



**KAOUTAR
JABRI**

Compte rendu du projet messages



BTS 2021-2023

Sommaire

- 01.** Introduction
- 02.** Le résultat attendu du projet
- 03.** Présentation et explication
du code
- 04.** Conclusion

Introduction

Ce projet est développé en php permet d'envoyer des messages dans un groupe de chat.

Résultat attendu

Le but du projet est de créer un chatroom qui permet à l'utilisateur de mettre son username et d'écrire son message.

Presentation et explication du code

Premièrement on trouve les modules, on trouve le thread manager qui fait pour faire la distinction entre la personne connecté et celle avec qui elle parle ainsi que le thread qui a une relation avec l'utilisateur et le chat message qui enregistre les messages

```
8
9 class ThreadManager(models.Manager):
10     def by_user(self, **kwargs):
11         user = kwargs.get('user')
12         lookup = Q(first_person=user) | Q(second_person=user)
13         qs = self.get_queryset().filter(lookup).distinct()
14         return qs
15
16
17 class Thread(models.Model):
18     first_person = models.ForeignKey(User, on_delete=models.CASCADE, null=True, blank=True, related_name='first_thread')
19     second_person = models.ForeignKey(User, on_delete=models.CASCADE, null=True, blank=True, related_name='second_thread')
20     updated = models.DateTimeField(auto_now=True)
21     timestamp = models.DateTimeField(auto_now_add=True)
22
23     objects = ThreadManager()
24     class Meta:
25         unique_together = ('first_person', 'second_person')
26
27
28 class ChatMessage(models.Model):
29     thread = models.ForeignKey(Thread, null=True, blank=True, on_delete=models.CASCADE, related_name='messages')
30     user = models.ForeignKey(User, on_delete=models.CASCADE)
31     message = models.TextField()
32     timestamp = models.DateTimeField(auto_now_add=True)
```

on trouve ici la partie ou on récupère les messages ainsi que l'utilisateur (celui qui est connecté) le websocket donne des résultats instantanées.

```
73
74
75     async def websocket_disconnect(self, event):
76         print('disconnect', event)
77
78     async def chat_message(self, event):
79         print('chat_message', event)
80         await self.send({
81             'type': 'websocket.send',
82             'text': event['text']
83         })
84
85     @database_sync_to_async
86     def get_user_object(self, user_id):
87         qs = User.objects.filter(id=user_id)
88         if qs.exists():
89             obj = qs.first()
90         else:
91             obj = None
92         return obj
93
94     @database_sync_to_async
95     def get_thread(self, thread_id):
96         qs = Thread.objects.filter(id=thread_id)
97         if qs.exists():
98             obj = qs.first()
99         else:
100             obj = None
101         return obj
```

Dans la partie Template l'affichage des information est fait par rapport à l'utilisateur connecté par une condition qui vérifie que le premier utilisateur est bien celui connecté

```
40         <span class="online_icon"></span>
41     </div>
42     <div class="user_info">
43         {% if thread.first_person == user %}
44             <span>{{ thread.second_person.username }}</span>
45         {% else %}
46             <span>{{ thread.first_person.username }}</span>
47         {% endif %}
48
49     {# <p>John</p>#}
50     </div>
51 </li>
52 {% endfor %}
53 </ul>
54 </div>
55 <div class="card-footer"></div>
56 </div></div>
57 <div class="col-md-8 col-xl-6 chat h-100 mt-4">
58     <div class="card">
59         {% for thread in Threads %}
60             <div class="messages-wrapper" {% if forloop.first %}hide is_active{%
61                 else %}hide{% endif %}" chat-id="chat-{{ thread.id }}" other-user-id="
62                 {% if thread.first_person == user %}
63                     {{ thread.second_person.id }}
64                 {% else %}
65                     {{ thread.first_person.id }}
66                 {% endif %}
67             ">
```

Conclusion

Pour conclure, le projet final permet d'envoyer des messages instantanément et affiche les informations selon la personne connecté.

