

# Розробка мобільних додатків

---

Лекція 13 - Пуш Сповіщення



# Вступ до **Push**-сповіщень

Push-сповіщення – це короткі спливаючі повідомлення від мобільного додатку, які з'являються на пристрої навіть тоді, коли сам додаток не запущений. Вони можуть інформувати користувача про нові повідомлення, оновлення, акції тощо, привертаючи його увагу у реальному часі.

Всі популярні платформи (iOS, Android тощо) підтримують push-сповіщення через власні служби, тому будь-який додаток, що має дозвіл від користувача, може надсилати такі сповіщення на пристрій.

Push-сповіщення відіграють важливу роль у мобільних застосунках, оскільки дозволяють підтримувати взаємодію з користувачем, нагадувати про себе і своєчасно доставляти важливу інформацію, що позитивно впливає на залученість та утримання аудиторії.

## Основні принципи роботи:

**Основні принципи роботи:** Для отримання push-сповіщень користувач повинен надати дозвіл – зазвичай при першому запуску застосунку система запитує, чи дозволяти надсилати сповіщення. Після цього додаток реєструється у службі сповіщень операційної системи (наприклад, **Apple Push Notification service** або **Firebase Cloud Messaging**) і отримує унікальний **push-токен**, який ідентифікує цей додаток на конкретному пристрої. Надалі бекенд-сервер використовує цей токен, щоб адресно надсилати повідомлення через відповідну службу на пристрій користувача


# Firebase Cloud Messaging (FCM)

**Firebase Cloud Messaging (FCM)** – це безкоштовний сервіс Google для надсилання **push-сповіщень та повідомлень у реальному часі** на Android, iOS і веб-застосунки.

- ◆ Сьогодні **FCM є стандартом для push-сповіщень** у мобільних застосунках.
- ◆ Його активно використовують **Google, Facebook, TikTok, Instagram** та інші компанії.
- ◆ FCM легко інтегрується з **React Native, Expo**.

# Ключові етапи розвитку **FCM**





**GCM (Google Cloud Messaging) → FCM (Firebase Cloud Messaging)**

 **17** **2010** – Google запускає **Google Cloud Messaging (GCM)**, який дозволяє надсилати push-сповіщення на Android.

 **17** **2016** – Google оголошує про **заміщення GCM новим сервісом FCM** у складі **Firebase**.

 **17** **2018** – завершення підтримки GCM - повний перехід на FCM .

## ♦ Що покращилося у FCM у порівнянні з GCM?

-  Простий інтеграційний процес із **Firestore SDK**.
-  Можливість надсилати повідомлення на **iOS, Android та Web** через один API.
-  **Аналітичні інструменти** через Firebase Analytics.
-  Підтримка **локалізованих push-сповіщень** та таргетингу на основі аналітики.

# Що таке **APNs**?

**Apple Push Notification Service (APNs)** – це офіційний сервіс **Apple**, який дозволяє надсилати **push-сповіщення** на **iOS, macOS, watchOS та tvOS** пристрої.

- ◆ APNs використовується для **доставки повідомлень у реальному часі**, оновлення контенту у фоновому режимі, нагадувань та інших подій.
- ◆ Працює через **Apple ID токени та сертифікати**.

# Історія **APNs**

17

**2009** – Apple запускає **binary API APNs** разом із iOS 3.0.

17

**2016** – Apple представляє **HTTP/2 API для APNs**, що забезпечує швидшу та ефективнішу доставку сповіщень.

17

**2020** – старий **binary API APNs** офіційно закрито; усі клієнти переходять на **HTTP/2 API**.

17

**2021** – Apple додає підтримку **iCloud Private Relay**(перший сервер знає **IP користувача**, але не бачить, куди він звертається;другий сервер знає **ціль запиту**, але не знає, хто його відправив), що змінює підхід до конфіденційності повідомлень.

# Локальні **vs** віддалені **push**-сповіщення

З точки зору користувача локальні і віддалені сповіщення виглядають схоже, але механізм їх доставки відрізняється.

# Локальні сповіщення

*Локальні сповіщення* генеруються самим додатком на пристрої і не потребують інтернет-з'єднання чи сервера. Вони можуть бути заздалегідь заплановані або викликані певною подією в додатку – наприклад, нагадування у календарі або будильник. Такі сповіщення не потребують зовнішньої інфраструктури і просто вимагають увімкнений пристрій.

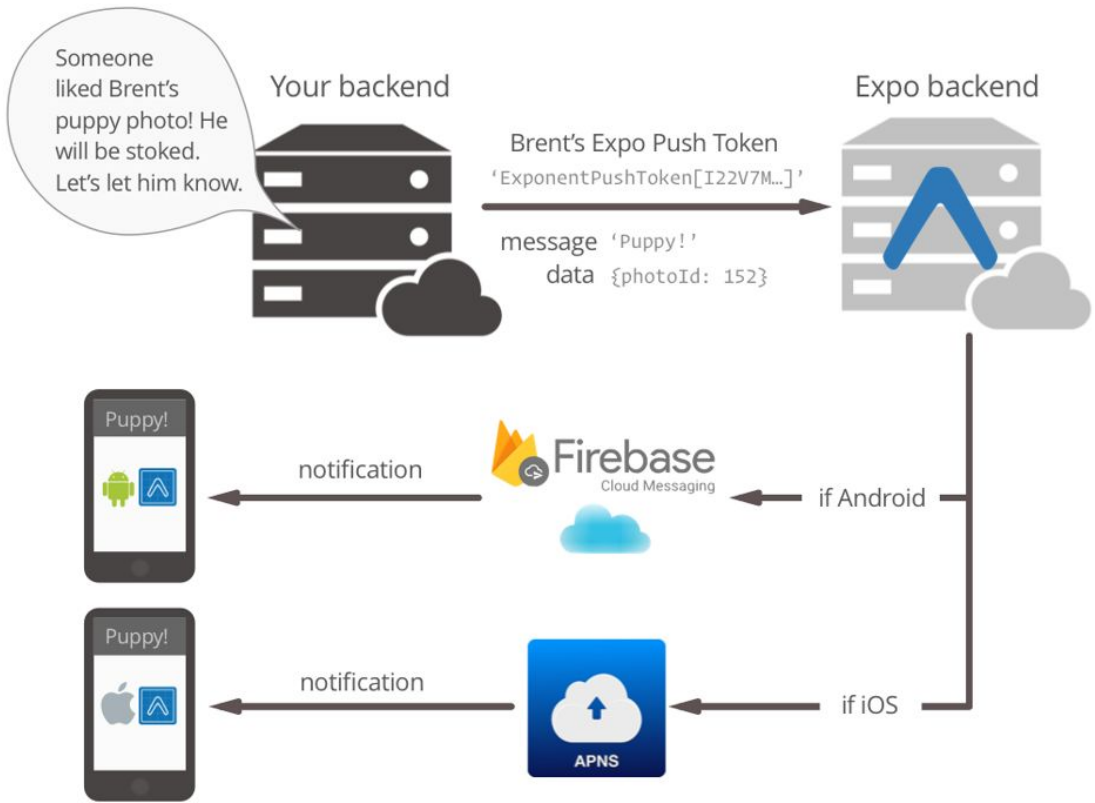
# Віддалені сповіщення

*Віддалені сповіщення (push-сповіщення)* надходять з віддаленого сервера. Додаток на пристрої отримує їх через централізовані служби доставки: для iOS це Apple Push Notification Service (APNs), для Android – Firebase Cloud Messaging (FCM). Віддалені сповіщення дозволяють надсилати більш персоналізований і динамічний контент, і вони приходять незалежно від того, чи використовується додаток в даний момент.

# Як працюють **Push**-сповіщення

**Архітектура push-сповіщень:** Система push-сповіщень включає три основні сторони: мобільний клієнт (додаток), сервер-джерело сповіщень, та служба доставки сповіщень, прив'язана до платформи (APNs для Apple або FCM для Android).

Процес виглядає так: коли користувач встановлює додаток і дає дозвіл на сповіщення, додаток реєструється у службі (APNs/FCM) і отримує унікальний **device token** (для APNs) або **registration token** (для FCM). Цей токен надсилається бекенду і зберігається там. Коли на сервері відбувається подія, що потребує сповістити користувача (наприклад, нове повідомлення), сервер формує повідомлення і відправляє його до служби APNs/FCM разом з токеном пристрою. Служба доставки (Apple або Firebase) визначає пристрій за токеном і пересилає повідомлення на відповідний пристрій користувача. Операційна система отримує це повідомлення через свій постійний зв'язок зі службою (платформи підтримують постійні з'єднання з APNs/FCM для доставки push-даних) і відображає сповіщення користувачу (або обробляє його у фоновому режимі).



**Ролі серверів та клієнта:** Мобільний клієнт відповідає за отримання дозволу і реєстрацію в службі сповіщень, аби отримати токен. Цей токен фактично є “адресою” пристрою в системі push-нотифікацій. Сервер (бекенд) володіє логікою, коли і що надсилати – він генерує контент сповіщення і ініціює його відправку.

Служби APNs (Apple) та FCM (Firebase) виступають шлюзами: вони приймають запити від нашого сервера і розсилають сповіщення на пристрої кінцевих користувачів. Для Apple-пристроїв доставка йде через постійне IP-з'єднання з APNs, яка шифровано спілкується з iOS-пристроєм. Для Android-пристроїв сервер надсилає повідомлення у FCM (хмарний сервіс Google), а той вже пересилає його на пристрій.

Оскільки взаємодія з APNs і FCM має свої протоколи і вимоги (сертифікати, ключі, тощо), часто використовуються готові SDK чи сервіси-посередники щоб спростити цю взаємодію.


**Життєвий цикл push-сповіщення:** Від моменту виникнення події до відображення повідомлення у користувача push-сповіщення проходить кілька етапів. Зазвичай цей процес складається з таких фаз:

1. **Тригер (Trigger):** Деяка подія або умова на сервері генерує запит на сповіщення (наприклад, новий чат від іншого користувача або наближення запланованого заходу).
2. **Створення повідомлення:** Бекенд формує текст і дані сповіщення, які потрібно доставити (заголовок, тіло, додаткові дані тощо).
3. **Відправка (Send):** Сервер відправляє згенерований payload у push-сервіс (APNs або FCM), вказуючи токен цільового пристрою.
4. **Доставка (Delivery):** Служба APNs/FCM отримує повідомлення і пересилає його на зазначений пристрій через мережу.
5. **Отримання і показ (Receive & Display):** Операційна система пристрою отримує сповіщення. Якщо додаток у фоні або закритий – ОС сама показує стандартне сповіщення (банер/алерт) користувачу. Якщо додаток активний – сповіщення надходить у додаток, і той може вирішувати, як його обробити (показати власний UI, тихо обробити, тощо).

Кожен із цих етапів критичний: помилка на будь-якому з кроків (відсутній токен, проблеми з мережею, відсутній дозвіл і т.д.) може призвести до того, що користувач не отримає сповіщення. Розробникам важливо правильно налаштувати усі складові цього ланцюжка.


# Налаштування **ECM**

<https://firebase.google.com/pricing>

Cloud Build minutes	<i>Not applicable</i>	No-cost up to 120min/day Then \$0.003/min
Container storage in <a href="#">Artifact Registry</a>	<i>Not applicable</i>	No-cost up to 500MB of storage Then <a href="#">Google Cloud pricing</a> *Pricing varies based on <a href="#">location</a>
<b>Cloud Messaging (FCM)</b>	No-cost	
<b>Cloud Storage</b>  <b>*.appspot.com legacy buckets</b> GB stored	5 GB	No-cost up to 5 GB Then \$0.026/GB

# Welcome to Firebase!

Tools from Google for building app infrastructure, improving app quality, and growing your business

☰ [View docs](#) 



Get started with a  
Firebase project



Runs in 3 min!

Try a Firebase  
sample app

Explore an app with Firestore, Authentication,  
and the Gemini API in a cloud-based IDE


 [Explore our view only demo project](#)

× Create a project

## Let's start with a name for your project <sup>?</sup>

Project name

kipz-nsi-fcm

 kipz-nsi-fcm

 ztu.edu.ua

I accept the [Firebase terms](#).

I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.

Already have a Google Cloud project?  
[Add Firebase to Google Cloud project](#)

Continue

# Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:

- ✕ A/B testing [?](#)
- ✕ User segmentation & targeting across Firebase products [?](#)
- ✕ Breadcrumbs logs in Crashlytics [?](#)
- ✕ Event-based Cloud Functions triggers [?](#)
- ✕ Free unlimited reporting [?](#)

Enable Google Analytics for this project  
Recommended

[Previous](#)

[Create project](#)



Preparing your project, please wait.

klpz-nsi-fcm

What's new

🏠 Genkit **NEW**

🌟 Vertex AI **NEW**

Product categories

Build ▾

Run ▾

Analytics ▾

AI ▾

🗄️ All products

Related development tools

[IDX](#) ⓘ

[Checks](#) ⓘ

**kipz-nsi-fcm**

Spark plan

📄 Getting started? Tell Gemini about your project

✅ Demo project is ready to view

View ✕

**Get started by adding  
Firebase to your app**



Add an app to get started





## × Add Firebase to your Android app

### ✓ Register app

Android package name: com.ztu.pushnotification, App nickname: PushNotification

### 2 Download and then add config file

Instructions for Android Studio below | [Unity](#) [C++](#)

[Download google-services.json](#)

Switch to the **Project** view in Android Studio to see your project root directory.

Move your downloaded `google-services.json` file into your module (app-level) root directory.

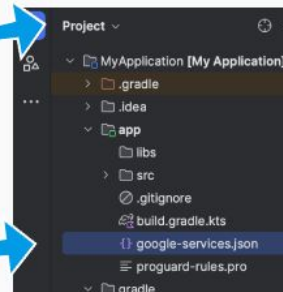


google-services.json

Next

### 3 Add Firebase SDK

### 4 Next steps



## × Add Firebase to your Android app

✓ Register app  
Android package name: com.ztu.pushnotification, App nickname: PushNotification

✎ Download and then add config file

✎ Add Firebase SDK

4 Next steps

You're all set!

Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

Previous

[Continue to console](#)



# Project settings

[General](#)

[Cloud Messaging](#)







[Integrations](#)

[Service accounts](#)

[Data privacy](#)

[Users and permissions](#)

## Your project

Project name	kipz-nsi-fcm 
Project ID 	kipz-nsi-fcm
Project number 	301304567882
Parent org/folder in GCP 	 ztu.edu.ua
Web API Key 	No Web API Key for this project

### Environment

This setting customizes your project for different stages of the app lifecycle

Environment type      Unspecified 

Add app

Android apps



**PushNotification**

com.ztu.pushnotification

### SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

 [See SDK instructions](#)

 [google-services.json](#)

App ID 

1:301304567882:android:eb3349177cfa4d846c6eb3

App nickname

PushNotification 

Package name

com.ztu.pushnotification

SHA certificate fingerprints 

Type 

[Add fingerprint](#)

[Remove this app](#)

kipz-nsi-fcm ▾

## Project settings

General Cloud Messaging Integrations Service accounts Data privacy Users and permissions

**i** Firebase Cloud Messaging has implemented the change for Apple's latest APNs server certificate update. You don't need to take any action. [Learn more](#)

### Firestore Cloud Messaging API (V1) Enabled ⋮

Recommended for most use cases. [Learn more](#) 🔗

Sender ID

Service Account

301304567882

[Manage Service Accounts](#) 🔗

### Cloud Messaging API (Legacy) Disabled ⋮

If you are an existing user of the legacy HTTP or XMPP APIs (deprecated on 6/20/2023), you must migrate to the latest Firebase Cloud Messaging API (HTTP v1) by 6/20/2024. [Learn more](#) 🔗

# Service Accounts

The screenshot shows the 'Service accounts' page in the Google Cloud console. The navigation bar includes 'Integrations', 'Service accounts', 'Data privacy', and 'Users and permissions'. A link for 'Manage service account permissions' is visible in the top right. The main content area is divided into a left sidebar and a main panel. The sidebar has a 'Firebase Admin SDK' header and three sections: 'Legacy credentials', 'Database secrets', and 'All service accounts'. The 'All service accounts' section shows '2 service accounts'. The main panel displays the details for the 'Firebase Admin SDK' service account, including a description, the service account email, and an 'Admin SDK configuration snippet' with radio buttons for 'Node.js', 'Java', 'Python', and 'Go'. A 'Generate new private key' button is located at the bottom of the main panel.

g Integrations **Service accounts** Data privacy Users and permissions

[Manage service account permissions](#)

**Firebase Admin SDK**

Legacy credentials

**Database secrets**

All service accounts

[2 service accounts](#)

**Firebase Admin SDK**

Your Firebase service account can be used to authenticate multiple Firebase features, such as Database, Storage and Auth, programmatically via the unified Admin SDK. [Learn more](#)

Firestore service account  
firebase-adminsdk-fbsvc@kipz-nsi-fcm.iam.gserviceaccount.com

Admin SDK configuration snippet

Node.js  Java  Python  Go

```
var admin = require("firebase-admin");  
  
var serviceAccount = require("path/to/serviceAccountKey.json");  
  
admin.initializeApp({  
  credential: admin.credential.cert(serviceAccount)  
});
```

[Copy](#)

[Generate new private key](#)

## Firestore Admin SDK

### Firestore Admin SDK

Your Firestore service account can be used to authenticate multiple Firestore features, such as Database, Storage and Auth, programmatically via the unified Admin SDK. [Learn more](#)

Legacy credentials



All services



### Generate new private key



**⚠** Your private key gives access to your project's Firestore services. Keep it confidential and never store it in a public repository.

Store this file securely, because your new key can't be recovered if lost.

Cancel

Generate key

```
admin.initializeApp({  
  credential: admin.credential.cert(serviceAccount)  
});
```



Generate new private key

# OneSignal

OneSignal – це сервіс для масового надсилання пуш-сповіщень, email, SMS та in-app повідомлень.

- ◆ Підтримує iOS, Android, Web та інші платформи
- ◆ Інтегрується з Firebase (FCM) та Apple APNs
- ◆ Безкоштовний план з підтримкою до 10,000 підписників
- ◆ Автоматизація повідомлень та таргетинг за поведінкою користувачів

# Як працює **OneSignal**?

- 1 Користувач встановлює застосунок і надає дозвіл на сповіщення.
- 2 OneSignal отримує push-токен від FCM (Android) або APNs (iOS).
- 3 Сервер OneSignal відправляє push-сповіщення через API.
- 4 Користувач отримує сповіщення в реальному часі.

# Основні функції **OneSignal**

## ◆ 1. Push-сповіщення

- ✓ **Масове розсилання повідомлень** на Android, iOS, Web.
- ✓ **Автоматичні повідомлення** (наприклад, після першого входу).
- ✓ **Сегментація** – відправлення тільки певним групам користувачів.

## ◆ 2. In-App Messages

- ✓ Сповіщення **всередині додатка** без пуш-токенів.
- ✓ Відмінно підходить для **реклами та промо-акцій**.

## ◆ 3. Таргетинг аудиторії

- ✓ Фільтрація користувачів за **локацією, пристроєм, поведінкою**.
- ✓ Відправка повідомлень **тільки активним або неактивним користувачам**.

## ◆ 4. Аналітика та A/B тестування

- ✓ Статистика відкриттів, кліків, конверсій.
- ✓ Тестування різних версій повідомлень.

## Self-service plans

Free  
**\$0/mo**

Free access to all channels  
Message limits may apply

[Get started now](#)

Free access to:

- ✓ 10,000/mo Free Email Sends
- ✓ Unlimited Mobile Push Sends
- ✓ Journeys Workflows
- ✓ GDPR Compliant
- ✓ A/B Testing

Growth

Starts at **\$19/mo**

Plus usage costs by channel  
[Estimate your monthly cost](#)

[Get started now](#)

All features from **Free** +

- ✓ 20,000/mo Free Email Sends
- ✓ Advanced In-App Messaging
- ✓ Journeys Branching
- ✓ Intelligent Delivery
- ✓ Expanded Integrations

## Annual plans

Professional  
**Custom**

Pricing by channel  
Volume-based discounting available

[Contact sales](#)

All features from **Growth** +

- ✓ 24/7 Prioritized Support
- ✓ CSV Exports
- ✓ Advanced Journeys
- ✓ Advanced Personalization
- ✓ Custom Outcome Tracking

Enterprise  
**Custom**

Tailored contract & features  
Volume-based discounting available

[Contact sales](#)

All features from **Professional** +

- ✓ Frequency Capping
- ✓ Full Live Activities Access
- ✓ Security/Legal Package + SLA
- ✓ Dedicated Success Manager
- ✓ Advanced User Permissions

# Що таке **GDPR**?

**GDPR (General Data Protection Regulation)** – це Загальний регламент захисту даних ЄС, який регулює збір, зберігання та використання персональних даних користувачів у країнах Європейського Союзу.



**Набрав чинності:** 25 травня 2018 року.



**Основна вимога:** Компанії повинні отримувати **явну згоду користувача** перед збором та обробкою персональних даних.

OneSignal збирає персональні дані, такі як:

- ✓ **Push-токени (FCM, APNs)**
- ✓ **IP-адреса користувача**
- ✓ **Ідентифікатор пристрою (Device ID)**
- ✓ **Геолокація (за згодою користувача)**

Щоб бути **GDPR Compliant**, OneSignal дозволяє:

- ✓ **Відключати автоматичний збір даних**
- ✓ **Запитувати згоду користувача перед збором даних**
- ✓ **Видаляти всі дані користувача за запитом**

# Welcome to OneSignal!

## Tell us about you

Please answer a few short questions so we can customize your OneSignal experience.

First Name \*

Last Name \*

Which category best describes your role? \*

What do you primarily want to use OneSignal for? (Select all that apply) \*

- Drive visits to website/app (e.g. onboarding, updates and reminders)
- Send marketing/promotional messages to drive revenue
- Provide company updates (e.g. announcements, product updates)
- Re-engage inactive users / reduce churn
- Send transactional notifications (e.g. order confirmations)
- Other

What channels do you plan to use OneSignal for? (Select all that apply) \*

- Email
- Push Notifications
- SMS
- In-App Messages
- Journeys

0 of 3 Steps

- 1 About You**
- 2 Create Organization
- 3 Create App



## Welcome to OneSignal!

### Create your first Organization

An organization is a group of 1 or more apps, e.g. a sandbox app and a production app.

Company or Organization Name \*

Website \*

I don't have a website

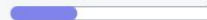
What industry best describes your company? \*

Company Size \*

Receive a live demo of OneSignal

Continue

1 of 3 Steps



- ✓ About You
- ② **Create Organization**
- ③ Create App



## Welcome to OneSignal!

### Now add your first app or website

Need help? [Read our Quickstart Guide.](#)


Name of your app/website \*

Create

2 of 3 Steps






- ✓ About You
- ✓ Create Organization
- ③ Create App


Upgrade 

 Continue your quick start guide.

Dashboard

## Let's get started

-  Onboard, engage, and reactivate your users
-  Seamlessly automate across channels
-  Drive higher conversions, revenue, and LTV


 Get Set Up

Quickstart documentation

## Settings

### Channels


#### Push & In-App

 Set up Push & In-App

#### Email

 Set up Email

#### SMS

 Set up SMS

### General

#### Team Members

Manage or invite team members

#### Keys & IDs

Set up security and API keys

#### Usage

View subscription & billing cycle usage

#### Manage App


Manage this app

## Push Settings


### Get started with OneSignal push

Easily manage mobile and web push in one place. Unlimited push for real-time updates, alerts, and more


[Push setup documentation](#)



Web



Google Android (FCM)




Apple iOS (APNs)



Huawei Android (HMS)




Amazon Fire



macOS



Windows (UWP)



Chrome Extensions

**Need help?** Invite a team member to complete this step.

+ Invite

Continue Go Back

## Google Android (FCM) Configuration

Now, let's configure your app. [Read our Android documentation](#) to learn how to complete the fields below.

Service Account JSON \* ?

Upload from your computer

Choose File

Retrieve from your Firebase Console

**Need help?** Invite a team member to complete this step.

+ Invite

Save & Continue

Go Back

0 of 3 Steps

1 Configure App Settings

2 Select SDK

3 Install and Test

## Google Android (FCM) Configuration

0 of 3 Steps

Now, let's configure your app. [Read our Android documentation](#) to learn how to complete the fields below.

Service Account JSON \* ?

Upload from your computer

This JSON file contains a private key to access FCM v1. Generate the private key in your Firebase project settings under "Service accounts".

[Firebase credentials documentation](#)

🔥 Retrieve from your Firebase Console

**Need help?** Invite a team member to complete this step.

+ Invite

Save & Continue

Go Back

Configure App Settings

Select SDK

Install and Test



PushNotificationExa... ▾

Dashboard

Messages ▾

Journeys

Audience ▾

Delivery ▾

Data ▾

Settings

Upgrade

Documentation

Support

## Google Android (FCM) Configuration

Now, let's configure your app. [Read our Android documentation](#) to learn how to complete the fields below.

Service Account JSON \*

 Choose File [Retrieve from your Firebase Console](#)**Need help?** Invite a team member to complete this step.

Invite

Save &amp; Continue












Go Back

0 of 3 Steps



  
① **Configure App Settings**

② Select SDK

③ Install and Test


-  Dashboard
-  Messages 
-  Journeys
-  Audience 
-  Delivery 
-  Data 
-  **Settings**

Upgrade 

-  Documentation
-  Support

## Google Android (FCM) Configuration

Select your target SDK. We'll take you through the steps to get your first user & send your first test notification.



Native Android



Cordova

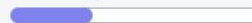


React Native / Expo



Unity

1 of 3 Steps



- Configure App Settings**
- 2 Select SDK**
- 3 Install and Test**

## Google Android (FCM) Configuration

Select your target SDK. We'll take you through the steps to get your first user & send your first test notification.



Native Android



Cordova



React Native / Expo



Unity


1 of 3 Steps

- Configure App Settings
- ② Select SDK**
- ③ Install and Test

# Google Android (FCM) Configuration

## 1. Install the SDK

Read our [React Native / Expo documentation](#) to learn how to install the SDK.

Your App ID: `0d825712-c858-4d19-93a6-ed86867a17a3` 

## 2. Test for Subscribed Users












Build and run your app. The OneSignal SDK, once set up correctly, will automatically handle subscribing your device to notifications.

 [Check Subscribed Users](#)



**Need help?** Invite a team member to complete this step.

Done

[Go Back](#)

-  Dashboard
-  Messages 
-  Journeys
-  Audience 
-  Delivery 
-  Data 
-  **Settings**

Upgrade 

-  Documentation
-  Support

## Settings

### Channels

#### Push & In-App


Active

Platforms set up and message settings

#### Email

 Set up Email

#### SMS

 Set up SMS

### General

#### Team Members

Manage or invite team members

#### Keys & IDs

Set up security and API keys

#### Usage

View subscription & billing cycle usage

#### Manage App

Manage this app

# Application ID

**OneSignal App ID** – це унікальний ідентифікатор вашої програми в OneSignal. Знайти його можна в адмін-панелі OneSignal:

1. **Зареєструйтеся або увійдіть у свій обліковий запис [OneSignal](#).**
2. Якщо у вас ще немає додатку, натисніть **Add New App** (або **New App / Website**) і пройдіть базове налаштування.
3. **На головній сторінці** або у **Settings** вашого додатку ви знайдете **OneSignal App ID**.
  - Зазвичай це рядок на кшталт: `12345abc-6789-de01-f234-56789abcd123`.
4. **Скопіюйте** цей ідентифікатор і використайте його в коді (`YOUR-ONESIGNAL-APP-ID`).

Якщо ж у вас кілька додатків у OneSignal, оберіть потрібний із переліку **All Apps** у верхньому меню. Для кожної програми (проєкту) буде свій унікальний **App ID**.

## Settings

### Channels

#### Push & In-App Active

Platforms set up and message settings



#### Email

[Set up Email](#)



#### SMS

[Set up SMS](#)



### General

#### Team Members

Manage or invite team members



#### Keys & IDs

Set up security and API keys



#### Usage

View subscription & billing cycle usage



#### Manage App

Manage this app



## Keys & IDs

### App ID

OneSignal App ID

0d825712-c858-4d19-93a6-ed86867a17a3 [🔗](#)

### API Keys

[+ Add Key](#)

Key Security

**⚠️ Treat your API key like a password.**

Anyone who has your REST API key will be able to send notifications from your app. Do not expose the REST API key in your application code. Do not share it on GitHub or anywhere else online.

[Rest API Key documentation.](#)

Key ID	Name	Created At	Updated At
	Legacy API Key		<a href="#">Disable</a>

Identity Verification for email + external\_id (recommended) [?](#)

# Встановлення

## Expo SDK setup - OneSignal

Install the OneSignal Expo plugin using the Expo CLI.

```
npx expo install onesignal-expo-plugin
```

Add the react-native-onesignal package to your project.

```
npm install --save react-native-onesignal
```

# app.json

app.json

```
{
  "expo": {
    "ios": {
      "bundleIdentifier": "com.yourcompany.yourapp",
      "infoPlist": {
        "UIBackgroundModes": ["remote-notification"]
      },
      "entitlements": {
        "aps-environment": "development",
        "com.apple.security.application-groups": [
          "group.${ios.bundleIdentifier}.onesignal"
        ]
      }
    },
    "android": {
      "package": "com.yourcompany.yourapp"
    },
    "plugins": [
      [
        "onesignal-expo-plugin",
        {
          "mode": "development"
        }
      ]
    ]
  }
}
```

# Prebuild

При використанні бібліотек із нативним кодом (OneSignal) необхідно виконати **prebuild**, оскільки такі модулі потребують інтеграції на рівні Android/iOS.

**Prebuild** — це процес генерації native-проєкту (android/, ios/) та підключення залежностей з `app.json`.

```
npx expo prebuild
```

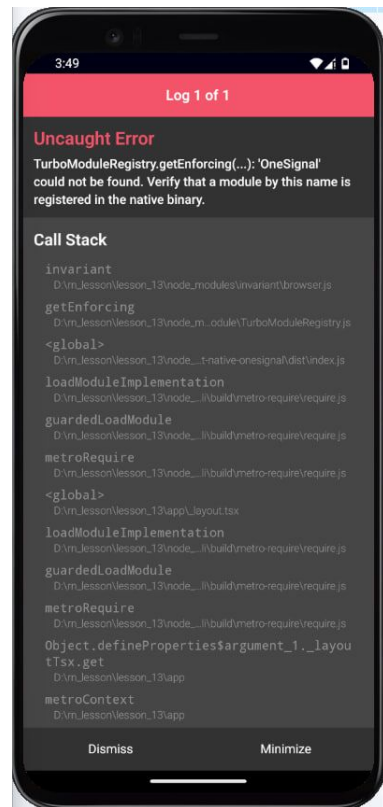
У більшості випадків він виконується **автоматично** при запуску:

- `npx expo run:android`
- `npx expo run:ios`

Без `prebuild`:

- нативний код не підключається
- бібліотека не працює

**Висновок:** `prebuild` є обов'язковим етапом для роботи native SDK у Expo-проєктах.



```
kipz_nsi@Best MINGW64 /d/rn_lesson/notifications (master)
$ npx expo run:android
✓ Created native directory
✓ Updated package.json
✓ Finished prebuild
> Port 8081 is being used by another process
✓ Use port 8082 instead? ... yes
> Building app...
Configuration on demand is an incubating feature.

> Configure project :
[ExpoRootProject] Using the following versions:
- buildTools: 36.0.0
- minSdk: 24
- compileSdk: 36
- targetSdk: 36
- ndk: 27.1.12297006
- kotlin: 2.1.20
- ksp: 2.1.20-2.0.1
```

```
> Press ? | show all commands
```

```
Installing D:\rn_lesson\notifications\android\app\build\outputs\apk\debug\app-debug.apk
```

# \_layout.jsx

Необхідно виконати базову ініціалізацію в коді. Зазвичай в головному файлі додатку (наприклад, `layout.jsx`) потрібно налаштувати обробник сповіщень і зареєструвати функцію отримання токена. Мінімальна конфігурація може включати встановлення глобального обробника, що визначає, як поводитися зі сповіщеннями, отриманими у foreground (наприклад, показувати їх чи ні)

```
useEffect(() => {
  OneSignal.Debug.setLogLevel(LogLevel.Verbose);

  OneSignal.initialize(ONESIGNAL_APP_ID);

  OneSignal.Notifications.requestPermission(true);

  // Обробник натискання на сповіщення
  Show usages new *
  const handleClick : (event: NotificationClickEvent) => void = (event: NotificationClickEvent) => {
    console.log("Notification clicked:", event);
  };

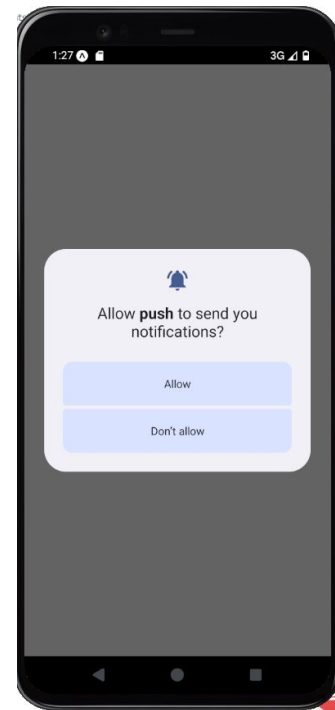
  // Обробник сповіщень у foreground
  Show usages new *
  const handleForeground : (event: NotificationWillDisplayEvent) => void = (event: NotificationWillDisplayEvent) => {
    event.getNotification().display();
  };

  OneSignal.Notifications.addEventListener("click", handleClick);
  OneSignal.Notifications.addEventListener(
    "foregroundWillDisplay",
    handleForeground
  );

  // Прибираємо слухачів при розмонтуванні, щоб уникнути витоків пам'яті
  return () => {
    OneSignal.Notifications.removeEventListener("click", handleClick);
    OneSignal.Notifications.removeEventListener(
      "foregroundWillDisplay",
      handleForeground
    );
  };
}, []);
```

# Запит дозволів користувача

Наступний крок – запитати у користувача дозвіл на надсилання йому сповіщень. Перед запитом добре перевірити поточний статус дозволів, щоб не питати двічі.



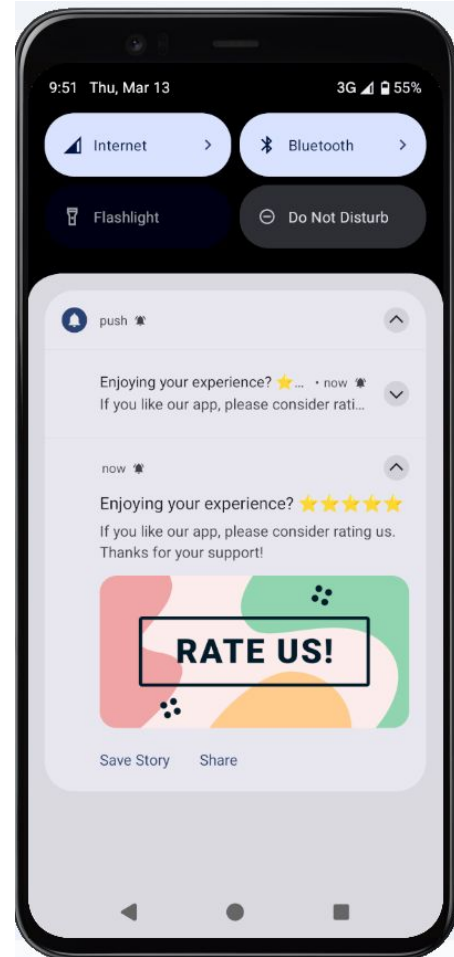
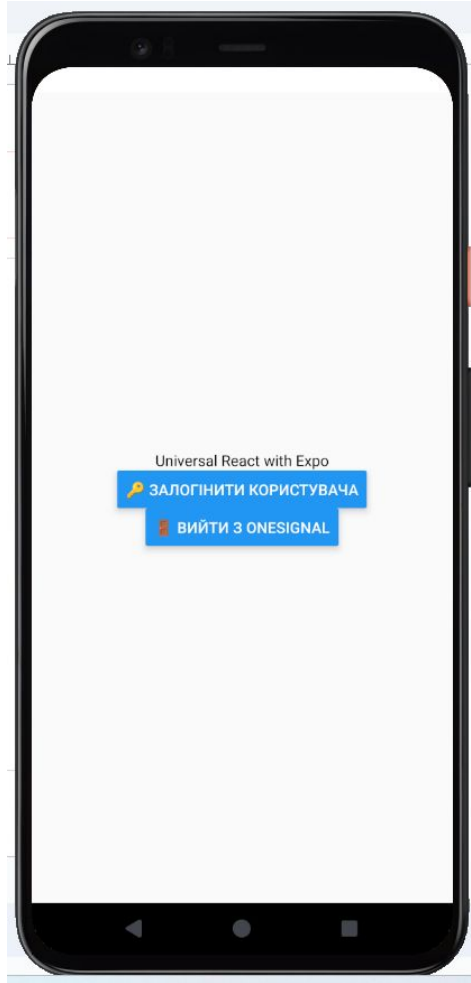
```

// Функція для логіну користувача в OneSignal
Show usages new *
const loginUser = async () => {
  OneSignal.Login( {externalId: 'kipz_hsi'} )
  OneSignal.User.pushSubscription.optIn();
  console.log('login')
};

// Функція для виходу з OneSignal
Show usages new *
const logoutUser = async () => {
  OneSignal.Notifications.clearAll();
  OneSignal.User.pushSubscription.optOut();
};


return (
  <View
    style={{
      flex: 1,
      justifyContent: "center",
      alignItems: "center",
      padding: 20
    }}
  >
    <Text>Universal React with Expo</Text>
    <Button title="🔑 Залогінити користувача" onPress={loginUser} />
    <Button title="🚪 Вийти з OneSignal" onPress={logoutUser} />
  </View>
);

```



## User Records


External ID ▼ Search 🔍

Channel	OneSignal ID ↕	Last Session ↕	First Session ↕
⋮	55163669-1a80-4577-837e-32b01146af0f		03/13/25, 9:50:59 pm
⋮	137269fb-5875-44f2-97b2-f3a4b1e62fdc		03/13/25, 9:15:48 pm
⋮ 	989fdfa7-1a83-48c9-8eb0-4756367acbe6	03/13/25, 10:10:23 pm	03/13/25, 9:24:19 pm

### Profile

- 📍 Country: Ukraine (UA)
- 🗺️ Language: en
- 🕒 Time Zone: GMT
- 👁️ Last Session: 2 hours ago
- 👁️ First Session: 2 hours ago

### Subscriptions (1)

- 📱  Google emulator (15) Subscribed ⋮


### IDs

- 🔑 OneSignal ID: 989fdfa7-1a83-48c9-8eb0-4756367acbe6 🔗
- 🔑 External ID: kipz\_nsi ⋮

## Subscription Records

Subscribed to Push

1

Subscribed to Email 


0

Subscription ID

Search



Select Segment...

Channel	Status	Status Details	Last Unsubscribed	Last Session	Usage Duration	Sessions	First Session
	Subscribed	-	03/13/25, 10:01:10 pm	03/13/25, 10:10:23 pm	1227	10	03/13/25, 9:24:19

# Push Notifications

All 13 Sent 13 Scheduled Drafts 0 A/B Tests

Search by name  Add Filter

Name	Labels	Status	Sent At	Created At	Delivered	CTR	
If you like our app, please consider rating us. Thanks for your support!		Delivered	03/13/25, 9:25:03 pm	03/13/25, 9:25:03 pm	1	0.00%	⋮
If you like our app, please consider rating us. Thanks for your support!		Delivered	03/13/25, 9:23:29 pm	03/13/25, 9:23:29 pm	1	0.00%	⋮
If you like our app, please consider rating us. Thanks for your support!		Delivered	03/13/25, 9:18:16 pm	03/13/25, 9:18:15 pm	1	0.00%	⋮
If you like our app, please consider rating us. Thanks for your support!		Delivered	03/13/25, 9:16:49 pm	03/13/25, 9:16:48 pm	1	0.00%	⋮
If you like our app, please consider rating us. Thanks for your support!		Delivered	03/13/25, 9:16:17 pm	03/13/25, 9:16:17 pm	1	0.00%	⋮
If you like our app, please consider rating us. Thanks for your support!		Delivered	03/13/25, 9:12:16 pm	03/13/25, 9:12:15 pm	1	0.00%	⋮
If you like our app, please consider rating us. Thanks for your support!		No Recipients	03/13/25, 9:05:22 pm	03/13/25, 9:05:21 pm	0	N/A	⋮
If you like our app, please consider rating us. Thanks for your support!		No Recipients	03/13/25, 9:04:03 pm	03/13/25, 9:04:03 pm	0	N/A	⋮
If you like our app, please consider rating us. Thanks for your support!		No Recipients	03/13/25, 8:54:36 pm	03/13/25, 8:54:36 pm	0	N/A	⋮
If you like our app, please consider rating us. Thanks for your support!		No Recipients	03/13/25, 8:41:35 pm	03/13/25, 8:41:35 pm	0	N/A	⋮
If you like our app, please consider rating us. Thanks for your support!		Delivered	03/13/25, 3:36:00 am	03/13/25, 3:36:00 am	1	0.00%	⋮

# Налаштування **OneSignal API**

Для надсилання push-сповіщень із бекенду потрібно налаштувати **OneSignal REST API**.

API використовується тоді, коли повідомлення має відправлятися не вручну з Dashboard, а програмно: після події в системі, дії користувача або зміни даних.

## Що потрібно підготувати

- **App ID**  
→ публічний ідентифікатор застосунку OneSignal
- **App API Key**  
→ приватний ключ для авторизації API-запитів  
→ зберігається тільки на бекенді або в `.env`

Обидва значення знаходяться в OneSignal Dashboard:

**Settings** → **Keys & IDs**. App ID можна використовувати в клієнтському коді, але App API Key є секретним і не має потрапляти у мобільний застосунок.

## Принцип роботи

1. Мобільний застосунок ініціалізує OneSignal SDK через App ID.
2. Користувач підписується на push-сповіщення.
3. Бекенд надсилає HTTP-запит до OneSignal API.
4. OneSignal доставляє push на потрібні пристрої.

# OneSignal API

## REST API overview - OneSignal

The screenshot shows the OneSignal website's REST API overview page. The page has a navigation bar with the OneSignal logo, language selection (English), a search bar, and a 'Ask AI' button. Below the navigation bar are links for Home, User Guides, Developer Guides, Tutorials, API Reference (selected), and Release Notes. A left sidebar contains links to Blog, GitHub, and YouTube, followed by a section titled 'Using the REST APIs' with a highlighted 'REST API overview' link, and an 'API reference' section with various API endpoints. The main content area is titled 'Using the REST APIs' and 'REST API overview', with a 'Copy page' button. The text describes the REST API's capabilities and provides a table of API goals and their corresponding endpoints.

OneSignal English

Search... Ctrl K Ask AI

Home User Guides Developer Guides Tutorials **API Reference** Release Notes

Blog  
GitHub  
YouTube

Using the REST APIs

**REST API overview**

Rate limits and error handling  
Idempotent API requests

API reference

Messages >  
Live Activities >  
Custom Events >  
Users >  
Subscriptions >  
Templates >  
Segments >  
Organization >  
Analytics >  
Deprecated >

Using the REST APIs

### REST API overview

Copy page

Programmatic access to OneSignal messaging, user management, segments, and data exports over HTTPS with rate limiting and retry support.

The OneSignal REST API follows the [REST architecture](#) and provides programmatic access to core messaging and user features. Use the API to send push notifications, emails, and SMS, manage Users, Subscriptions, Segments, export data, and configure apps.

### Which API to use

Goal	API
Send a push, email, or SMS	<a href="#">Create Message</a>
Add or update user data	<a href="#">Create User / Update User</a>
Target a group of users	<a href="#">Create Segments</a> or use <a href="#">Filters</a> inline
Export analytics or user data	<a href="#">CSV Export / CSV of Events</a>
Manage app configuration	<a href="#">Create App / Update App</a>

POST Push notification

POST /notifications?c=push

Send

## Push notification

Send a message using the push notification channel.

### Header

**Authorization** string required

Key YOUR\_APP\_API\_KEY

Your App API key with prefix **Key**. See [Keys & IDs](#).

### Body

**app\_id** string required

YOUR\_APP\_ID

Your OneSignal App ID in UUID v4 format. See [Keys & IDs](#).

**contents** object required

The main message body with [language-specific values](#). Supports [Message Personalization](#).

**contents.en** string required

Default message.

The required message language type. See

[Supported Languages](#).

+ Add new property

Push notification


cURL

```
curl --request POST \  
--url 'https://api.onesignal.com/notifications?c=push' \  
--header 'Authorization: Key YOUR_APP_API_KEY' \  
--header 'Content-Type: application/json' \  
--data '  
{  
  "app_id": "YOUR_APP_ID",  
  "contents": {  
    "en": "Default message."  
  },  
  "include_aliases": {  
    "external_id": [  
      "<string>"  
    ]  
  },  
  "target_channel": "push",  
  "include_subscription_ids": [  
    "<string>"  
  ],  
  "included_segments": [  
    "<string>"  
  ],  
  "excluded_segments": [  
    "<string>"  
  ],  
  "filters": [  
    {  
      "field": "tag",  
      "relation": "=",  
      "key": "<string>",  
      "value": "<string>"  
    }  
  ],  
}
```



## Keys & IDs

### OneSignal App ID

0d825712-c858-4d19-93a6-ed86867a17a3 

### API Keys

[+ Add Key](#)

 Treat your API key like a password.

Anyone who has your REST API key will be able to send notifications from your app. Do not expose the REST API key in your application code. Do not share it on GitHub or anywhere else online.

[Rest API Key documentation](#)

Key ID	Name	Created at	Updated at	
2xumuyem[REDACTED]	test	17.04.2025	17.04.2025	⋮
b6dqqn5dr[REDACTED]	Push	03.04.2025	03.04.2025	⋮

Identity Verification for email + external\_id (recommended) 

# Надсилання **push** через **OneSignal API**

```
console.log("API KEY:", ONESIGNAL_API_KEY);
const response : Response = await fetch("https://api.onesignal.com/notifications", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
    Authorization: `Key ${ONESIGNAL_API_KEY}`,
  },
  body: JSON.stringify({
    app_id: ONESIGNAL_APP_ID,
    target_channel: "push",
    // include_aliases таргетує конкретного юзера по external_id
    include_aliases: { external_id: ["kipznsi"] },
    contents: { en: message },
  }),
});
```

```
const response : Response = await fetch("https://api.onesignal.com/notifications", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
    Authorization: `Key ${ONESIGNAL_API_KEY}`,
  },
  body: JSON.stringify({
    app_id: ONESIGNAL_APP_ID,
    target_channel: "push",
    // include_aliases таргетує конкретного юзера по external_id
    // include_aliases: { external_id: ["kipznsi"] },
    // include_subscription_ids таргетує конкретного юзера по subscription_ids
    include_subscription_ids: [subId],
    // Заголовок сповіщення
    headings: { en: "Hello!" },
    // Кастомні дані – у handleClick через event.notification.additionalData
    data: { screen: "chat", chatId: "123" },
    // Кнопки дій
    buttons: [
      { id: "open", text: "Відкрити" },
      { id: "dismiss", text: "Закрити" },
    ],
    // Відкласти відправку
    // send_after: "2026-05-02 15:00:00 GMT+0300",
    contents: { en: message },
  }),
});
```

# Тестування **Push**-сповіщень

Тестувати push-сповіщення може бути непросто, оскільки вони залежать від реального пристрою та сторонніх сервісів. Ось кілька порад і методів для відлагодження цієї функціональності:

**Реальні пристрої:** Для перевірки потрібно мати фізичний пристрій (Android або iOS) з встановленим додатком. Переконайтеся, що пристрій підключений до інтернету, і що у налаштуваннях системи дозволено отримувати сповіщення для вашого додатку.

**Логування та дебаг:** Додайте `console.log` у всі критичні місця: при отриманні токена (логувати токен), при відправці (логувати відповіді від `fetch/SDK`). Перевіряйте консоль Expo або Xcode/Android Studio логкат, щоб бачити, що відбувається. Якщо сповіщення не приходить, але відповідь від Expo API була "ok", проблема може бути на рівні Apple/Google – таке буває, якщо пристрій офлайн. Якщо ж Expo API повертає помилку, уважно прочитайте її: часто буває "DeviceNotRegistered" (пристрій не зареєстрований) – це значить, що токен не дійсний. Інша можлива помилка – "MessageRateExceeded" (перевищено ліміт відправки).

# Тестування сценаріїв: Перевірте всі ситуації:

Коли додаток відкритий (foreground) – чи правильно викликається слухач, чи не дублюється показ (може, ви і банер показуєте, і в додатку – вирішіть, що вам потрібно).

Коли додаток у фоні – чи сповіщення з'являється в шторці, чи по натисканню відкриває потрібний екран.

Коли додаток закритий – чи приходить сповіщення.

Якщо у сповіщеннях є різні **канали** (для Android) – переконайтесь, що вони створені і мають потрібні налаштування (напр. канал `default` має увімкнені звуки, інакше сповіщення можуть приходити без звуку).

**Налагодження проблем доставки:** Якщо частина користувачів скаржиться, що не отримують сповіщення, треба з'ясувати, на якому етапі проблема. Переконайтесь, що у цих користувачів взагалі маєте токени і вони актуальні. Можливо, вони не дали дозвіл. Часто користувачі вручну вимикають сповіщення для додатку в системних налаштуваннях – тоді ваше відправлення буде успішним з точки зору серверів, але сам телефон його заблокує.

# Найкращі практики роботи з **Push**-сповіщеннями

Правильно налаштовані push-сповіщення здатні покращити взаємодію з користувачами, але зловживання ними може відлякати аудиторію. Розглянемо декілька найкращих практик, яких варто дотримуватись:

**Отримання згоди та очікування користувача:** Завжди пояснюйте, **навіщо** вашому додатку потрібні сповіщення, перед тим як запитувати дозвіл. Користувачі з більшою імовірністю погодяться, якщо розуміють цінність – наприклад, “Дозвольте нам надсилати вам сповіщення про статус ваших замовлень в реальному часі”. Запитуйте дозвіл у той момент, коли користувачеві це має сенс (наприклад, після того як він здійснив дію, для якої будуть сповіщення), а не одразу при першому запуску без контексту.

**Релевантність і цінність контенту:** Кожне сповіщення повинно нести користь або цікаву інформацію для отримувача. Уникайте розсилки тривіальних або дублюючих повідомлень. Персоналізуйте сповіщення під користувача, якщо можливо (звертайтеся на ім'я, надсилайте про ті функції, якими він користується). Користувач має відчувати, що сповіщення створені саме для нього, а не масовий спам.

**Оптимальна частота повідомлень:** Не перевантажуйте користувачів надмірною кількістю сповіщень. “Бомбардування” повідомленнями призводить до **втоми від сповіщень** – в результаті користувач може або вимкнути їх для вашого додатку, або взагалі видалити застосунок. Знайдіть баланс: наприклад, надіслати одне зведене сповіщення на день, ніж 10 окремих протягом години. Якщо є багато оновлень, можливо варто об’єднати їх (“У вас 5 нових лайків і 3 коментарі” замість 8 окремих сповіщень).

**Правильний таймінг:** Враховуйте час доби та часової пояси користувача. Надсилати сповіщення пізно вночі або рано-вранці (3-4 година ночі) – погана ідея, якщо це не щось критичне. Краще планувати розсилки на денний час, або навіть дозволити користувачу налаштувати *тихі години*, коли його не турбувати. Якщо у вас глобальні користувачі – сегментуйте по часових поясах, щоб кожен отримував повідомлення у прийнятний локальний час.

**Давайте користувачу контроль:** Хороша практика – дозволити користувачам обирати, які типи сповіщень вони хочуть отримувати. Можна в налаштуваннях додатку зробити перемикачі. Наприклад, хтось хоче отримувати сповіщення про особисті повідомлення, але не про рекламні акції. Хтось навпаки. Надаючи вибір, ви знижуєте шанс, що користувач повністю відключить **усі** сповіщення для застосунку. На Android для цього є механізм **Notification Channels** – різні канали можуть відповідати різним категоріям (оновлення, нагадування, пропозиції тощо), і користувач системно може вимкнути окремі канали. Ви ж можете використовувати канали, щоб коректно класифікувати сповіщення.

# OneSignal Status



SUBSCRIBE TO UPDATES

All Systems Operational

Uptime over the past 90 days. [View historical uptime.](#)

