**Push Notification SDK - Complete Developer Guide**

**Table of Contents**

1. Overview
2. Prerequisites
3. Firebase Setup
4. Web Dashboard Registration
5. SDK Integration
6. Configuration
7. Usage Examples
8. API Reference
9. Troubleshooting
10. Support

**Overview**

The Push Notification SDK provides a complete solution for integrating push notifications into your Android application. It includes:

* **Firebase Cloud Messaging (FCM)** integration
* **User preference management** with customizable interests
* **Notification history tracking**
* **Real-time user analytics**
* **Easy-to-use UI components** for notification setup

**Key Features**

* ✅ Simple 4-step integration
* ✅ Customizable notification categories
* ✅ User preference management
* ✅ Analytics and reporting
* ✅ Modern Material Design UI

**Prerequisites**

Before integrating the SDK, ensure you have:

* **Android Studio** 4.0 or higher
* **Minimum SDK version**: 24 (Android 7.0)
* **Target SDK version**: 35 (Android 15)
* **Java 17** or **Kotlin** support
* **Google Play Services** available on target devices

**Firebase Setup**

**Step 1: Create Firebase Project**

1. Go to [Firebase Console](https://console.firebase.google.com/)
2. Click **"Create a project"** or **"Add project"**
3. Enter your project name (e.g., "MyApp Notifications")
4. Choose whether to enable Google Analytics (recommended)
5. Select your Analytics account or create a new one
6. Click **"Create project"**

**Step 2: Add Android App to Firebase**

1. In your Firebase project, click **"Add app"** and select **Android**
2. Enter your app details:
   * **Android package name**: Your app's package name (e.g., com.yourcompany.yourapp)
   * **App nickname**: A friendly name for your app
   * **Debug signing certificate SHA-1**: Optional but recommended for development
3. Click **"Register app"**

**Step 3: Download Configuration File**

1. Download the google-services.json file
2. Place it in your app's app/ directory (same level as build.gradle)
3. **Important**: This file contains your Firebase configuration and should be kept secure

**Step 4: Add Firebase SDK**

Add the following to your **project-level** build.gradle file:

buildscript {

dependencies {

classpath 'com.google.gms:google-services:4.4.0'

}

}

Add to your **app-level** build.gradle file:

plugins {

id 'com.google.gms.google-services'

}

dependencies {

implementation platform('com.google.firebase:firebase-bom:33.13.0')

implementation 'com.google.firebase:firebase-messaging'

}

**Web Dashboard Registration**

**Step 1: Access the Dashboard**

1. Navigate to our web dashboard: **[Your Dashboard URL]**
2. Click **"Sign Up"** or **"Create Account"**
3. Fill in your company/developer information
4. Verify your email address

**Step 2: Create New Application**

1. After logging in, click **"Create New Application"**
2. Fill in the application details:

**Application Information:**

* + **Application Name**: Your app's display name
  + **Platform**: Select "Android"
  + **Interests**: Enter comma-separated categories (e.g., "sports, news, technology")

**Firebase Configuration:**

* + **Client ID**: Click "Generate" to create a unique identifier
  + **Firebase Service Account JSON**: Upload your Firebase service account key

**Step 3: Generate Firebase Service Account Key**

1. Go to [Firebase Console](https://console.firebase.google.com/)
2. Select your project
3. Click the **gear icon** → **"Project settings"**
4. Go to **"Service accounts"** tab
5. Click **"Generate new private key"**
6. Download the JSON file
7. Upload this file to the dashboard

**Step 4: Get Your App ID**

After creating the application, you'll receive:

* **App ID**: A unique identifier (e.g., 6825f0b2f5d70b84cf230fbf)
* **API Keys**: For server-side integration
* **Dashboard Access**: To manage notifications and view analytics

**⚠️ Important**: Save your App ID - you'll need it for SDK initialization.

**SDK Integration**

**Step 1: Add SDK to Your Project**

**Option A: AAR File (Recommended)**

1. Download the pushnotificationsdk-release.aar file
2. Create a libs folder in your app directory if it doesn't exist
3. Copy the AAR file to app/libs/
4. Add to your app/build.gradle:

dependencies {

implementation files('libs/pushnotificationsdk-release.aar')

// Required dependencies

implementation platform('com.google.firebase:firebase-bom:33.13.0')

implementation 'com.google.firebase:firebase-messaging'

implementation 'com.squareup.retrofit2:retrofit:2.9.0'

implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

}

**Step 2: Add Required Permissions**

The SDK automatically includes these permissions, but ensure they're not conflicting:

<!-- In your AndroidManifest.xml -->

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />

**Step 3: Network Security Configuration**

For HTTP connections (if your server doesn't use HTTPS), add to your AndroidManifest.xml:

<application

android:networkSecurityConfig="@xml/network\_security\_config"

android:usesCleartextTraffic="true">

<!-- Your app content -->

</application>

Create app/src/main/res/xml/network\_security\_config.xml:

<?xml version="1.0" encoding="utf-8"?>

<network-security-config>

<domain-config cleartextTrafficPermitted="true">

<domain includeSubdomains="false">your-server-domain.com</domain>

</domain-config>

</network-security-config>

**Configuration**

**Basic Setup**

In your main activity (usually MainActivity.java):

import com.example.pushnotificationsdk.PushNotificationManager;

import com.example.pushnotificationsdk.UserInfo;

import com.example.pushnotificationsdk.SDKConfiguration;

import com.example.pushnotificationsdk.InterestOption;

public class MainActivity extends AppCompatActivity {

private PushNotificationManager notificationManager;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

initializeSDK();

}

private void initializeSDK() {

// 1. Initialize with your App ID from the dashboard

String appId = "YOUR\_APP\_ID\_HERE"; // Replace with your actual App ID

notificationManager = PushNotificationManager.initialize(this, appId);

// 2. Configure notification categories (optional)

configureSDK();

// 3. Set current user information

setCurrentUser();

// 4. Start the SDK

notificationManager.start();

}

private void configureSDK() {

// Configure notification categories and UI settings

SDKConfiguration config = notificationManager.getConfigurationBuilder()

.setSignupTitle("Enable Notifications")

.setSignupSubtitle("Choose what notifications you'd like to receive")

// Add your notification categories

.addInterest(new InterestOption("breaking\_news", "Breaking News", "Important news alerts", true))

.addInterest(new InterestOption("sports", "Sports", "Sports scores and updates"))

.addInterest(new InterestOption("weather", "Weather", "Weather alerts and forecasts"))

.addInterest(new InterestOption("technology", "Technology", "Tech news and updates"))

// Optional: Show user profile fields

.showAgeField(false)

.showGenderField(false)

.build();

notificationManager.configure(config);

}

private void setCurrentUser() {

// Create user info - replace with your actual user data

List<String> emptyInterests = new ArrayList<>(); // Will be selected in setup screen

UserInfo currentUser = new UserInfo(

"user\_" + getCurrentUserId(), // Your user's unique ID

getCurrentUserGender(), // "male", "female", or "other"

getCurrentUserAge(), // User's age

emptyInterests, // Empty initially - user will select

0.0, // Latitude (not used)

0.0 // Longitude (not used)

);

notificationManager.setCurrentUser(currentUser);

}

// Helper methods - implement these based on your app's user management

private String getCurrentUserId() {

// Return your app's current user ID

return "user123"; // Replace with actual implementation

}

private String getCurrentUserGender() {

// Return user's gender from your user profile

return "male"; // Replace with actual implementation

}

private int getCurrentUserAge() {

// Return user's age from your user profile

return 25; // Replace with actual implementation

}

// Launch notification setup screen

private void showNotificationSetup() {

notificationManager.launchNotificationSetupScreen(this);

}

// Launch notification history screen

private void showNotificationHistory() {

notificationManager.launchNotificationHistoryScreen(this);

}

// Launch settings screen

private void showSettings() {

notificationManager.launchSettingsScreen(this);

}

}

**Advanced Configuration**

**Custom Interest Categories**

// Define your own notification categories

List<InterestOption> customInterests = Arrays.asList(

new InterestOption("promotions", "Promotions", "Special offers and discounts", true),

new InterestOption("events", "Events", "Upcoming events and activities"),

new InterestOption("updates", "App Updates", "New features and improvements"),

new InterestOption("social", "Social", "Friend activities and social updates")

);

SDKConfiguration config = notificationManager.getConfigurationBuilder()

.setInterests(customInterests)

.setSignupTitle("Stay Connected")

.setSignupSubtitle("Select your notification preferences")

.build();

**Usage Examples**

**Example 1: E-commerce App**

public class EcommerceMainActivity extends AppCompatActivity {

private PushNotificationManager notificationManager;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

initializeNotifications();

}

private void initializeNotifications() {

// Initialize with your e-commerce app ID

notificationManager = PushNotificationManager.initialize(this, "ecommerce\_app\_id\_123");

// Configure for e-commerce notifications

SDKConfiguration config = notificationManager.getConfigurationBuilder()

.setSignupTitle("Stay Updated")

.setSignupSubtitle("Get notified about deals and order updates")

.addInterest(new InterestOption("order\_updates", "Order Updates", "Shipping and delivery notifications", true))

.addInterest(new InterestOption("deals", "Deals & Offers", "Special discounts and promotions"))

.addInterest(new InterestOption("new\_products", "New Products", "Latest product launches"))

.addInterest(new InterestOption("price\_drops", "Price Alerts", "Price drop notifications for wishlist items"))

.build();

notificationManager.configure(config);

// Set user info from your customer profile

if (isUserLoggedIn()) {

UserInfo customer = new UserInfo(

getCustomerId(),

getCustomerGender(),

getCustomerAge(),

new ArrayList<>(),

0.0, // Latitude not used

0.0 // Longitude not used

);

notificationManager.setCurrentUser(customer);

}

notificationManager.start();

}

// Show notification preferences when user clicks settings

public void onNotificationSettingsClick(View view) {

if (isUserLoggedIn()) {

notificationManager.launchNotificationSetupScreen(this);

} else {

// Redirect to login first

showLoginScreen();

}

}

}

**Example 2: News App**

public class NewsMainActivity extends AppCompatActivity {

private PushNotificationManager notificationManager;

private void initializeNotifications() {

notificationManager = PushNotificationManager.initialize(this, "news\_app\_id\_456");

SDKConfiguration config = notificationManager.getConfigurationBuilder()

.setSignupTitle("Breaking News Alerts")

.setSignupSubtitle("Choose your news categories")

.addInterest(new InterestOption("breaking", "Breaking News", "Urgent news alerts", true))

.addInterest(new InterestOption("politics", "Politics", "Political news and updates"))

.addInterest(new InterestOption("sports", "Sports", "Sports news and scores"))

.addInterest(new InterestOption("technology", "Technology", "Tech news and innovations"))

.addInterest(new InterestOption("business", "Business", "Business and financial news"))

.addInterest(new InterestOption("entertainment", "Entertainment", "Celebrity and entertainment news"))

.build();

notificationManager.configure(config);

// For news apps, you might want to set a generic user profile

UserInfo reader = new UserInfo(

getDeviceId(), // Use device ID if no user account

"unknown", // Gender not required for news

0, // Age not required

new ArrayList<>(),

0.0, // Latitude not used

0.0 // Longitude not used

);

notificationManager.setCurrentUser(reader);

notificationManager.start();

}

}

**API Reference**

**PushNotificationManager**

**Initialization Methods**

// Initialize SDK with App ID

public static PushNotificationManager initialize(Context context, String appId)

// Get current instance (must call initialize first)

public static PushNotificationManager getInstance()

// Start Firebase messaging

public void start()

**User Management**

// Set current user information

public void setCurrentUser(UserInfo userInfo)

// Get current user

public UserInfo getCurrentUser()

// Register user for notifications

public void registerUser()

public void registerUser(UserInfo userInfo)

// Update user information

public void updateUser(UserInfo userInfo)

**Configuration**

// Configure SDK settings

public void configure(SDKConfiguration config)

// Get configuration builder

public SDKConfiguration.Builder getConfigurationBuilder()

**UI Methods**

// Launch notification setup screen

public void launchNotificationSetupScreen(Context context)

// Launch notification history screen

public void launchNotificationHistoryScreen(Context context)

// Launch settings screen

public void launchSettingsScreen(Context context)

**Utility Methods**

// Get Firebase token

public void getToken(OnTokenReceivedListener listener)

// Unregister device

public void unregisterDevice()

**UserInfo Class**

public class UserInfo {

public UserInfo(String userId, String gender, int age, List<String> interests, double latitude, double longitude)

// Getters and setters

public String getUserId()

public String getGender()

public int getAge()

public List<String> getInterests()

public double getLatitude()

public double getLongitude()

}

**SDKConfiguration.Builder**

public class SDKConfiguration.Builder {

// Set notification categories

public Builder setInterests(List<InterestOption> interests)

public Builder addInterest(InterestOption interest)

// UI customization

public Builder setSignupTitle(String title)

public Builder setSignupSubtitle(String subtitle)

// Field visibility

public Builder showAgeField(boolean show)

public Builder showGenderField(boolean show)

// Gender options

public Builder setGenderOptions(String[] options)

// Build configuration

public SDKConfiguration build()

}

**InterestOption Class**

public class InterestOption {

public InterestOption(String id, String name, String description)

public InterestOption(String id, String name, String description, boolean defaultSelected)

// Getters

public String getId()

public String getName()

public String getDescription()

public boolean isDefaultSelected()

}

**Troubleshooting**

**Common Issues**

**1. "SDK not initialized" Error**

**Problem**: Calling SDK methods before initialization **Solution**: Always call PushNotificationManager.initialize() first

// Wrong

PushNotificationManager.getInstance().start(); // Will throw error

// Correct

PushNotificationManager manager = PushNotificationManager.initialize(this, appId);

manager.start();

**2. Firebase Token Not Received**

**Problem**: FCM token generation fails **Solutions**:

* Ensure google-services.json is in the correct location
* Check internet connection
* Verify Firebase project configuration
* Check if Google Play Services is available

**3. Network Security Policy Error**

**Problem**: HTTP requests blocked on Android 9+ **Solution**: Add network security configuration

<!-- In AndroidManifest.xml -->

<application android:networkSecurityConfig="@xml/network\_security\_config">

**4. Notifications Not Appearing**

**Possible causes**:

* POST\_NOTIFICATIONS permission not granted (Android 13+)
* App is in battery optimization
* Notification channel not created properly

**Solutions**:

// Request notification permission (Android 13+)

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.TIRAMISU) {

if (ContextCompat.checkSelfPermission(this, Manifest.permission.POST\_NOTIFICATIONS)

!= PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(this,

new String[]{Manifest.permission.POST\_NOTIFICATIONS}, 1);

}

}

**Debug Logging**

Enable debug logging to troubleshoot issues:

// Add this to see detailed SDK logs

adb logcat -s PushSDK

Common log messages:

* ✅ SDK started with App ID: xxx - Successful initialization
* ❌ Current user not set - Need to call setCurrentUser()
* 🚀 Registering device to server - User registration in progress
* ✅ Device registered successfully - Registration completed

**Performance Tips**

1. **Initialize Once**: Only call initialize() once in your app lifecycle
2. **Background Processing**: SDK handles background tasks automatically
3. **Memory Usage**: SDK uses minimal memory footprint
4. **Battery Optimization**: Location tracking is optimized for battery life

**Support**

**Getting Help**

1. **Documentation**: Check this guide and the API reference
2. **Debug Logs**: Use adb logcat -s PushSDK to see detailed logs
3. **Dashboard**: Check your app dashboard for analytics and errors
4. **Contact Support**: Email us at support@yourcompany.com

**Reporting Issues**

When reporting issues, please include:

* Android version and device model
* SDK version
* App ID
* Relevant log output
* Steps to reproduce the issue

**Best Practices**

1. **Initialize Early**: Initialize the SDK in your main activity's onCreate()
2. **Handle Permissions**: Always check and request required permissions
3. **User Experience**: Don't show notification setup immediately - let users explore your app first
4. **Testing**: Test on different Android versions and devices
5. **Analytics**: Monitor your dashboard for user engagement and issues

**Integration Checklist**

Before going live, ensure you have:

* Firebase project created and configured
* google-services.json file added to your app
* Firebase service account JSON uploaded to dashboard
* App ID obtained from dashboard
* SDK initialized with correct App ID
* User information properly set
* Notification categories configured
* Permissions properly requested
* Network security configuration added (if using HTTP)
* Tested on multiple devices and Android versions
* Verified notifications are received and displayed

**Changelog**

**Version 1.0.0**

* Initial release
* Firebase Cloud Messaging integration
* User preference management
* Location-based notifications
* Material Design UI components
* Analytics and reporting

**© 2024 Your Company Name. All rights reserved.**

For technical support, please contact: **support@yourcompany.com** For business inquiries: **business@yourcompany.com**