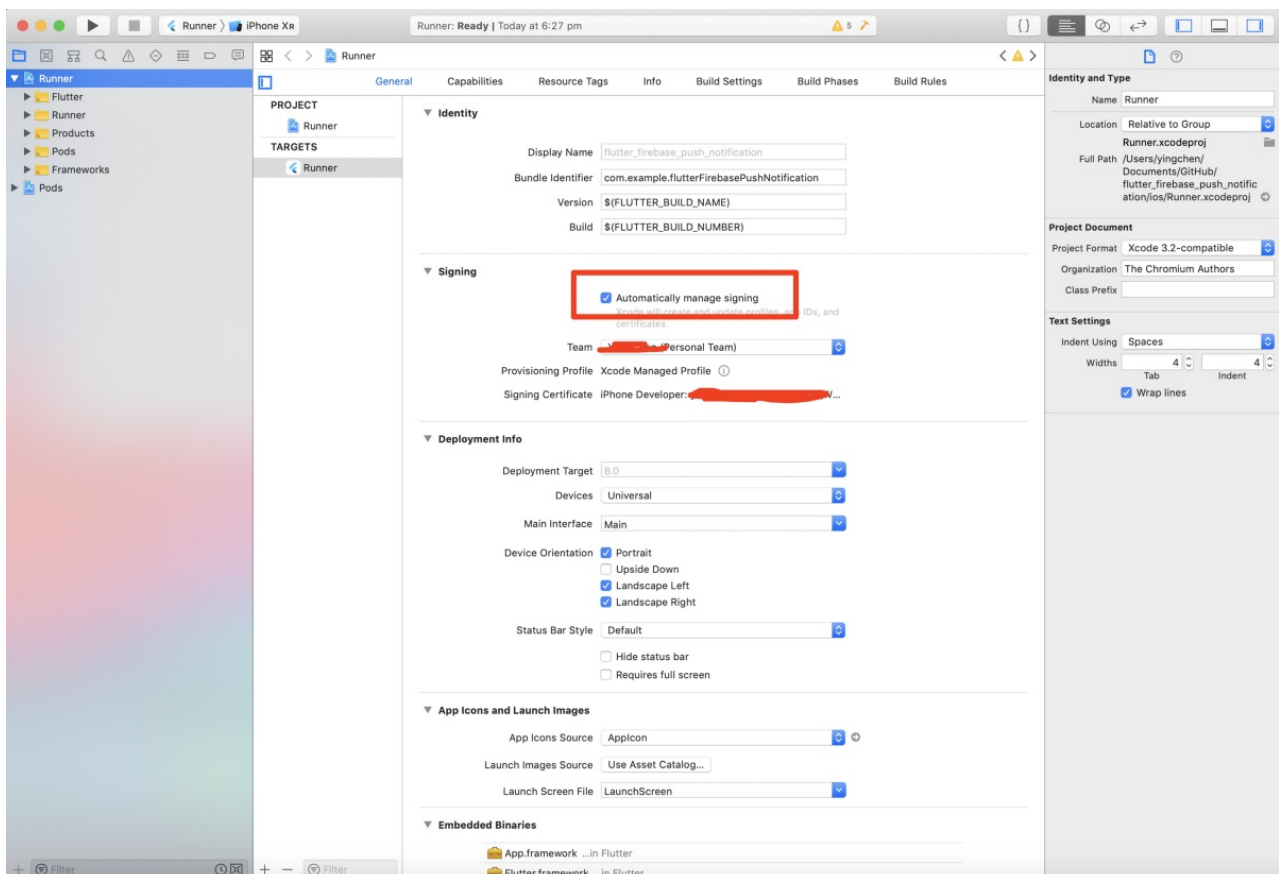


Download google service json file **GoogleService-Info.plist**

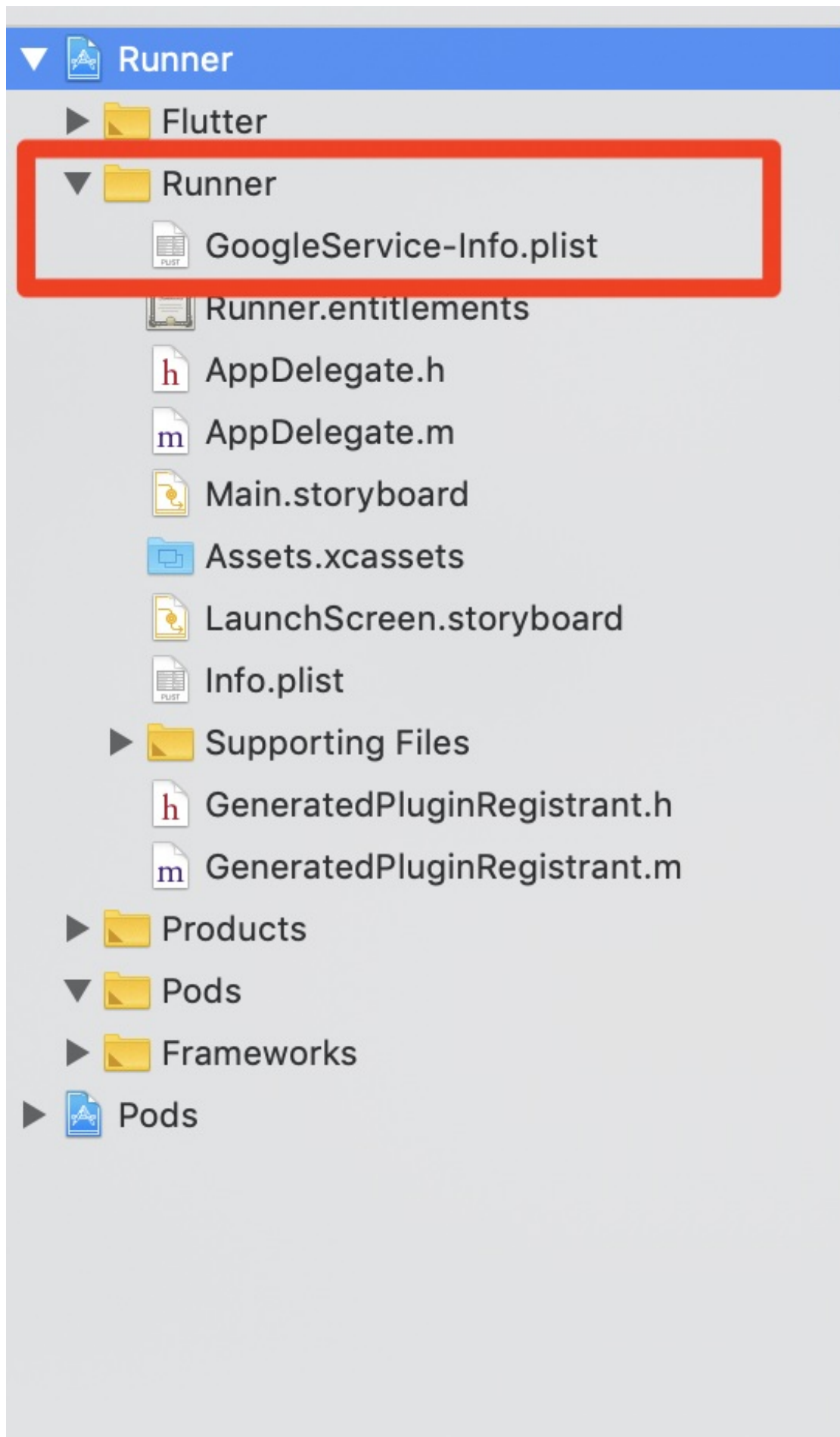
Open **ios/Runner.xcworkspace**



Select automatic signing.



Coy GoogleService-Info.plist into project

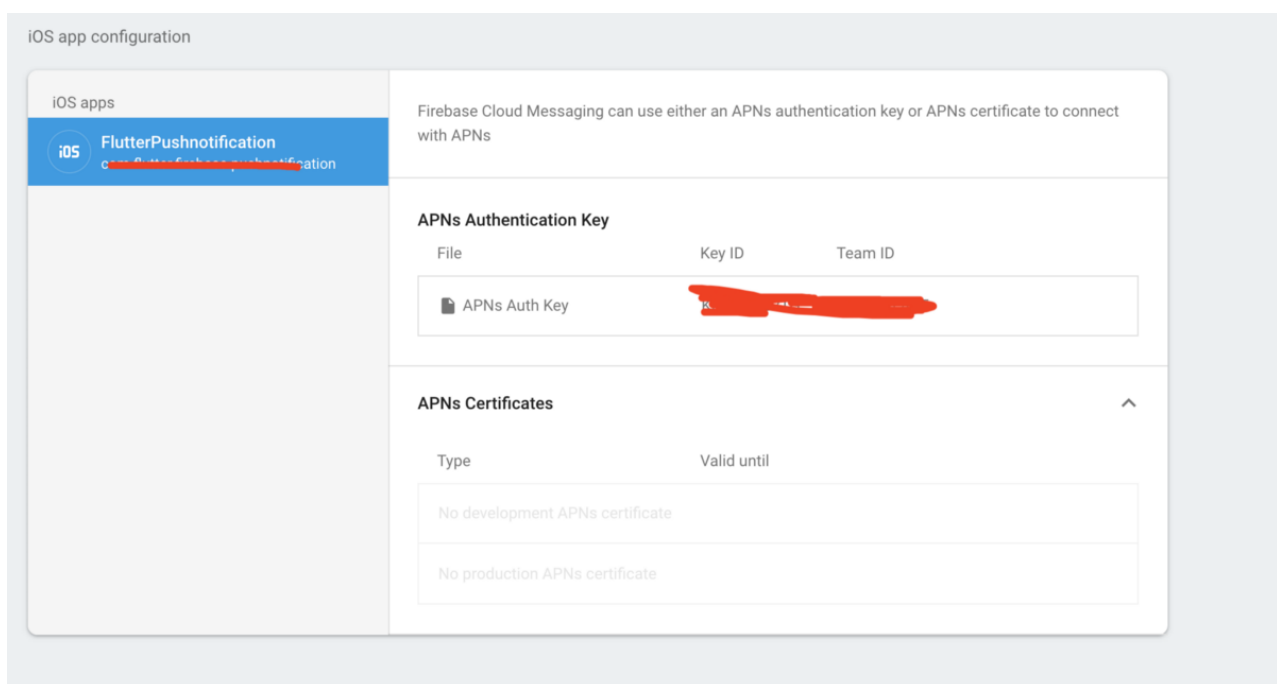


In Xcode, select **Runner** in the Project Navigator. In the Capabilities Tab turn on **Push Notifications** and **Background Modes** , and enable **Background fetch** and **Remote notifications** under **Background Modes** .

Upload your APNs authentication key

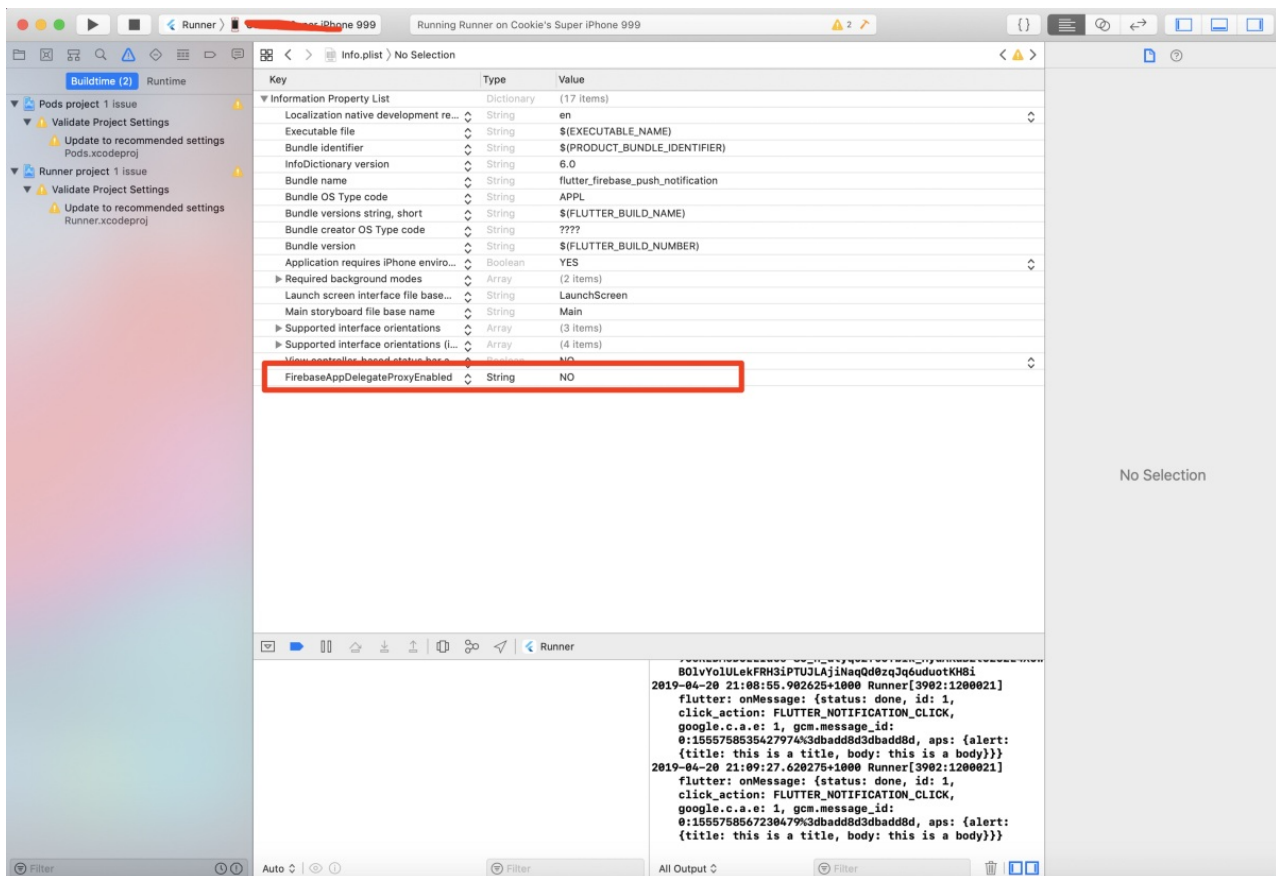
Upload your APNs authentication key to Firebase. If you don't already have an APNs authentication key, see [Configuring APNs with FCM](#).

1. Inside your project in the Firebase console, select the gear icon, select , and then select the tab.
2. In under , click the button.
3. Browse to the location where you saved your key, select it, and click . Add the key ID for the key (available in in the) and click .



Method swizzling in Firebase Cloud Messaging

The FCM SDK performs method swizzling in two key areas: mapping your APNs token to the FCM registration token and capturing analytics data during downstream message callback handling. Developers who prefer not to use swizzling can disable it by adding the flag **FirebaseAppDelegateProxyEnabled** in the app's Info.plist file and setting it to NO (boolean value). Relevant areas of the guides provide code examples, both with and without method swizzling enabled.



Implement lib/main.dart

```

import 'package:firebase_messaging/firebase_messaging.dart';import
'package:flutter/material.dart';class PushMessagingExample extends StatefulWidget {
  _PushMessagingExampleState createState() => _PushMessagingExampleState();}class
_PushMessagingExampleState extends State<PushMessagingExample> { String
_homeScreenText = "Waiting for token..."; String _messageText = "Waiting for
message...";final FirebaseMessaging _firebaseMessaging = FirebaseMessaging(); void
initState() { super.initState(); _firebaseMessaging.configure( onMessage: (Map<String,
dynamic> message) async { setState(() { _messageText = "Push Messaging
message: $message"; }); print("onMessage: $message"); }, onLaunch:
(Map<String, dynamic> message) async { setState(() { _messageText = "Push
Messaging message: $message"; }); print("onLaunch: $message"); },
onResume: (Map<String, dynamic> message) async { setState(() { _messageText =
"Push Messaging message: $message"; }); print("onResume: $message"); }, );
_firebaseMessaging.requestNotificationPermissions( const iosNotificationSettings(sound:
true, badge: true, alert: true)); _firebaseMessaging.onIosSettingsRegistered
.listen((IosNotificationSettings settings) { print("Settings registered: $settings"); });
_firebaseMessaging.getToken().then((String token) { assert(token != null); setState(() {
_homeScreenText = "Push Messaging token: $token"; }); print(_homeScreenText); });
} Widget build(BuildContext context) { return Scaffold( appBar: AppBar( title:
const Text('Push Messaging Demo'), ), body: Material( child: Column(
children: <Widget>[ Center( child: Text(_homeScreenText), ),
Row(children: <Widget>[ Expanded( child: Text(_messageText), )
],), )); }}void main() { runApp( MaterialApp( home:
PushMessagingExample(), ), );}
  
```

Github source code:

Flutter tutorials. Contribute to railsgem/FlutterTutorial development by creating an account on GitHub.

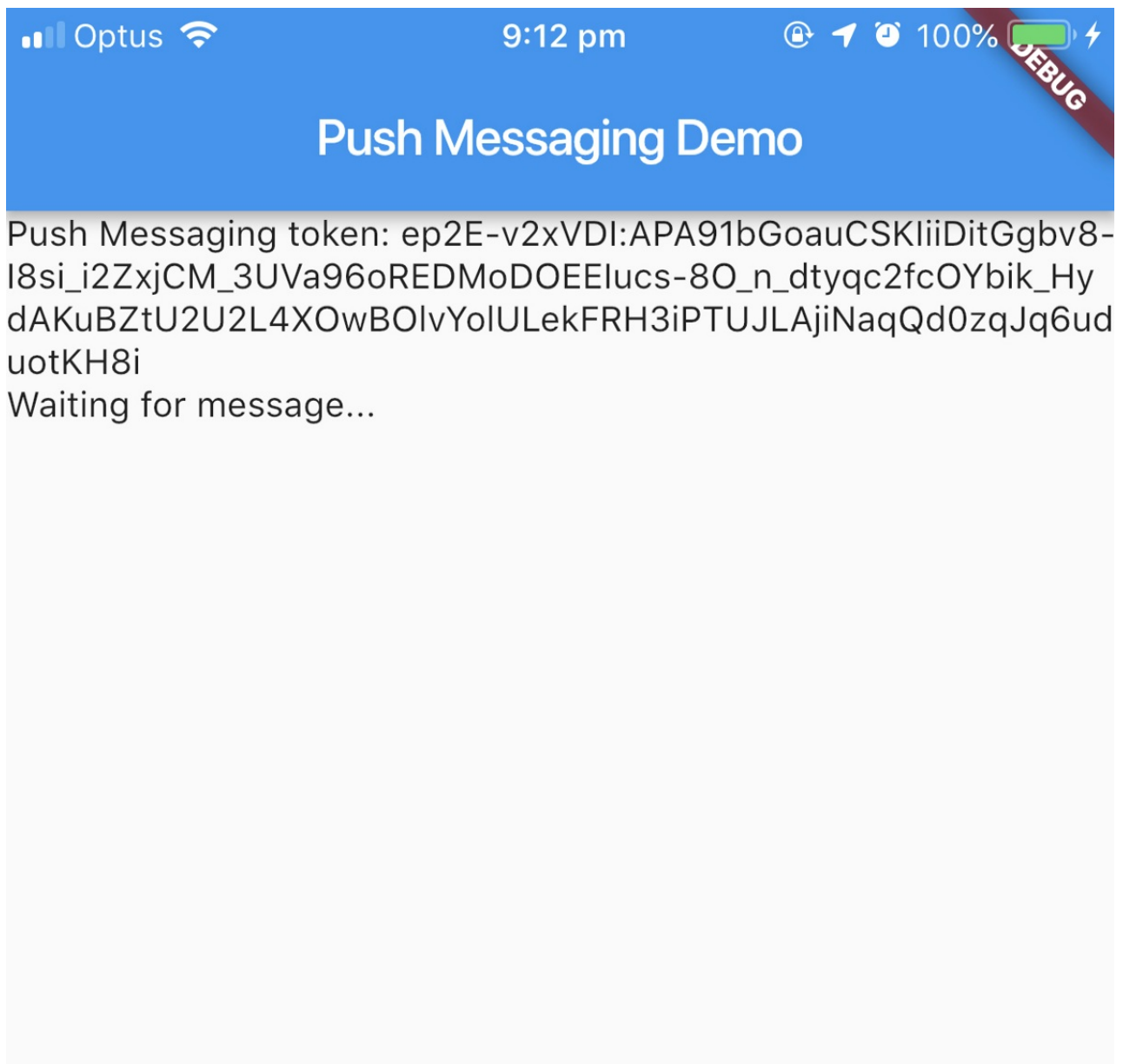
github.com

Deploy to a physical device!

You might not get a notification in a simulator. You might get the following error. So let's test it on a physical device.

```
5.17.0 — [Firebase/Messaging][I-FCM012002] Error in
application:didFailToRegisterForRemoteNotificationsWithError: remote notifications are
not supported in the simulator
```

Run Xcode to deploy the app to the physical device.



Sending Messages

Refer to the [Firebase documentation](#) about FCM for all the details about sending messages to your app. When sending a notification message to an Android device, you need to make sure to set the `click_action` property of the message to `FLUTTER_NOTIFICATION_CLICK`. Otherwise the plugin will be unable to deliver the notification to your app when the users clicks on it in the system tray.

For testing purposes, the simplest way to send a notification is via the [Firebase Console](#). Make sure to include `click_action: FLUTTER_NOTIFICATION_CLICK` as a "Custom data" key-value-pair (under "Advanced options") when targeting an Android device. The Firebase Console does not support sending data messages.

Alternatively, a notification or data message can be sent from a terminal:

```
DATA='{ "notification": { "body": "this is a body", "title": "this is a title" }, "priority": "high",  
"data": { "click_action": "FLUTTER_NOTIFICATION_CLICK", "id": "1", "status": "done" }, "to": "  
<FCM TOKEN>" }'curl -H "Content-Type:application/json" -X POST -d "$DATA" -H  
"Authorization: key=<FCM SERVER KEY>"
```

Remove the `notification` property in `DATA` to send a data message.

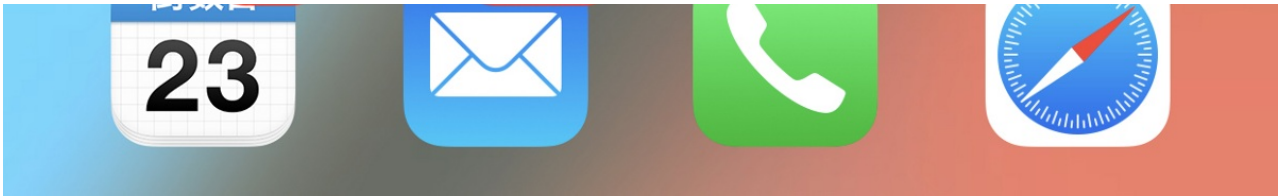


this is a title
this is a body

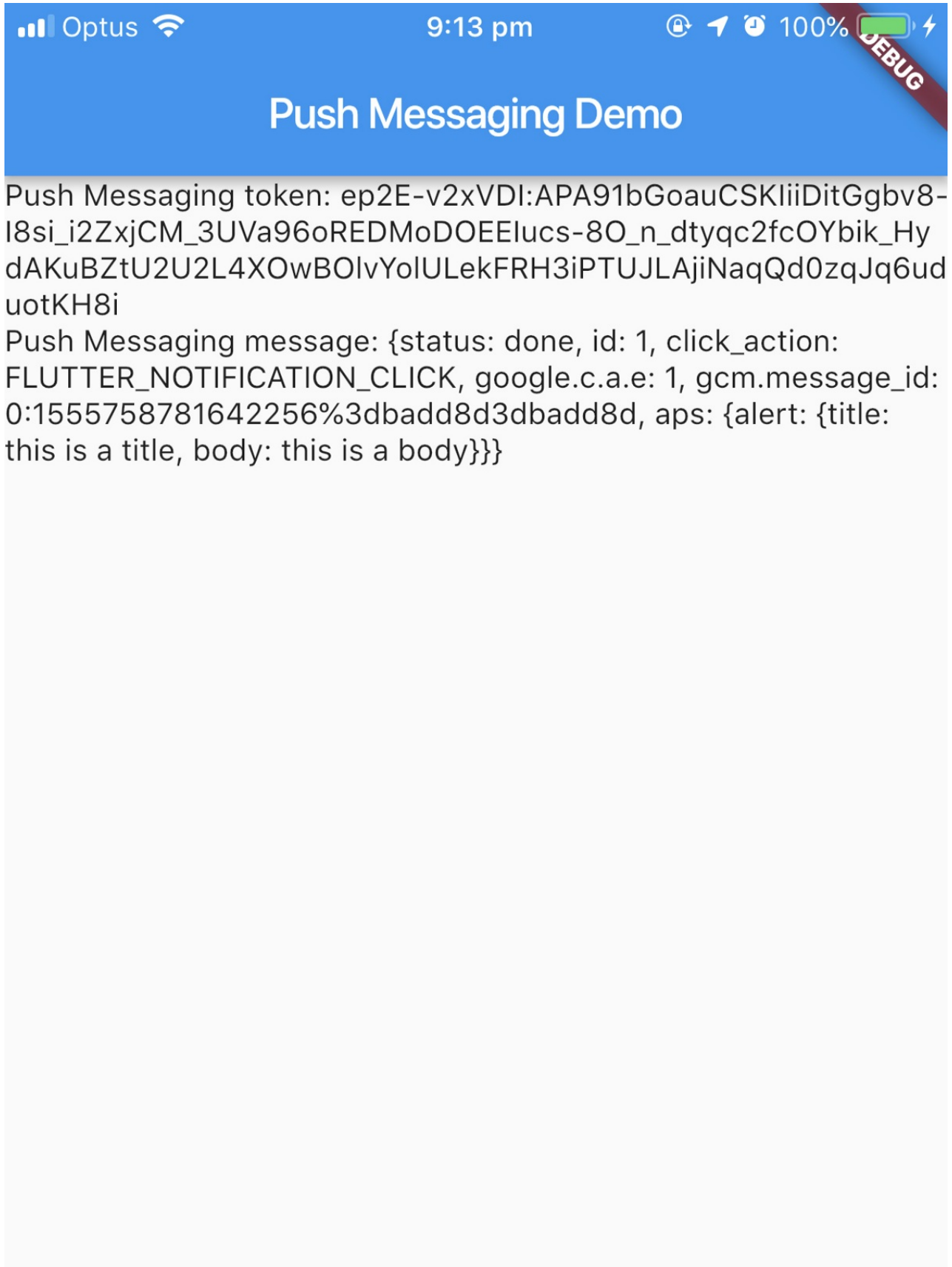
flutter_firebase...

你好，
十月





You will find the message displays on the screen.



Download Source code:

railsgem/FlutterTutorial

Flutter tutorials. Contribute to railsgem/FlutterTutorial development by creating an account on GitHub.

github.com

References:

flutter/plugins

Plugins for Flutter, including FlutterFire, maintained by the Flutter team — flutter/plugins

github.com

Set up a Firebase Cloud Messaging client app on iOS | Firebase

In the same way that iOS typically delivers an APNs device token on app start, FCM provides a registration token via...

firebase.google.com

More Flutter Tutorials

Flutter tutorial part 1: How to build a flutter app

To create a flutter app, check yours have installed the flutter properly.

[medium.com](#)

Flutter Tutorial part 2: Building a listview in Flutter

In this tutorial, I will show you how to build a listview in Flutter.

[medium.com](#)

Thanks for reading!