Chat Room

## **Tech stack**

* Ruby on Rails v6.0.0 -> Web framework
* PostgresSQL 9.6 -> Database
* Redis 5.0.5 -> Caching
* Sidekiq 5.2 -> Message broker and background jobs
* ActionCable -> Enabling web sockets

## **The Application**

I have used the above tech stacks to complete the given task. When searched for few examples online I found out that Ruby on Rails web framework needed less effort to get started. So i started with Ruby on Rails for this task. One of the awesome features that comes with Rails is ActionCable. With ActionCable, we can build all the real-time features you can think of via websocket. So I used ActionCable to enable web sockets.

The idea is a chat room that has multiple messages, each message has a content and a sender. Note that a message doesn’t have a “**receiver**”. This allows a room to have *any number of users* since you don’t need to care about the receiver of the messages, since all the messages from the senders will end up appearing in a room regardless of how many participants in the room. So, this is a data structure that I use:

* **Conversation (room)**: *has\_many* messages, users and has an id
* **Message**: *belongs\_to* a conversation, has a sender, has the text content
* **Sender**: is a User

To store all the messages in a chanel I used PostgressSQL as the database and used it to pull the common keywords for every message and implemented a job by database triggers and publish a customized dashboard for every user to rank the keywords for the last 10 minutes on the web page.

While the development i had issues with the connection of the user to a channel and it took longer for the message to be sent. The next issue I had was when messages from a user were sent simultaneously the system took time to send them.

FCM was used to make a push notification every time a user is not available for answering than 30 secs. Auto Reply when users are not online are also done using a background job. Unseen message notification in a channel is displayed when the user in texting in another channel. Here I had used sidekiq for message broking and for other background jobs. A prank button is presented to send prank messages of 10rps in a channel.

## **References**

<https://iridakos.com/tutorials/2019/04/04/creating-chat-application-rails-websockets.html>

<https://www.youtube.com/watch?v=HEGeoELGTnU>

<https://iridakos.com/tutorials/2019/04/07/dockerizing-a-rails-application.html>