**Html**

**ID:**

聊天室 : room\_{{ room.id }}

User : user\_{{ user.id }}

通知 : notification\_{{ele.id}}

**Views.py**

**Room:**

**[GET]**

**Return content:**

1. userid = request.user.id #本人id
2. All\_Notification = Notification.objects.filter(to\_user\_id=owner)
3. All\_User = User.objects.exclude(id=userid)
4. All\_Involved\_Group = Group.objects.filter(user\_\_in=[userid])

* 2~4 確認登入後再query

**[POST]**

**先確認type (分辨不同post) => *request.POST.get("type","")***

1. **Type = "Clean\_Seen"**

//改變通知已讀狀態

seen\_id = request.POST.getlist('seen\_id[]', '')

#將ajax傳回data中取出欲改變狀態的list

Notification\_Select\_List = Notification.objects.filter(id\_\_in=seen\_id)

1. **Type = “Create\_Group\_Single”**

to\_user = request.POST.get("user","")

#發送對象id

Redundant\_Group = Group.objects.filter(type="single", user\_\_in = [request.user.id]).filter(user\_\_in = [to\_user])

#篩選出是否有存在的聊天室，type=single時聊天室成員僅會有2個，使用兩次filter可以找到正確的group

※[field]\_\_in 會找只要有包含list中至少一項的group，因此filter使用兩次方能找到正確的group

確認若無已存在的group則創立

Create\_Group = Group.objects.create(display\_name=request.user.username+"\_with\_"+User.objects.get(id = to\_user).username , type="single")  
Create\_Group.save()  
Create\_Group.user.add(request.user)  
Create\_Group.user.add(User.objects.get(id = to\_user))

並回傳group 的id和display\_name