Supabase realtime | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/websockets/supabase-realtime/intro-to-supabase-realtime.html> |
| Date de scraping | 2025-04-08T15:56:17.274511+00:00 |

## Intro to Supabase realtime ​

## What is realtime? ​

Realtime is a feature that allows your app to show live updates instantly, without making incessant API calls.  
For example, in a chat window with lots of messages, it wouldn't be efficient to keep checking for new updates through traditional API requests every few seconds.  
✅ Common use cases for Realtime: chat applications, collaborative tools, and live notifications.  
❌ When not to use Realtime: low interaction applications, high data volume, and complex transactional systems.

## Supabase realtime features ​

Supabase provides three categories of realtime functionalities:  
Broadcast : Send ephemeral messages from client to clients with low latency. For example, to track users' cursors in a collaborative tool. Presence : Track and synchronize shared state between clients. For example, to show how many users are currently online. Postgres Changes : Listen to Postgres database changes and send them to authorized clients. For example, to notify users of a new comment in one of their projects.  
You can learn more about these concepts in Supabase's user docs.

## How realtime works in theory? ​

Before you dive into setting up realtime in a WeWeb project, it can be helpful to get a little bit of an overview of how realtime works.  
The general idea is the following:  
A realtime channel allows a client (the user's browser) and server (Supabase) to send and receive realtime events. Those events can be ephemereal (Broadcast) or permanent (Database changes). If users want to receive or send events in a channel, they first need to subscribe to that channel (via a WeWeb workflow action). Once they've subscribed to a channel, users can broadcast messages to that channel (via a WeWeb workflow action). They can also listen to events sent by other users in that channel (via a WeWeb workflow triggered on page or app load).

## Support chat example ​

Taking the example of a support chat, here's how you could build it in WeWeb.  
When a user clicks on the support icon of your website, you would trigger:  
the Subscribe to channel action so they join the chat channel. the Presence state action so they can see how many support team members are online. the Broadcast a message action to send a message in the chat.  
You would also have an app workflow that:  
listens for messages from other users in the chat channel. updates a variable with those messages.  
You'd then use that variable to display those messages on the page.  
That's the theory.  
Learn more about how Supabase realtime works in practice:  
Subscribe to and unsubscribe from a channel Send and receive messages Send and receive presence information