ADMob -Flutter:

**pubspec.yaml**

dependencies:  
 flutter:  
 sdk: flutter  
  
  
 *# The following adds the Cupertino Icons font to your application.  
 # Use with the CupertinoIcons class for iOS style icons.* cupertino\_icons: ^1.0.2  
 google\_mobile\_ads: ^2.3.0

if you see error like ReadBuffer and WriteBuffer missing:

follow this:

|  |
| --- |
| I also got this problem user\_messaging\_codec.dart:27:19: Error: 'WriteBuffer' isn't a type and Error: 'ReadBuffer' isn't a type. Cause for google mobile ads plugin.  solution goto this location of your drive:- C:\src\flutter.pub-cache\hosted\pub.dartlang.org\google\_mobile\_ads-2.3.0\lib\src\ump\user\_messaging\_codec.dart  and edit googleads user\_messaging\_codec.dart and add :-- "import 'package:flutter/foundation.dart' show ReadBuffer, WriteBuffer;" this line to import the read and write buffer then save the file using ctrl+s and ctrl+s+k.  then do :- 1) flutter clean 2) flutter pub get  it solve my problem, this answer is not found in internet . Follow our youtube channel THECOMPUTERMIND for more. link:-<https://www.youtube.com/@thecomputermind1> |

Now add this to Android Manifest.xml

<meta-data  
 android:name="com.google.android.gms.ads.APPLICATION\_ID"  
 android:value="ca-app-pub-3940256099942544~3347511713"  
/>

**AndroidManifest.xml**

<meta-data  
 android:name="flutterEmbedding"  
 android:value="2" />  
<!-- This is for Ads-->  
 <meta-data  
 android:name="com.google.android.gms.ads.APPLICATION\_ID"  
 android:value="ca-app-pub-3940256099942544~3347511713"  
 />  
 </application>  
</manifest>

Add this to app/build.gradle

implementation 'com.google.android.gms:play-services-ads:21.5.0'

**app/build.gradle:**

flutter **{** source '../..'  
**}**dependencies **{** implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin\_version"  
 implementation 'com.google.android.gms:play-services-ads:21.5.0'  
**}**

Set min SDK version to 21 and target and compile SDK version to 33 in app/build.gradle

def localProperties = new Properties()  
def localPropertiesFile = rootProject.file('local.properties')  
if (localPropertiesFile.exists()) {  
 localPropertiesFile.withReader('UTF-8') **{** reader **->** localProperties.load(reader)  
 **}**}  
  
def flutterRoot = localProperties.getProperty('flutter.sdk')  
if (flutterRoot == null) {  
 throw new GradleException("Flutter SDK not found. Define location with flutter.sdk in the local.properties file.")  
}  
  
def flutterVersionCode = localProperties.getProperty('flutter.versionCode')  
if (flutterVersionCode == null) {  
 flutterVersionCode = '1'  
}  
  
def flutterVersionName = localProperties.getProperty('flutter.versionName')  
if (flutterVersionName == null) {  
 flutterVersionName = '1.0'  
}  
  
apply plugin: 'com.android.application'  
apply plugin: 'kotlin-android'  
apply from: "$flutterRoot/packages/flutter\_tools/gradle/flutter.gradle"  
  
android **{** compileSdkVersion **33** ndkVersion flutter.ndkVersion  
  
 compileOptions **{** sourceCompatibility JavaVersion.VERSION\_1\_8  
 targetCompatibility JavaVersion.VERSION\_1\_8  
 **}** kotlinOptions **{** jvmTarget = '1.8'  
 **}** sourceSets **{** main.java.srcDirs += 'src/main/kotlin'  
 **}** defaultConfig **{** // *TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).* applicationId "com.example.admob\_practice"  
 // You can update the following values to match your application needs.  
 // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-build-configuration.  
 minSdkVersion **21** targetSdkVersion **33** versionCode flutterVersionCode.toInteger()  
 versionName flutterVersionName  
 **}** buildTypes **{** release **{** // *TODO: Add your own signing config for the release build.* // Signing with the debug keys for now, so `flutter run --release` works.  
 signingConfig signingConfigs.debug  
 **}  
 }  
}**flutter **{** source '../..'  
**}**dependencies **{** implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin\_version"  
 implementation 'com.google.android.gms:play-services-ads:21.5.0'  
**}**

In our main.dart file

Inside main function we need to add:

WidgetsFlutterBinding.*ensureInitialized*()**;**await MobileAds.*instance*.initialize()**;**

Make main function as async

For testing purposes add:

MobileAds.*instance*.updateRequestConfiguration(  
 RequestConfiguration(testDeviceIds: ['8C07A346A17C154027F53CEE2D368AFC']))**;**

Complete main.dart file is here:

**main.dart:**

|  |
| --- |
| import 'package:admob\_practice/Wrapper.dart'**;** import 'package:flutter/material.dart'**;** import 'package:google\_mobile\_ads/google\_mobile\_ads.dart'**;** void main() async{  WidgetsFlutterBinding.*ensureInitialized*()**;** await MobileAds.*instance*.initialize()**;** MobileAds.*instance*.updateRequestConfiguration(  RequestConfiguration(testDeviceIds: ['8C07A346A17C154027F53CEE2D368AFC']))**;** runApp(MaterialApp(  debugShowCheckedModeBanner: false**,** home: MyApp()   ))**;** }  var ListE = [  {'name': 'Sun1'}**,** {'name': 'Sun2'}**,** {'name': 'Sun3'}**,** ]**;** class MyApp extends StatelessWidget {  const MyApp({Key? key}) : super(key: key)**;** @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: Text('Appbar')**,** )**,** body: ElevatedButton(  onPressed: (){  Navigator.*push*(context**,** MaterialPageRoute(  builder: (context) => Wrapper(Sunlist: ListE)**,** ))**;** }**,** child: Text('Press')**,** )    )**;** } } |

Now we can use adds, this implementation works for banner Ads:

**Wrapper.dart:**

|  |
| --- |
| import 'package:flutter/material.dart'**;** import 'package:google\_mobile\_ads/google\_mobile\_ads.dart'**;** class Wrapper extends StatefulWidget {  final List<Map<String**,** dynamic>> Sunlist**;** Wrapper({required this.Sunlist})**;** @override  State<Wrapper> createState() => \_WrapperState()**;** }    class \_WrapperState extends State<Wrapper> {   late BannerAd bannerAd**;** bool isAdLoaded = false**;** var adUnit = 'ca-app-pub-3940256099942544/6300978111'**;** initBannerAd() {  print(adUnit)**;** bannerAd= BannerAd(  size: AdSize.*banner***,** adUnitId: adUnit**,** listener: BannerAdListener(  onAdLoaded: (ad){  setState(() {  isAdLoaded = true**;** })**;** }**,** onAdFailedToLoad: (ad**,**error){  ad.dispose()**;** print(error)**;** }**,** )**,** request: AdRequest(  // keywords: <String>['foo', 'bar'],  // contentUrl: 'http://foo.com/bar.html',  // nonPersonalizedAds: true,  )**,** )**;** bannerAd.load()**;** print(adUnit)**;** }    @override  void initState() {   super.initState()**;** initBannerAd()**;** }    @override  Widget build(BuildContext context) {     return Scaffold(  appBar: AppBar(title: Text('AdPage')**,**)**,** body: SingleChildScrollView(  child: Column(   children: [   Container(height: MediaQuery.*of*(context).size.height\***0.6,** color: Colors.*red***,** )**,** ElevatedButton(onPressed: (){}**,** child: Center(child: Text('button')**,**))  ]**,** )**,** )**,** bottomNavigationBar: isAdLoaded  ? SizedBox(  height: bannerAd.size.height.toDouble()**,** width: bannerAd.size.width.toDouble()**,** child: AdWidget(ad: bannerAd**,**)**,** ): const SizedBox()  )**;** } } |