

하필 우리 앱에  
채팅이 필요할 때

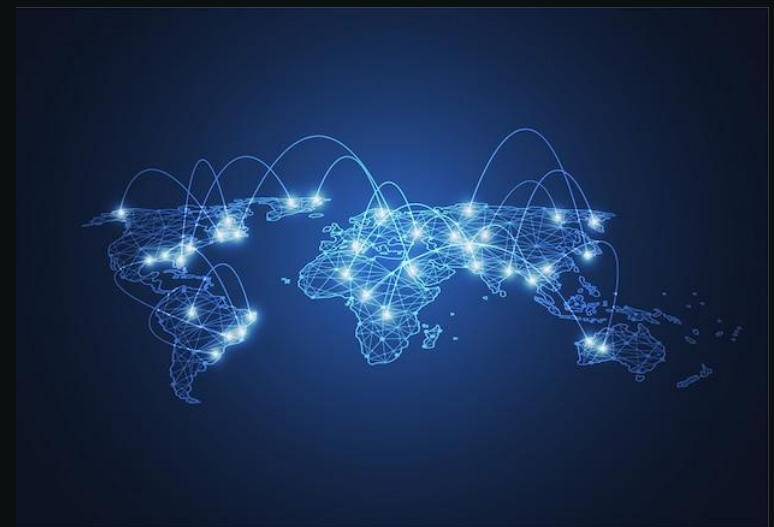
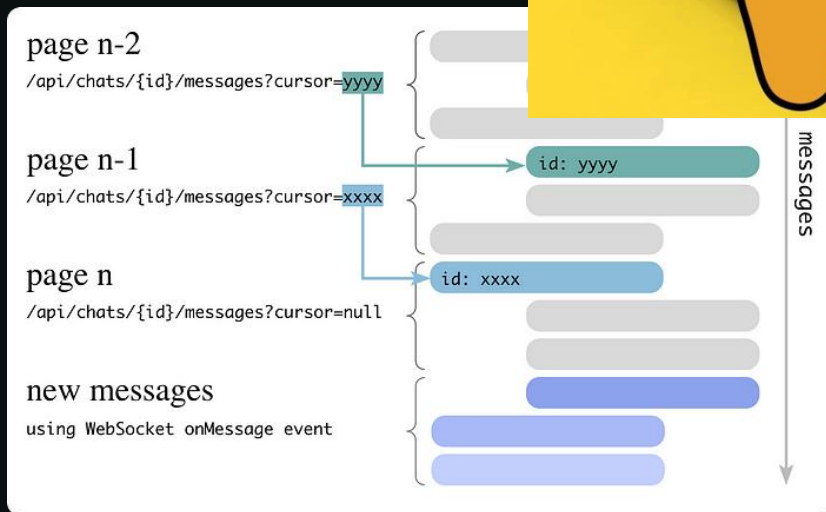
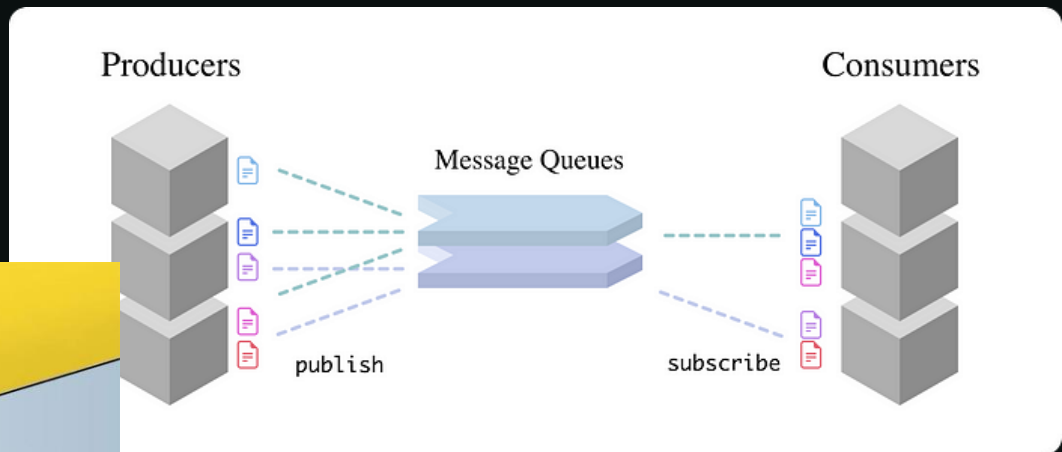
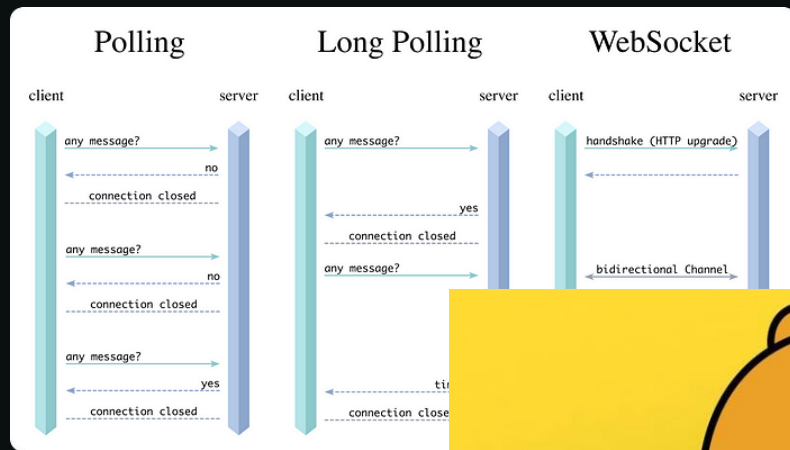
흑우마스터

엑스크루 CTO

jojangwon@kakao.com



**?? : 우리 앱에 채팅 기능 좀 넣어주세요**







**?? : 앵? 채팅 다른 앱들 다 있던데 못해요?**

HONEYCAM

 **골목식당**

수요일 밤 11시 10분



**유료 서비스**

# Talk Plus

💰 기본 요금

₩290,000~

- 다른 어떤 솔루션보다 가볍게 시작할 수 있습니다.
- 중소형 서비스도 부담 없이 채팅 기능을 도입해 성장을 달성할 수 있습니다.

👤 기본 사용량

월 누적 DAU  
60,000

- 누적 DAU를 기준으로 사용량을 책정합니다.
- 누적 60,000 DAU (약 MAU 10,000)에 상응하는 기본 사용량을 제공합니다.

👥 동시접속자 수

무제한

- 서버에 동시 접속하는 사용자 수를 제한하지 않습니다.
- 추가 과금에 대한 걱정없이 쾌적하게 인프라를 사용하세요.



# Talk Plus



[더 알아보기 ↗](#)



[더 알아보기 ↗](#)



[더 알아보기 ↗](#)



[더 알아보기 ↗](#)

{REST} Rest API

[더 알아보기 ↗](#)



[더 알아보기 ↗](#)

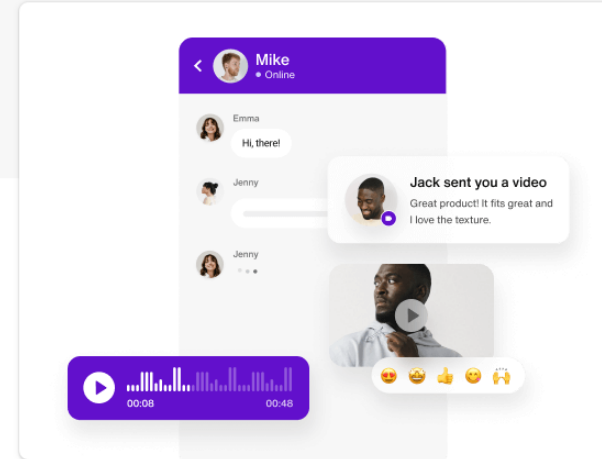
## Chat

A suite of tools and resources designed for building and managing real-time chat experiences. It covers the necessities for developers to seamlessly integrate chat functionality into their application.

With our Chat SDK, developers have access to a wide range of functions for creating custom chat experiences to suit their business needs.

To streamline the development process, you can use the UIKit for Chat. This is a collection of pre-built UI components that developers can use to quickly implement user-friendly chat interfaces.

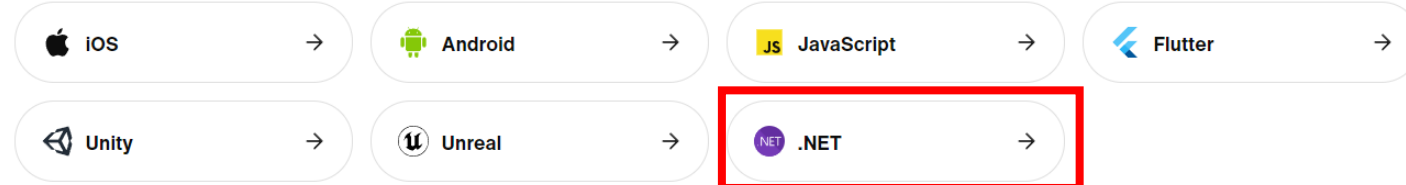
Additionally, our Platform API provides server-side control over your chat functionality. While the native SDKs handle many of the requests and responses on the client side, Platform API allows developers to manage and manipulate chat functionalities directly from their server.



## UIKit



## SDKs



## 디벨로퍼

센드버드의 채팅을 자유롭게  
테스트해보세요

## 무료

센드버드의 채팅을 자유롭게  
테스트해보세요

**100 MAU**

동시접속 10명 가능

### 디벨로퍼 기능:

- 모든 프로 기능 포함
- 신용카드 결제 불필요
- 커뮤니티를 통한 고객 지원

무료 사용하기

## 스타터 5천

기본적인 기능으로 부담 없이  
시작해보세요

**\$399/월**



### 스타터 기능:

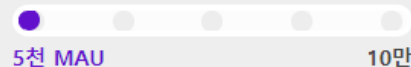
- 최신 메신저 필수요소
- 기본관리도구
- 티켓을 통한 고객지원

무료 사용하기

## 프로 5천

더 풍부한 기능으로 비즈니스를  
성장시키세요

**\$599/월**



### 프로 기능:

(모든 스타터 기능 포함)

- 메시지 자동 번역
- 고급관리 도구 및 필터
- 티켓을 통한 고객지원

무료 사용하기

## 엔터프라이즈

센드버드 플랫폼의 가장 강력한 기능들을  
사용해보세요

수백만 MAU를  
위한 맞춤형 가격

### 엔터프라이즈 기능:

(모든 프로 기능 포함)

- 데이터 내보내기
- 전용서버 옵션
- 최우선 고객지원

상담 요청하기

## 무료

모든 플랫폼 기능을 활용하세요

- ✓ 월간 메시지 600만 개
- ✓ 동시 채널 200개
- ✓ 동시 연결 200개
- ✓ 신용카드가 필요하지 않습니다
- ✓ 실시간 채팅 및 이메일 지원

시작하다

## 종량제

성장에 따라 필요한 유연성과 제어력

- ✓ \$2.50 / 백만 메시지
- ✓ \$15.00 / 동시 채널 1000개
- ✓ \$15.00 / 천 동시 연결

을 더한:

- ✓ 99.999% 가동 시간 SLA
- ✓ 실시간 채팅 및 이메일 지원

계획 만들기

## 커스텀

무제한적인 규모와 미션 크리티컬 요구 사항에 대한 계획

- ✓ 시간별 청구 옵션
- ✓ MAU 모델링 가격
- ✓ 규모에 따른 대량 구매 할인

을 더한:

- ✓ 99.999% 가동 시간 SLA
- ✓ 건축 지도
- ✓ 연중무휴 미션 크리티컬 지원
- ✓ 보안 및 규정 준수 인증
- ✓ 지역 데이터 라우팅
- ✓ CNAME, SSO, Datadog 등...

문의하기

























## Pricing Information

Below are the total costs for these different subscription durations.  
Additional taxes or fees may apply.

Abyl Realtime				
Units	Description	1 MONTH	12 MONTHS	24 MONTHS
Bootstrap	100 connections & channels, 3 mil msgs/month, 16KiB max msg size	\$0	\$0	\$0
Micro	500 connections & channels, 10 mil msgs/month, 64KiB max msg size	\$55	\$660	\$1,320
Small	1,000 connections & channels, 30 mil msgs/month, 64KiB max msg size	\$89	\$1,068	\$2,136
Medium	2,500 connections & channels, 100 mil msgs/month, 64KiB max msg size	\$199	\$2,388	\$4,776
Large connections+	25,000 connections & channels, 200 mil msgs/month, 64KiB max msg size	\$799	\$9,588	\$19,176
Large messages+	10,000 connections & channels, 600 mil msgs/month, 64KiB max msg size	\$999	\$11,988	\$23,976



# Ably Realtime

<div><b>JavaScript</b> </div> <div>Ably SDK for JavaScript.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>Java</b> </div> <div>Ably SDK for Java.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>Python</b> </div> <div>Ably SDK for Python.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>Flutter</b> </div> <div>Ably SDK for Flutter.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>PHP</b> </div> <div>Ably SDK for PHP.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>Android</b> </div> <div>Ably SDK for Android.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>
<div><b>React</b> </div> <div>Ably React Hooks package.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>C#.NET</b> </div> <div>Ably SDK for C#.NET.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>Node.js</b> </div> <div>Ably SDK for Node.js.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>Kotlin</b> </div> <div>Ably SDK for Kotlin.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>Unity</b> </div> <div>Ably SDK for Unity.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>	<div><b>Xamarin</b> </div> <div>Ably SDK for Xamarin.</div> <div><a href="#">Setup →</a></div> <div><a href="#">GitHub</a> </div>



**직접 구축 하시는게.. 마음 편하시죠?**

**오픈소스 셀프 호스팅**

# Matrix & Rocket Chat




# Matrix

**Your Privacy Isn't a Commodity!**

- Self Hosted
- Open Source
- End to End Encrypted
- Federated
- Audio / Video / Chat

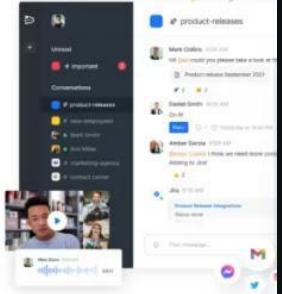
BIG DATA



## Let every conversation flow - without compromise

Secure and compliant collaboration platform.  
Own your data, customize anything, integrate everything.

[Get started — it's free](#) ↓ Scroll down



# Nakama

Game Console SDKs

[Unity/.Net](#)

Unreal

C++

JavaScript

Dart (Flutter)

Godot 3

Godot 4

Swift

Java/Android

Cocos2d-x C++

Cocos2d-x JavaScript

Defold

Snippets

Authoritative In-game Items

Authoritative Storage

Banning Users

Custom Authentication

Updating Group Metadata

Updating User Metadata

[Nakama](#) / [Client Libraries](#) / [Unity/.Net](#)

## Nakama Unity Client Guide

This client library guide will show you how to use the core Nakama features (without full game logic or UI) of an [Among Us \(external\)](#) inspired game called



Sagi-shi gameplay

CLIENT

```
1 var roomUsers = new List<UserPresence>(10);
2 socket.ReceivedChannelPresence += presenceEvent =>
3 {
4     foreach (var presence in presenceEvent.Leaves)
5     {
6         roomUsers.Remove(presence);
7     }
8
9     roomUsers.AddRange(presenceEvent.Joins);
10    Console.WriteLine("Room users: [{0}]", string.Join(",\n ", roomUsers));
11 };
12
13 var roomName = "PizzaFans";
14 var persistence = true;
15 var hidden = false;
16 var channel = await socket.JoinChatAsync(roomName, ChannelType.Room, persistence, hidden);
17 roomUsers.AddRange(channel.Presences);
```

### Receive messages


A user joins a chat channel to start receiving messages in real-time. Each new message is received by an event handler. Messages are delivered in the order they are handled by the server.

CLIENT

```
1 socket.ReceivedChannelMessage += message =>
2 {
3     Console.WriteLine("Received: {0}", message);
4     Console.WriteLine("Message has channel id: {0}", message.ChannelId);
5     Console.WriteLine("Message content: {0}", message.Content);
6 };
```



# Nakama

설치	 <b>Nakama</b> <div> <span>Star 8,306</span> <span>Fork 1,035</span> </div> <p>Nakama는 소셜 및 실시간 게임 및 앱을 위한 확장 가능한 서버입니다.</p> <p>Nakama 서버를 사용하면 앱과 게임에 사용자 인증, 소셜 네트워킹, 저장소 및 실시간 데이터 교환을 추가할 수 있습니다. 개발업체 <a href="#">Heroic Labs</a> 모든 소셜 및 실시간 게임 및 앱에 도입되기는 어렵지만 필수적인 서비스를 처리합니다.</p> <p>이 서버는 대규모로 실행되도록 설계되었습니다. 나카마는 <a href="#">분산 클러스터 식으로 작동합니다</a> 따라서 게임이나 앱이 App Store에 소개된 경우, 인스턴스만 몇 개 더 시작하면 추가 로드를 쉽게 처리할 수 있습니다.</p> <p>개발하는 동안 macOS, Linux 또는 Windows 시스템에서 이 서버를 실행할 수 있습니다. 프로덕션으로 이동하려면 클라우드로 이동하거나 Heroic Labs를 사용 합니다. <a href="#">Heroic Cloud</a>.</p> <h2>기능</h2> <p>서버에서 모든 것은 처리하는 동안 프로젝트 구축에 집중할 수 있습니다.</p> <ul style="list-style-type: none"> <li><a href="#">사용자 계정</a></li> <li><a href="#">소셜 프로필</a></li> <li><a href="#">실시간 채팅</a></li> <li><a href="#">데이터 저장소</a></li> <li><a href="#">멀티플레이어 대결</a></li> <li>및 그 이상</li> </ul> <p><a href="#">사용자 계정</a></p> <p>사용자란 서버 내의 ID입니다. 모든 사용자는 등록되어 있고 프로필이 있어 다른 사용자가 찾고 친구가 되거나 그룹에 가입하고 채팅할 수 있습니다. 사용자는 기록을 소유하고 다른 사용자와 공개 정보를 공유하고 다양한 소셜 제공자를 통해 인증할 수 있습니다.</p> <p><a href="#">그룹</a></p> <p>그룹을 사용하면 플레이어는 팀 또는 커뮤니티 단위로 함께 채팅하고 플레이할 수 있습니다. 이러한 플레이어 그룹은 공개 또는 비공개일 수 있으며, 가입하려면 그룹 관리자의 승인이 필요합니다.</p> <p><a href="#">친구</a></p> <p>친구는 소셜 커뮤니티를 구축하는 좋은 방법입니다. 사용자는 자신의 친구 목록에 다른 사용자를 추가하고, 온라인 상태 또는 마지막으로 온라인 상태였던 시간을 확인하고, 실시간으로 함께 채팅하고, 게임 플레이 또는 협업에서 함께 상호 작용할 수 있습니다.</p> <p><a href="#">실시간 채팅</a></p> <p>직접, 그룹 또는 공개 채팅 채널을 통해 커뮤니티 상호 작용을 활성화합니다. 사용자는 원하는 채팅을 찾아 가입하거나 새 채팅을 만들 수 있습니다. 이 채널의 메시지가 영구적이 되어 오프라인 사용자가 나중에 검토할 수 있거나, 온라인 사용자에게만 표시될 수 있습니다.</p> <p><a href="#">인증</a></p> <p>인앱 알림을 사용하면 한 명 이상의 사용자에게 메시지를 쉽게 브로드캐스트할 수 있습니다.</p> <p><a href="#">순위표</a></p> <p>순위표는 모든 게임에 소셜 및 경쟁 요소를 추가할 수 있는 좋은 방법입니다.</p>
Docker Compose	
Windows	
macOS 바이너리	
Linux	
CLI 명령	
구성	
업그레이드	
Nakama 콘솔	
상태	
사용자 관리	
구성	
런타임 모듈	
계정	
사용자 그룹	
저장소	
순위표	
Chat Messages (en)	
대결	
API 탐색기	
아키텍처 개요	
데이터 개인 정보	
벤치마크	
릴리스 정보	

Nakama / 개념 / 상태

CLIENT.NET/Unity

# 상태

Nakama 사용자는 연결 시 상태 메시지를 설정하고 온라인 상태에서 업데이트할 수 있습니다.

사용자는 상태 변경에 대한 알림을 받기 위해 서로 서로 팔로우할 수 있습니다. 이것은 친구가 온라인 상태인지 그리고 최근 활동 상태가 무엇인지 알려고 할 때 아주 좋습니다

상태는 각 연결에 대해 설정되며 사용자가 연결을 끊으면 사라집니다. 사용자가 여러 장치에서 연결된 경우 각 장치의 상태가 다를 수 있습니다.

## 상태 설정

기본적으로 사용자는 처음 연결하는 경우 상태가 없으며 팔로워에게 온라인으로 표시되지 않습니다. 온라인으로 표시되려면 사용자가 상태를 설정해야 합니다.

상태는 사용자가 팔로워에게 보내는 문자 메시지처럼 간단할 수 있거나, 사용자가 현재 포함되어 있는 [실시간 멀티플레이어 대결 ID](#)와 같은 복잡한 정보가 있는 구조화된 JSON 문자열이 될 수 있습니다. 따라서 친구들이 바로 참석하여 가입할 수 있습니다!

CLIENT

```
1  await socket.UpdateStatusAsync("Hello everyone!");
```

이 작업으로 필요할 때마다 상태를 설정하고 업데이트할 수 있습니다.

## 오프라인으로 표시

사용자를 오프라인으로 표시하거나 “보이지 않는” 상태로 나타내려면 해당 상태를 지웁니다. 팔로워는 사용자가 연결을 끊을 때와 같은 상태 업데이트를 받습니다.

CLIENT

```
1  await socket.UpdateStatusAsync(null);
```

# Nakama

Game Console SDKs

[Unity/.Net](#)

Unreal

C++

JavaScript

Dart (Flutter)

Godot 3

Godot 4

Swift

Java/Android

Cocos2d-x C++

Cocos2d-x JavaScript

Defold

Snippets

Authoritative In-game Items

Authoritative Storage

Banning Users

Custom Authentication

Updating Group Metadata

Updating User Metadata

Installation

The client is available from the:

- [Unity Asset Store](#)
- [Heroic Labs GitHub releases page](#)

The `Nakama.unitypackage` contains all source code and DLL dependencies required in the client code.

After downloading the file:

- Drag or import it into your Unity project
- Set the editor scripting runtime version to .NET 4.6 (from the **Edit** → **Project Settings** → **Player** → **Configuration** menu).
- From the **Assets** menu create a new C# script and a [client object](#)

Unity 2019.4.1+

Alternatively, you can checkout a specific release or commit by adding the following to the `manifest.json` file in your project's `Packages` folder:

```
"com.heroiclabs.nakama-unity": "https://github.com/heroiclabs/nakama-unity.git?path=/Packages/Nakama#<commit | tag>"
```

Updates

New versions of the Nakama Unity Client and the corresponding improvements are documented in the [Changelog](#).

Unity / .NET SDK Differences

In general, this client guide can also be used in conjunction with the [Nakama .NET SDK](#) with a few minor differences which are outlined below.

Socket Creation

The `Client.NewSocket` extension is unavailable in the .NET SDK. Instead use:

```
1 var socket = Socket.From(client);
```

Logging

Logging to the console in this guide uses the Unity specific `Debug.LogFormat` method. In the .NET SDK use:

```
1 Console.WriteLine("Hello, {0}!", "Nakama");
```

## Nakama .NET

## Nakama

.NET client for Nakama server written in C#.

[Nakama](#) is an open-source server designed to power modern games and apps. Features include user accounts, chat, social, matchmaker, realtime multiplayer, and much [more](#).

This client implements the full API and socket options with the server. It's written in C# with minimal dependencies to support Unity, Xamarin, Godot, XNA, and other engines and frameworks.

Full documentation is online - <https://heroiclabs.com/docs>

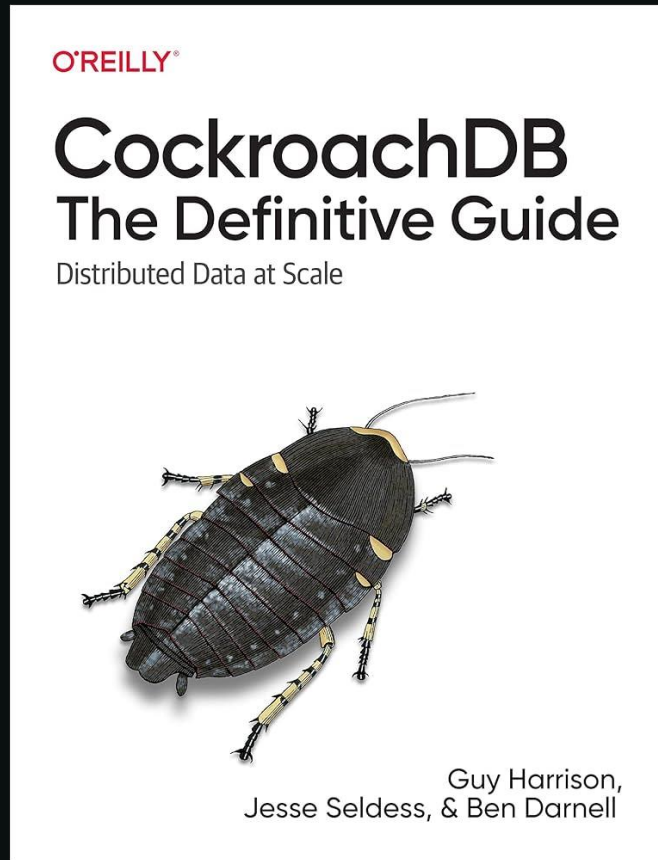
## Getting Started

You'll need to setup the server and database before you can connect with the client. The simplest way is to use Docker but have a look at the [server documentation](#) for other options.

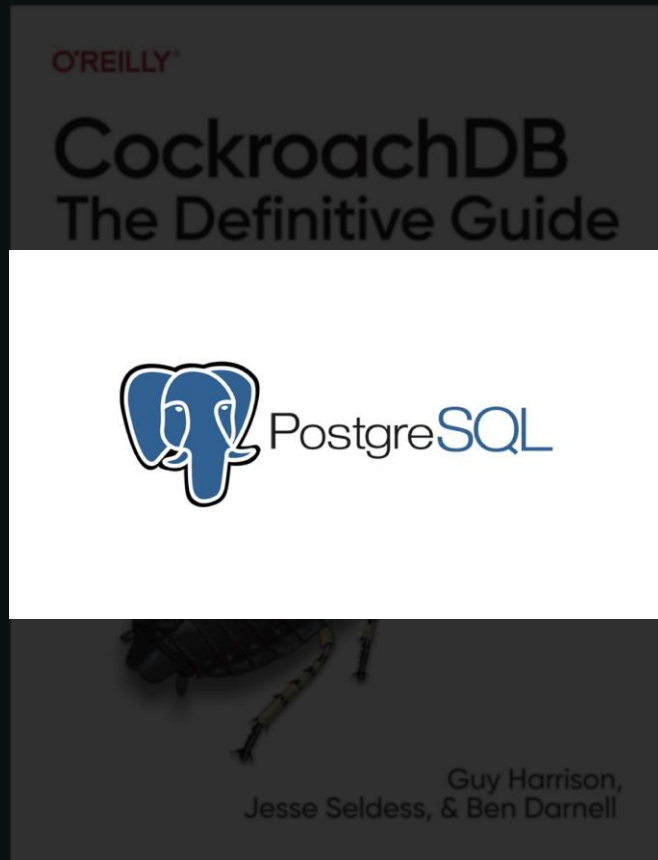
1. Install and run the servers. Follow these [instructions](#).
2. Download the client from the [releases page](#) and import it into your project. You can also [build from source](#).
3. Use the connection credentials to build a client object.

```
// using Nakama;
const string scheme = "http";
const string host = "127.0.0.1";
const int port = 7350;
const string serverKey = "defaultkey";
var client = new Client(scheme, host, port, serverKey);
```

# Nakama



# Nakama



# Nakama

나카마 CPU

Heroic Cloud 배포의 평균 범위

1

2

4

8

12

16

24

64세 이상

데이터베이스 CPU

Heroic Cloud 배포의 평균 범위

1

2

삼

4

6

8

12

16세 이상

✔ 지속적인 통합

✖ 회복력

✖ 고가용성

✖ 확장

Heroic Cloud의 Nakama 가격:

비용: \$600 /월

Heroic Cloud에 가입하세요

모든 계획에는 다음이 포함됩니다.

✔ 나카마엔터프라이즈

✔ DAU/MAU/CCU 제한 없음

✔ 전용 하드웨어 리소스

✔ 전용 관리형 데이터베이스 인스턴스

✔ 서버 모듈을 위한 지속적인 통합

✔ 자동 데이터베이스 백업

✔ 측정항목 및 모니터링

✔ 시스템 확장

✔ 로드 밸런서

✔ SSL 인증서

✔ 로그 집계

✔ [나카마에 대해 더 알아보세요.](#)

✔ [Heroic Cloud에 대해 자세히 알아보세요.](#)

Nakama on Heroic Cloud

Overview

Pricing

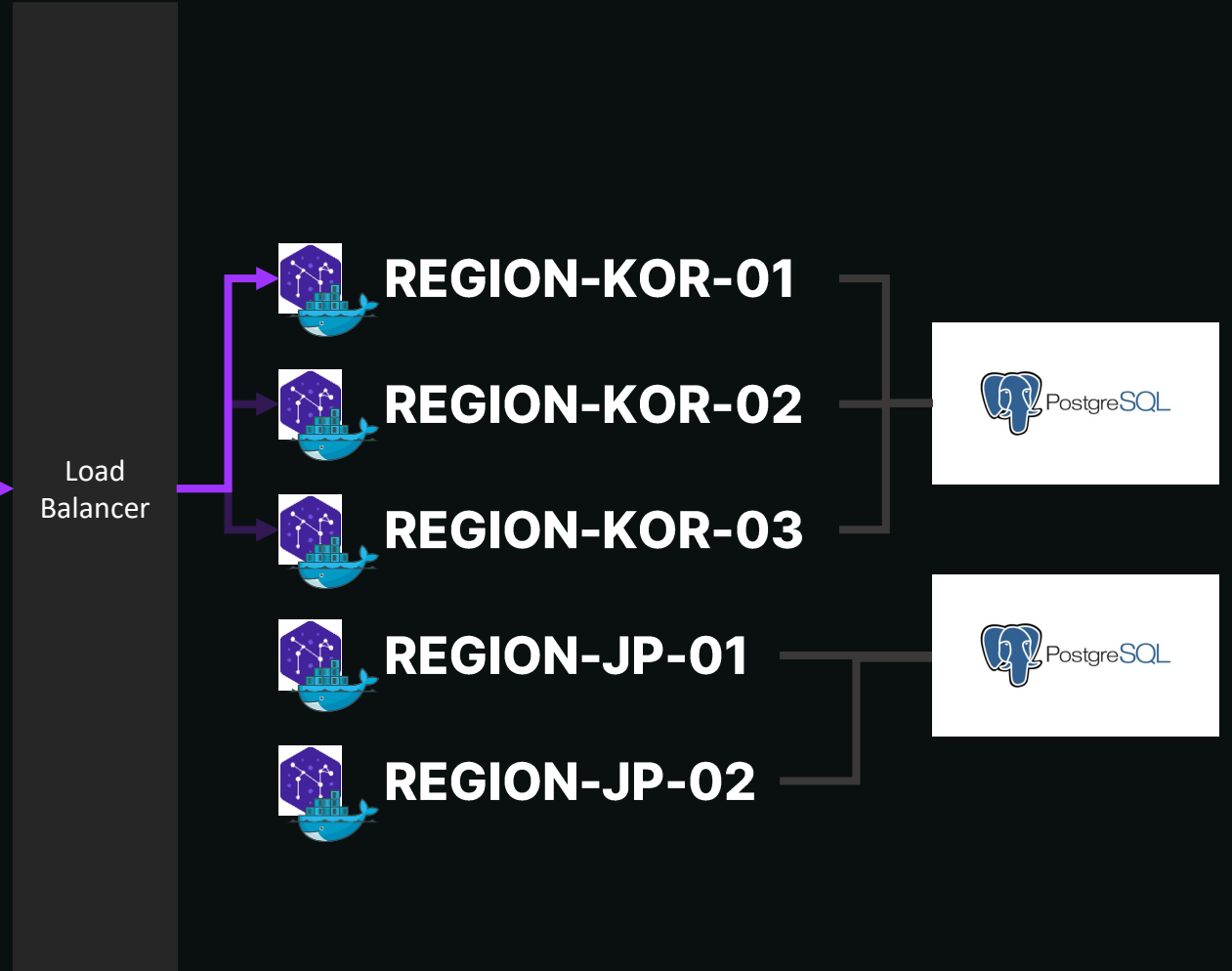
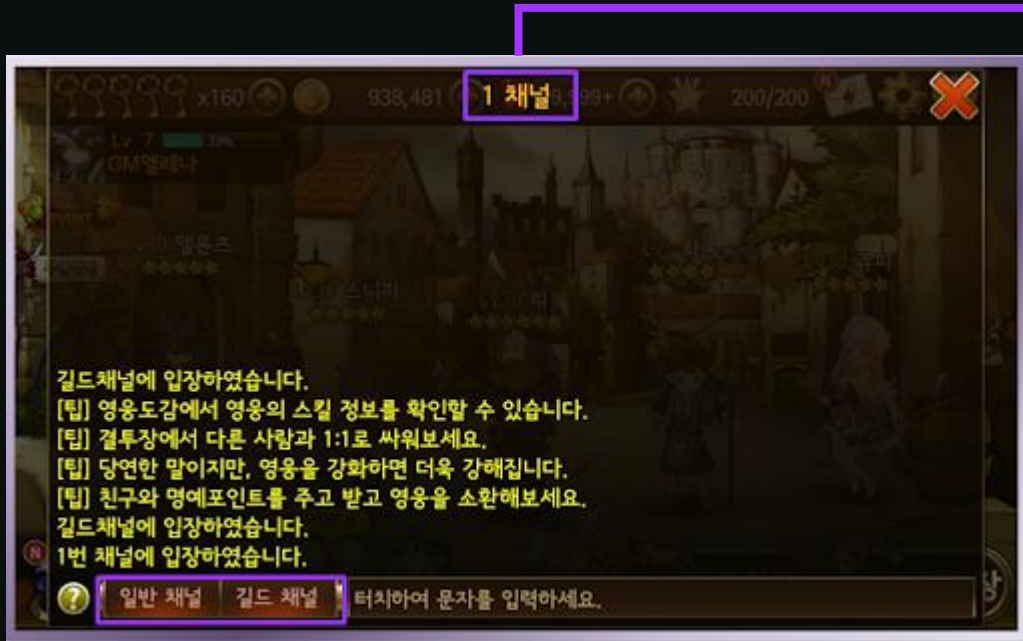
## Pricing Information

This software is priced along a consumption dimension. Your bill will be determined by the number of units you use. Additional taxes or fees may apply.

Nakama on Heroic Cloud	
Units	Cost
Nakama CPU (per day)	\$13.33 / unit
Database CPU (per day)	\$6.67 / unit
Nakama CPU - development (per day)	\$9.34 / unit
Database CPU - development (per day)	\$4.66 / unit



# Nakama



**데모**  
**Nakama**

**직접 구축**

**선택지 1**

**SignalR with RDB**

## SignalR with RDB

- ASP.NET SignalR
- ASP.NET Core SignalR
- Azure SignalR Service



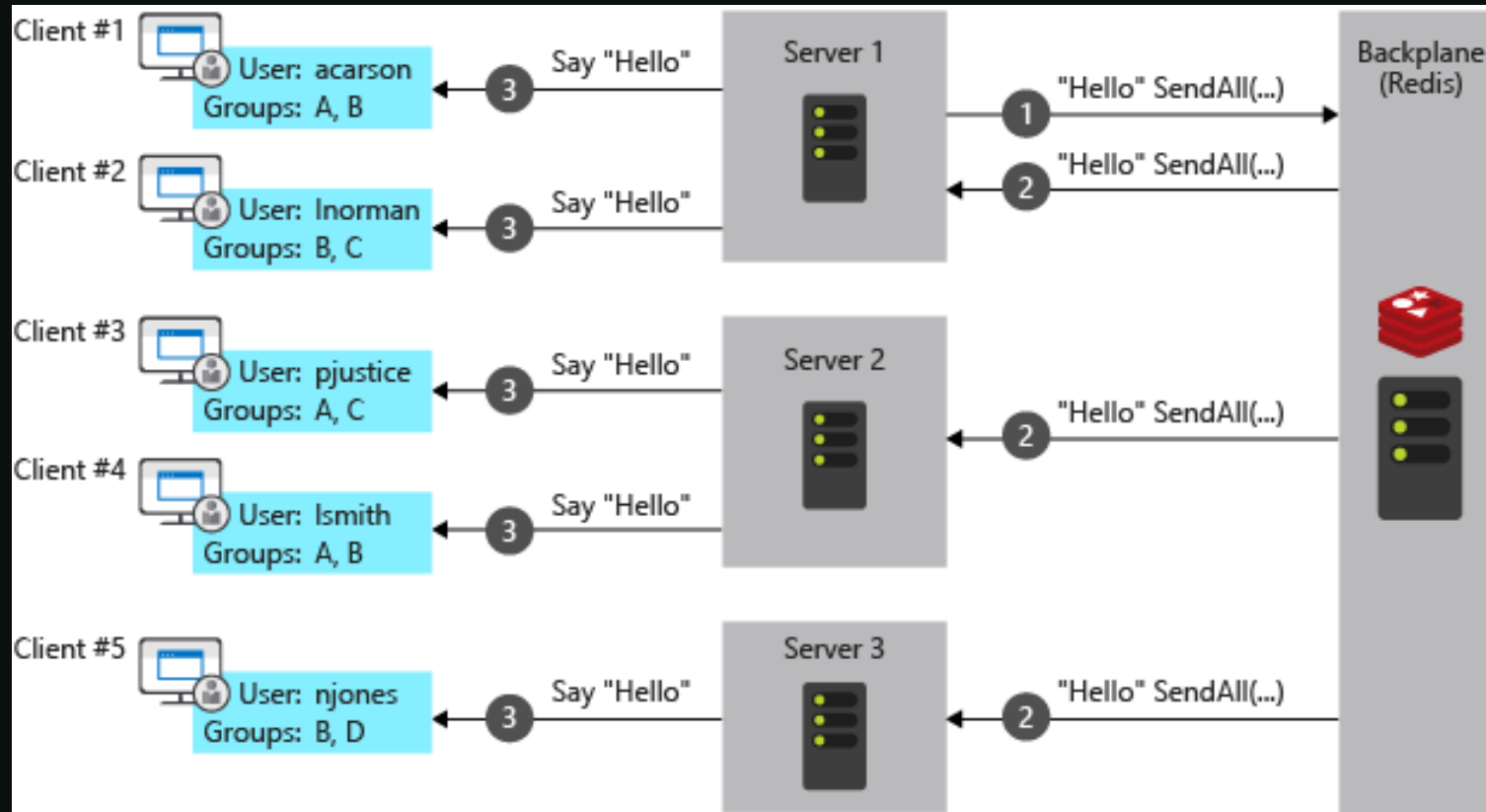
## SignalR with RDB

- ASP.NET SignalR
- ASP.NET Core SignalR
- Azure SignalR Service
- JSON 기반 텍스트 프로토콜
- MessagePack 기반 바이너리 프로토콜
- Websocket
- 완전 관리형 서비스

## SignalR with RDB

- ASP.NET SignalR
- ASP.NET Core SignalR
- Azure SignalR Service
- JSON 기반 텍스트 프로토콜
- MessagePack 기반 바이너리 프로토콜
- Websocket
- 완전 관리형 서비스
- 기존 RDB 사용 가능
- Redis Backplane
- SSE, HTTP Long polling

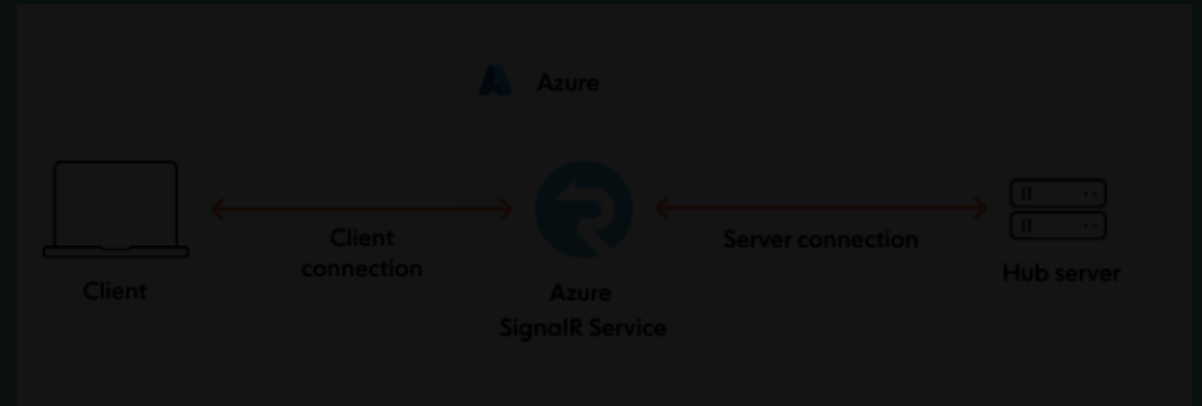
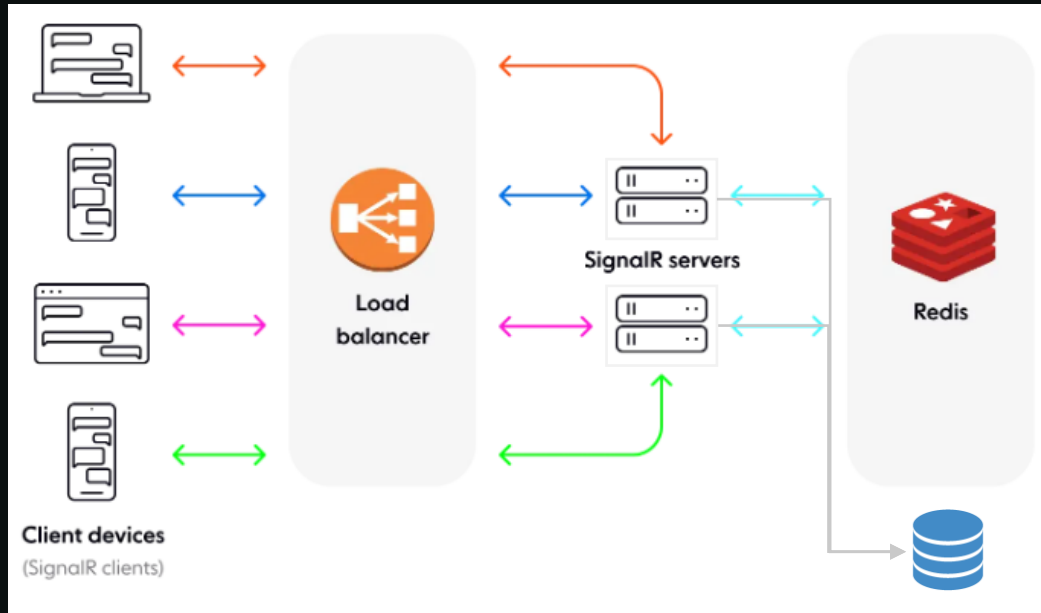
# SignalR with RDB



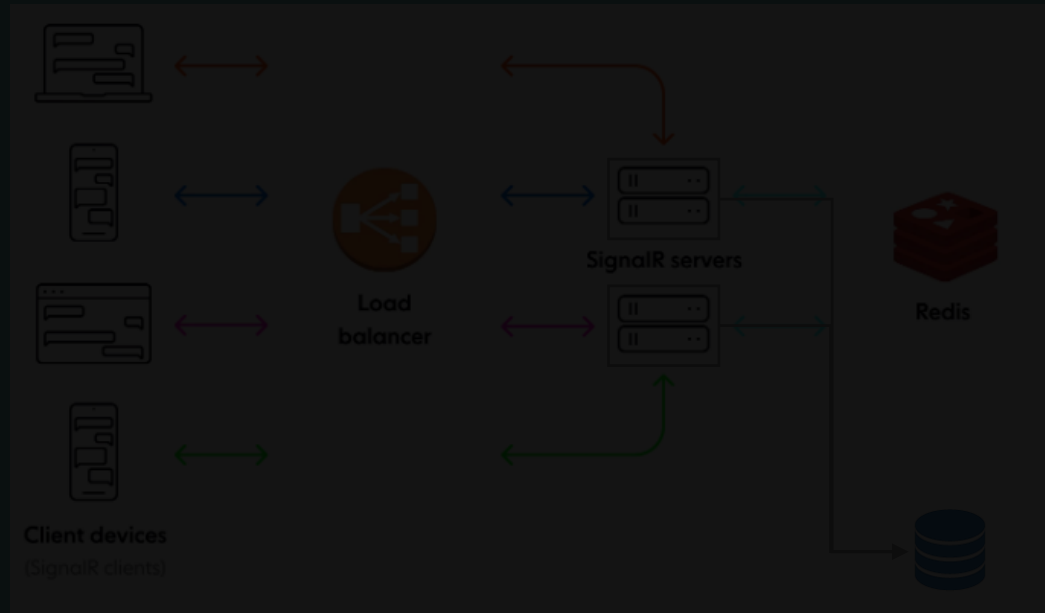
Redis Backplane

# SignalR with RDB

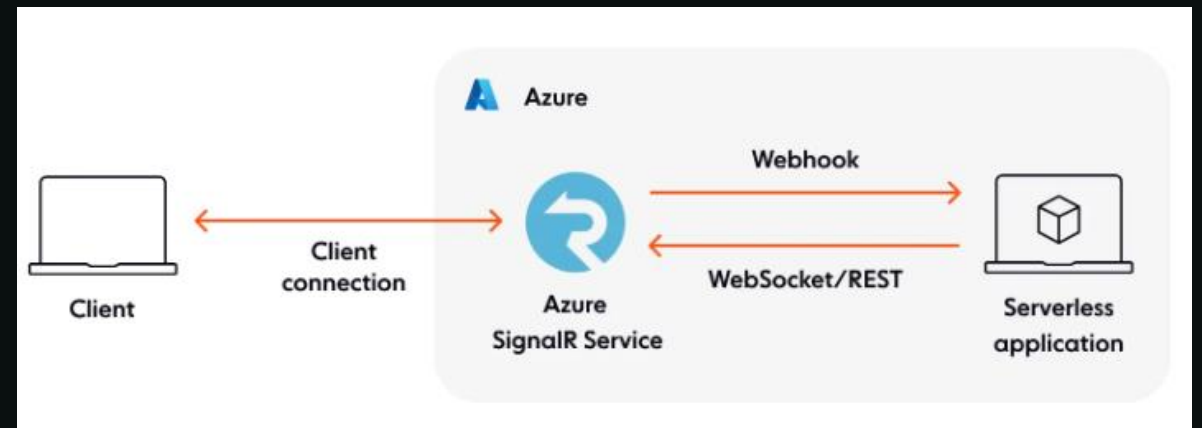
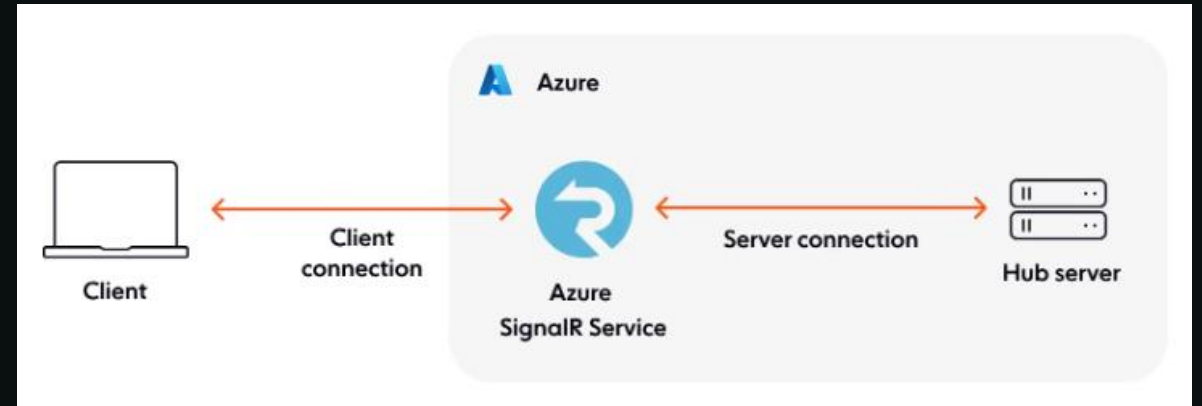
## Self Hosted SignalR



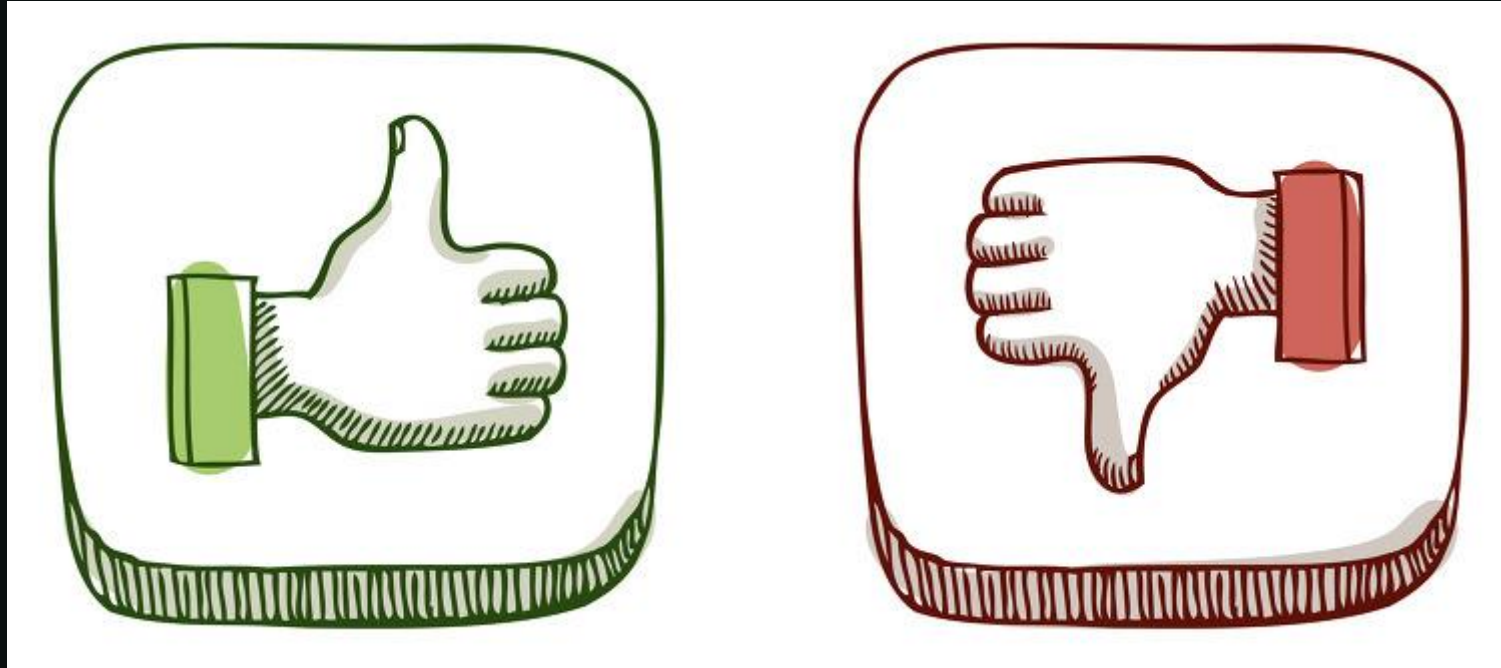
# SignalR with RDB



## Azure SignalR



## SignalR with RDB



**선택지 2**

**Supabase Realtime**



# Supabase



Build And Test passing nuget v1.0.0

Documentation can be found [below](#), on the [Supabase Developer Documentation](#) and additionally in the [Generated API Docs](#).

**\*\*CHANGELOG is available in the repository root. \*\***

## [NOTICE FOR v1.0.0]

- The `supabase-csharp` Nuget package has been renamed to `Supabase` and a depreciation notice set to encourage adoption.
- Almost all APIs stay the same when migrating from v0.16.x *except* the change in namespace from `Postgrest` to `Supabase.Postgrest`. Some minor refactoring will be required in the codebase.
- The assembly name has been changed from `supabase` to `Supabase`.

## Features

- ✓ Integration with [Supabase.Realtime](#)
  - Realtime listeners for database changes
- ✓ Integration with [Postgrest](#)
  - Access your database using a REST API generated from your schema & database functions
- ✓ Integration with [Gotrue](#)
  - User authentication, including OAuth, email/password, and native sign-in
- ✓ Integration with [Supabase Storage](#)
  - Store files in S3 with additional managed metadata
- ✓ Integration with [Supabase Edge Functions](#)



**supabase-csharp** by Joseph Schultz <joseph@acupofjose.com>, **89.8K** downloads  
A C# implementation of the Supabase client

This package version is deprecated. Use [Supabase](#) instead.



**supabase-storage-csharp** by Joseph Schultz <joseph@acupofjose.com>, **75.5K** downloads  
A C# implementation of the Supabase Storage client

This package version is deprecated. Use [Supabase.Storage](#) instead.




**supabase-core** by Joseph Schultz <joseph@acupofjose.com>, **66.3K** downloads  
Shared core for supabase-csharp

This package version is deprecated. Use [Supabase.Core](#) instead.




**Supabase** by Joseph Schultz <joseph@acupofjose.com>, **75** downloads  
A C# implementation of the Supabase client

 supabase

DOCS

← Back to Main Menu

 C#

v1.0

Introduction

Installing

Initializing

DATABASE

Fetch data

Insert data

Update data

Upsert data

Delete data

Call a Postgres function

Using filters

Using modifiers

AUTH

Create a new user

Listen to auth events

Sign in a user

Sign in a user through OTP

Sign in a user through OAuth

Sign out a user

Verify and log in through OTP

Retrieve a session

Retrieve a user

Update a user


EDGE FUNCTIONS

Invokes a Supabase Edge Function.

REALTIME


Subscribe to channel


Q Search docs...

 K

Supabase.com

Dashboard





Initializing

Initializing a new client is pretty straightforward. Find your project url and public key from the admin panel and pass it into your client initialization function.

Supabase

 is heavily dependent on Models deriving from `BaseModel`. To interact with the API, one must have the associated model (see example) specified.

Leverage `Table`, `PrimaryKey`, and `Column` attributes to specify names of classes/properties that are different from their C# Versions.

Standard

Dependency Injection (Maui-like)

With Models Example

```
var url = Environment.GetEnvironmentVariable("SUPABASE_URL");
var key = Environment.GetEnvironmentVariable("SUPABASE_KEY");

var options = new Supabase.SupabaseOptions
{
    AutoConnectRealtime = true
};

var supabase = new Supabase.Client(url, key, options);
await supabase.InitializeAsync();
```

Fetch data

Performing vertical filtering with SELECT.

Getting your data

Selecting specific columns

Query f

Querying with count option

Querying JSON data

— LINQ expressions do not currently support parsing embedded resource columns. For these cases, `string` will need to be used.

General Availability Week

Learn more

Dismiss

# Supabase

## Self-Hosting

Host Supabase on your own infrastructure.

There are several ways to host Supabase on your own computer, server, or cloud.

### Officially supported

#### Docker

Deploy Supabase within your own infrastructure using Docker Compose.

#### BYO Cloud

Contact our Enterprise sales team if you need Supabase managed in your own cloud.

Supabase is also a hosted platform. If you want to get started for free, visit [supabase.com/dashboard](https://supabase.com/dashboard).

### Community supported

There are several community-driven projects to help you deploy Supabase. We encourage you to try them out and contribute back to the community.

#### Kubernetes

Helm charts to deploy a Supabase on Kubernetes.

#### Terraform

A community-driven Terraform Provider for Supabase.

#### Traefik

A self-hosted Supabase setup with Traefik as a reverse proxy.

#### AWS

A CloudFormation template for Supabase.









# Supabase

FREE \$0

PRO \$25

TEAM \$599

 Realtime			
Postgres Changes			
Concurrent Peak Connections 	200 included	500 included then \$10 per 1000	500 included then \$10 per 1000
Messages Per Month 	2 Million included	5 Million included then \$2.50 per Million	5 Million included then \$2.50 per Million
Max Message Size	250 KB	3 MB	3 MB

# 데모 Supabase

# 선택지 3

## Redis Pub/Sub

# 데모 Redis



## 선택지 4

**Message Broker**  
**(Emitter, Kafka, RabbitMQ, MQTT)**

기술	설명	클러스터링	장점	단점
Emitter	실시간 웹 어플리케이션을 위한 메시지 브로커 WebSocket을 사용하여 실시간 메시징에 최적화	쉬운 확장성 및 높은 가용성 제공	보안과 채널 암호화 기능을 포함하여 쉬운 클러스터링 지원	상대적으로 레퍼런스 부족
Kafka	높은 처리량, 확장성을 제공하는 분산 메시징 시스템 대규모 데이터 처리에 적합	매우 높은 확장성 및 뛰어난 내결함성 클러스터링 관리와 유지보수 어려움	매우 높은 처리량과 확장성 분산 시스템에 효과적	실시간 메시징에 상대적으로 높은 지연 시간과 복잡한 설정과 학습 시간 필요
RabbitMQ	범용 메시지 브로커로 다양한 메시징 패턴 지원 빠른 속도로 작은 메시지 처리에 적합	다중 노드 간 메시지 동기화 지원 비교적 간단한 확장성과 장애 대응 메시지 순서와 일관성 유지 어려움	다양한 프로그래밍 언어 호환 실시간 작업에 적합한 낮은 지연 시간	Kafka에 비해 높은 처리량을 요하는 서비스에는 적합하지 않음 고성능 확장성이 제한적

# 앱 화면 구성

# 앱 화면 구성



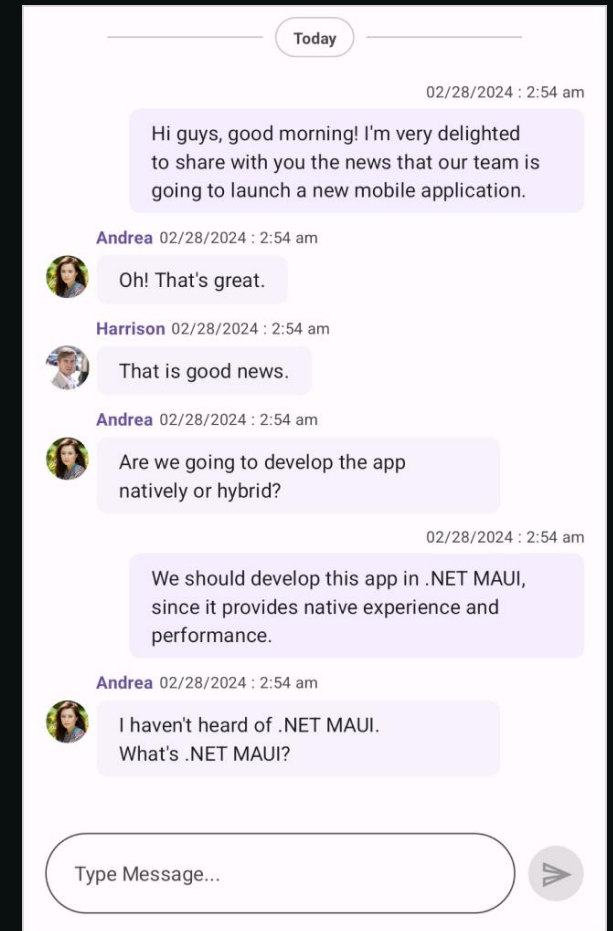
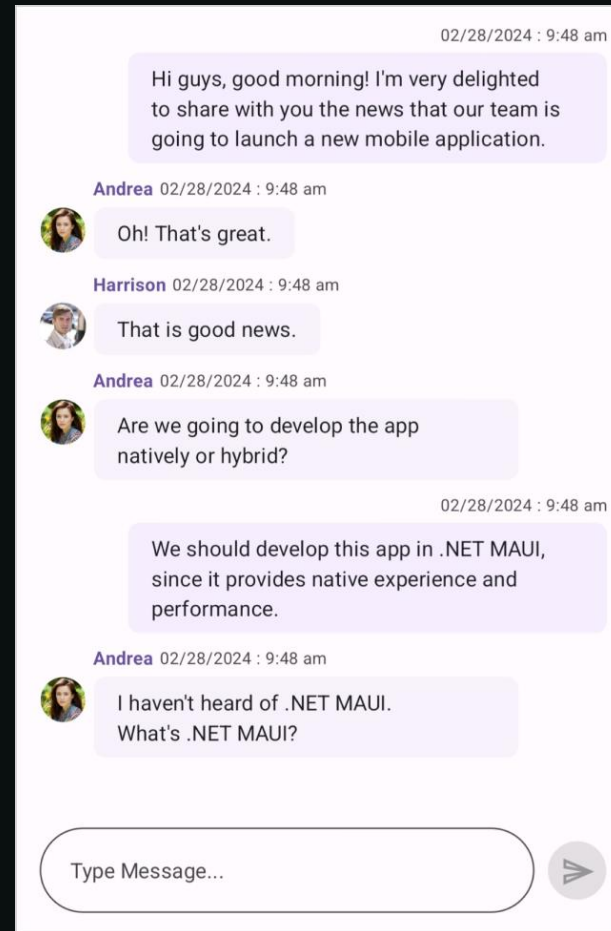
## .NET MAUI Chat: A fully customizable conversational UI

- A modern conversational UI to facilitate communication between users.
- Supports various message types, and essential features like load more, typing indicator, time-break mode, and suggestions.

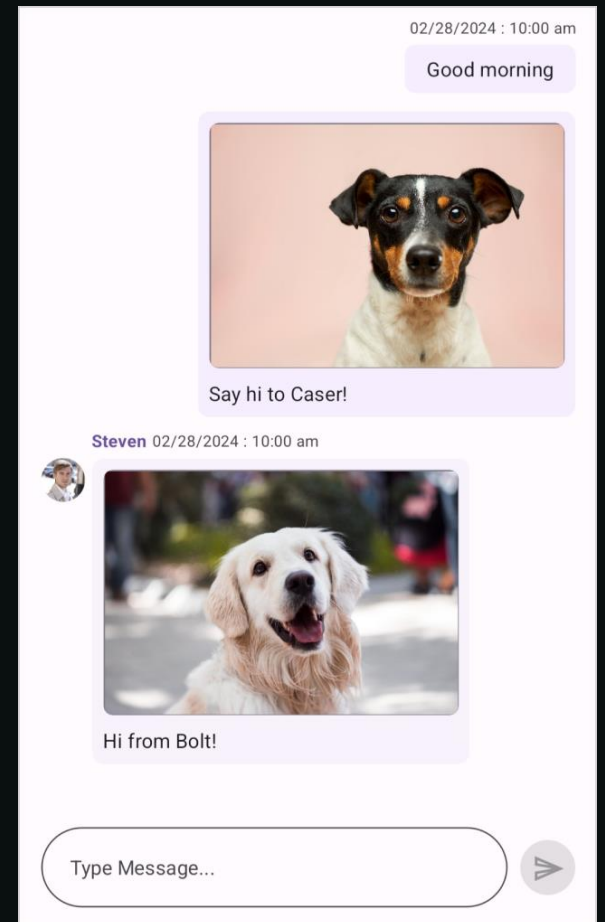
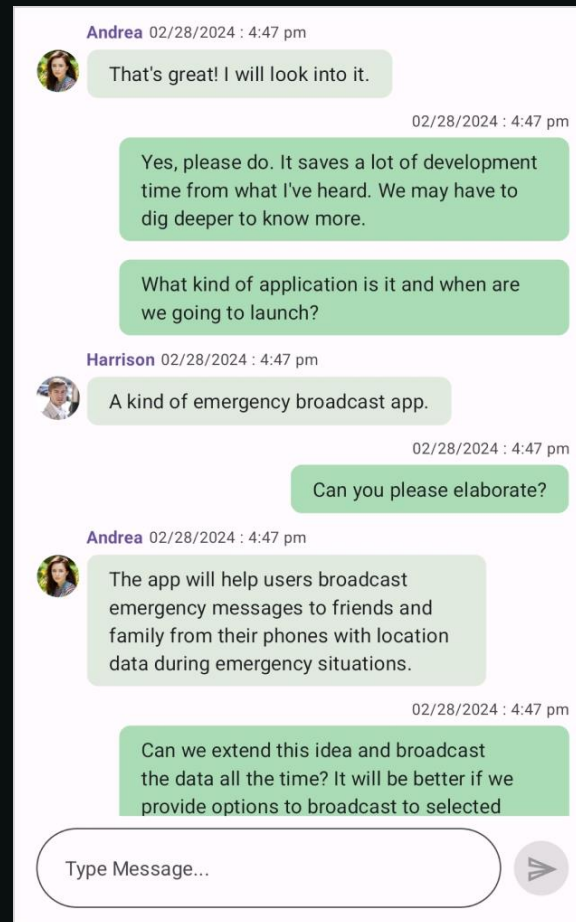
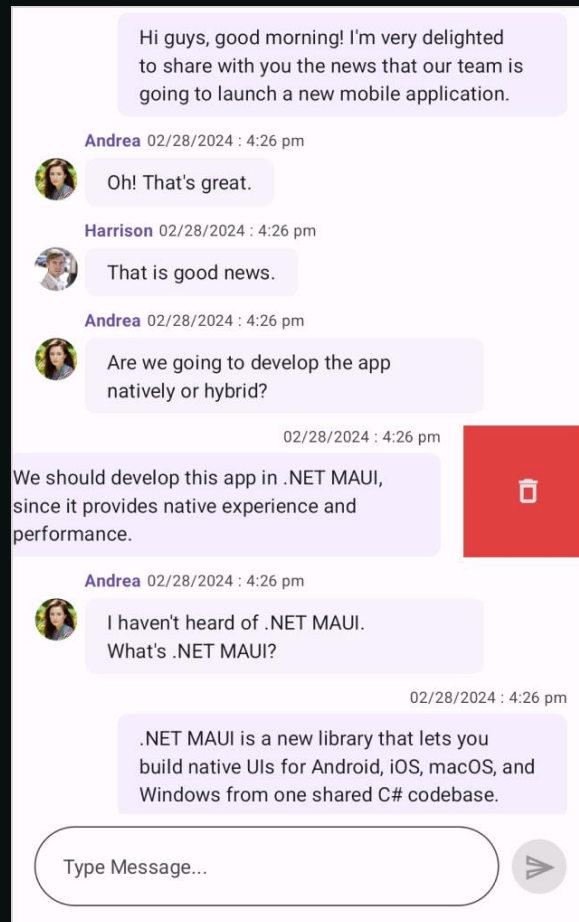
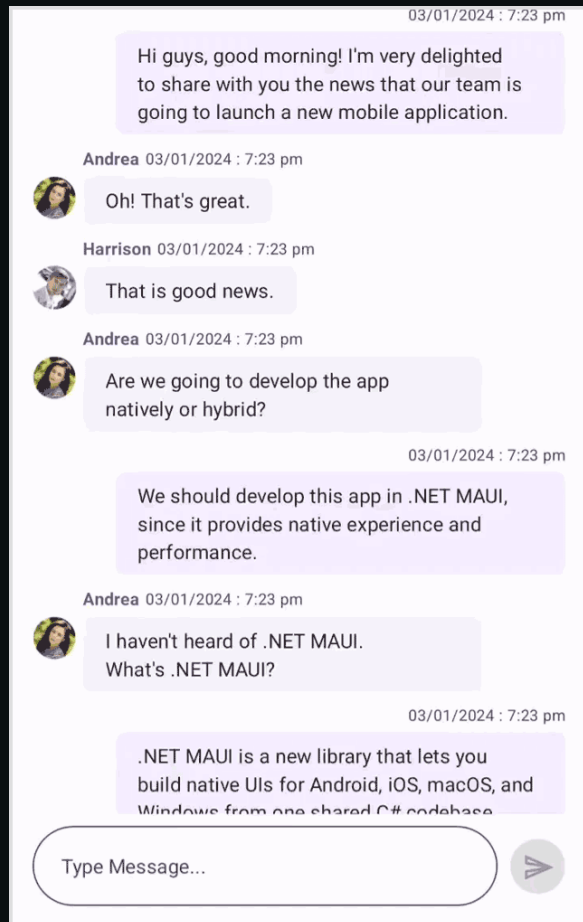
**FREE TRIAL**

**VIEW DEMOS**

*No credit card required.*



# 앱 화면 구성



**질의응답 및 마무리**

# 만들다가 어려우시다면!

.NET MAUI 단톡방

<https://open.kakao.com/o/g8lzoOm>



닷넷데브

<https://forum.dotnetdev.kr>



유튜브

흑우마스터의 야매코딩

