Subscribe & Unsubscribe to Supabase channels | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/websockets/supabase-realtime/subscribe-channel.html> |
| Date de scraping | 2025-04-08T15:56:20.023506+00:00 |

## Subscribe & unsubscribe to a Supabase channel ​

## Why subscribing matters ​

If you want users of your app to be able to listen and react to events in a realtime channel, you first need to make them subscribe to the channel.

## Use cases ​

A few use cases:  
When user clicks on help icon , they subscribe to the chat channel where instant messages come and go. When user signs up to the app, they subscribe to the releases channel where new events are created every time a new product update is inserted in the database. When user opens a tab , they subscribe to the cursor channel where new events are created with their mouse position every time their mouse moves.

## Two types of channels ​

There are two types of Supabase realtime channels users can subscribe to:  
Broadcast : Send ephemeral messages from client to clients with low latency. For example, to track users' cursors in a collaborative tool. 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.

## Subscribe to broadcast ​

There are use cases where you don't need to store realtime events in a database. For example, live collaboration tools or online games that track every active user's cursor position.  
For users to send and receive realtime cursor positions, they first need to subscribe to a realtime Broadcast channel:  
  
The Channel name can be anything you want. Here we named it canvas . It's the name we will need to reference later when we want to send and receive messages in this channel.  
The Type is Broadcast because we are working with ephemereal events that don't need to be stored in our database.  
We chose to subscribe to only the cursor events because we only want to listen to this specific event type but we could leave the input empty or type in \* to subscribe to all the events that go through this channel. What you choose will depend on your use case.  
The Listen self and Listen presence can be toggled on or off depending on your preferences and use case.  
Learn more about these options here in the Supabase user docs.

### Images

<https://docs.weweb.io/assets/supabase-realtime-subscribe-broadcast.DezFluGO.png>

## Next steps ​

Once we have subscribed to a Broadcast channel, we can send, receive and react to realtime events from that channel .

## Subscribe to db changes ​

In many cases, you want users to be able to receive and react to realtime changes in a database. For example, when someone at your company adds a record in the releases table of your database, you want to let users know there's a new product update.  
For users to send and receive realtime database events, they first need to subscribe to a Database changes channel:  
  
In the example above, you can see the user is subscribing to:  
All events in the releases table that can be found in the public schema of my Supabase db  
The Channel name can be anything you want. Here we named it releases . It's the name we will need to reference later when we want to send and receive events in this channel.  
The Listen presence can be toggled on or off depending on your preferences and use case.

### Images

<https://docs.weweb.io/assets/supabase-realtime-enable-table.1MRW6cwi.png>

<https://docs.weweb.io/assets/supabase-realtime-subscribe-db.DPdb7Wdz.png>

## Next steps ​

Once we have subscribed to a Database changes channel, we can receive and react to realtime events from that channel .