



<https://1.101.fr>

Follow the instruction at

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

Seed

Seeding allow to some data to a database

bin/rake db:seed

Similarly you can create some data from the console

bin/rails console

Membership.create(chat_id: 1, user_id: 1)

Current User

The App DOES NOT have a login for now, then for does NOT have a notion of a current user.

In this TD, we will pretend to have a current user by passing a ***user_id=1*** query params2
ie.: /chats?user_id=1

- In order to have an easy access to the current user from anywhere let's add it to the ***ApplicationController***

```
# app/controllers/application_controller.rb
class ApplicationController < ActionController::API
  def current_user_id
    params[:user_id]
  end
end
```

Unauthorized

To return a 401 ***render nothing: true, status: 401***

Example, to return an (inconditional) 401 when someone try to fetch all users ***GET / users***

```
# app/controllers/user_controller.rb
class UsersController < ApplicationController
  before_action :set_user, only: [:show, :update, :destroy]

  # GET /users
  def index
    render nothing: true, status: 401
  end

  # ...
end
```

Limit user's actions

- anyone can create a new user
 - ***POST /users -> 201***
- any users can list all users
 - ***GET /users -> 200***
- any users can fetch a specific users
 - ***GET /users/1 -> 200***
- a user can only update itself
 - ***PUT /users/1?user_id=1 -> 200***
 - ***PUT /users/1?user_id=2 -> 401***
- a user can only delete itself
 - ***DELETE /users/1?user_id=1 -> 204***
 - ***DELETE /users/1?user_id=2 -> 401***

Limit chat's actions

- any users can create a new chat (but we need a user)
 - ***POST /chats?user_id=1 -> 201***
 - ***POST /chats -> 401***
- any users can list all chats (but we need a user)
 - ***GET /chats?user_id=1 -> 200***
 - ***GET /chats -> 401***
- any users can fetch a specific chat (but we need a user)
 - ***GET /chats/1?user_id=1 -> 200***
 - ***GET /chats/1 -> 401***
- assuming chat 1 has been created by user 2
 - only the owner (***chat.user_id***) can update the chat
 - ***PUT /chats/1?user_id=2 -> 204***
 - ***PUT /users/1?user_id=1 -> 401***
 - only the owner (***chat.user_id***) can delete the chat
 - ***DELETE /chats/1?user_id=2 -> 204***
 - ***DELETE /users/1?user_id=1 -> 401***

Limit message's actions

In order to limit message actions, we need to introduce a new class ***Membership***. This class will hold the chat's membership, but will for now be populated manually (without any restriction)

- create a membership table

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

- create some memberships (via the console)

```
bin/rails console  
Membership.create(chat_id: 1, user_id: 1)
```


Enforce Membership

find a membership *membership = Membership.find_by(chat_id: 123, user_id: 456)*

- any users can create a membership
 - ***POST /chats?user_id=1 -> 201***
 - ***POST /chats -> 401***
- assuming user 1 is the owner of a chat 1
- assuming user 2 is a member of a chat 1
- assuming user 3 is NOT a member of a chat 1
 - only members (or owner) of a chat can fetch the chat's message
 - ***GET /chats/1/messages?user_id=1 -> 201***
 - ***GET /chats/1/messages?user_id=2 -> 201***
 - ***GET /chats/1/messages?user_id=3 -> 401***
- assuming user 2 is the creator of message 3
- only the message's user can update a message
 - ***UPDATE /chats/1/messages/3?user_id=2 -> 201***
 - ***UPDATE /chats/1/messages/3?user_id=1 -> 401***
- only the message's user can delete a message
 - ***DELETE /chats/1/messages/3?user_id=2 -> 204***
 - ***DELETE /chats/1/messages/3?user_id=1 -> 401***

Add a Chat membership status

add a status field (string) to membership

- add a field to a table ***bin/rails generate migration AddFieldnameToTablename fieldname:type*** (type: ***string***)
- migrate the table ***bin/rake db:migrate***

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