



<https://1.101.fr>

Follow the instruction at

<https://github.com/huguesbr/ufr-blois-development-web-messenger.git>

Our Chat now have users, chats and messages.

The current user's is faked by passing a ***?current\_user\_id=123*** where ***123*** is the id of the desired current user.

Action	Users	Chats	Messages
list	anybody	any user	any user
create	anybody	any user	any user
update	current user	chat's owner	message's owner
delete	current user	chat's owner	message's owner
show	anybody	any user	any user

At the end of this TD we want limit the access to a chat's message via a membership

Action	Users	Chats	Messages	Membership
list	anybody	any user	user with an accepted membership	any user
create	anybody	any user	user with an accepted membership	any user
update	current user	chat's owner	message's owner	chat's owner
delete	current user	chat's owner	message's owner	chat's owner
show	anybody	any user	user with an accepted membership	any user

# Force current user to be the creator of a chat / message

In our previous implementation, any user could create a chat / message pretending to be someone else

We want to force the chat/message creator to be the current user

```
# app/controllers/chats_controller.rb  
def chat_params  
  chat_params = params.require(:chat).permit(:name)  
  # force the chat creator to be the current user  
  chat_params.merge(user_id: current_user_id)  
end
```

same for ***messages\_controller.rb***

# Membership

A membership table will hold a membership (being a member) of a chat and a user

It will (for now) be populated manually (without any restriction)  
To avoid creating multiple membership for the same user you can

```
@membership = Membership.find_or_initialize_by(membership_create_params)  
# vs  
@membership = Membership.new(membership_create_params)
```

- create a membership table

***bin/rails generate scaffold Membership chat:references user:references***  
***bin/rake db:migrate***

# Create membership for chat creator

We want a chat creator to have an implicit membership for the chat it created  
***after\_create*** allow to trigger an action after an active record creation

– [https://apidock.com/rails/ActiveRecord/Callbacks/after\\_create](https://apidock.com/rails/ActiveRecord/Callbacks/after_create)

```
# app/models/chat.rb  
class Chat < ApplicationRecord  
  belongs_to :user  
  has_many :messages, dependent: :destroy  
  
  # ensure chat's creator have a membership  
  after_create do |chat|  
    Membership.create(user: chat.user, chat: chat)  
  end  
end
```



# Update our seed

Our ***seed.rb*** create some messages, we want to be sure to also create a chat's membership for this users

We don't need to create membership for the chat owner, as the ***after\_create*** on ***chat.rb*** will take care of it

```
# db/seeds.rb
Chat.all.each do |chat|
  Membership.create(chat_id: chat.id, user_id: chat.user_id)
end
```

- create memberships for chat's messages

*# bin/rails console*

*Message.all.each do |message|*

*Membership.find\_or\_create\_by(chat\_id: message.chat\_id, user\_id: message.user\_id)*  
*end*

We also want to add this to ***seed.rb*** to properly seed the db when  
using ***bin/rake db:seed***

If we create the membership for a chat owner on chat's creation we  
don't need to do it in ***seed.rb***

# Enforce Membership

we've create a membership table, we now want to use it to restrict chat access to user with a membership  
at this step anybody can still create/update memberships  
trigger an action before processing a request ***before\_action :verify\_membership***  
find a membership ***membership = Membership.find\_by(chat\_id: 123, user\_id: 456)***  
verify a membership entry exists ***Membership.exists?(chat\_id: 123, user\_id: 456)*** -> true/false

Action	Users	Chats	Messages	Membership
list	anybody	any user	user with a membership	anybody
create	anybody	any user	user with a membership	anybody
update	current user	chat's owner	message's owner	anybody
delete	current user	chat's owner	message's owner	anybody
show	anybody	any user	user with a membership	anybody

# Add a Chat membership status

add a field to a table ***bin/rails generate migration AddFieldnameToTablename fieldname:type*** (type: ***string***)  
to set a default value in a migration  
***add\_column :memberships, :status, :string, default: 'pending'***  
migrate the table ***bin/rake db:migrate***

Action	Users	Chats	Messages	Membership
list	anybody	any user	user with an accepted membership	any user
create	anybody	any user	user with an accepted membership	any user
update	current user	chat's owner	message's owner	chat's owner
delete	current user	chat's owner	message's owner	chat's owner
show	anybody	any user	user with an accepted membership	any user

- add a status field (string) to membership
- a new membership should be created as ***pending***
- allow the owner of a chat to update a membership status (-> ***accepted***)
- only allow user with an accepted membership to access chat's message

don't forget to update your ***seed.rb***