|  |
| --- |
| import { Component, OnInit, Input } from '@angular/core'; |
|  | import { LoginService } from '../Service/LoginService'; |
|  | import { UserLogin } from '../Model/UserLogin'; |
|  | import { ActivatedRoute, Router } from '@angular/router'; |
|  | import { HubConnectionService } from '../Service/HubConnectionService'; |
|  | import { MessageService } from '../Service/MessageService'; |
|  |  |
|  | @Component({ |
|  | selector: 'app-chat-list', |
|  | templateUrl: './chat-list.component.html', |
|  | styleUrls: ['./chat-list.component.css'] |
|  | }) |
|  | export class ChatListComponent implements OnInit { |
|  | notifysenders: string[] = []; |
|  | interval: any; |
|  | public users: Array<UserLogin> = []; |
|  | sender: string |
|  | senderupdate = new UserLogin(); |
|  | unread: number = 1; |
|  | countmsg: number = 1; |
|  | recevier: string; |
|  | names: string; |
|  | constructor(private msgservice: MessageService, private \_dataservice: LoginService, private router: Router, private route: ActivatedRoute, private hubservice: HubConnectionService) { } |
|  |  |
|  | mapcount() { |
|  | var map = this.notifysenders.reduce(function (prev, cur) { |
|  | prev[cur] = (prev[cur] || 0) + 1; |
|  | return prev; |
|  | }, {}); |
|  |  |
|  | // console.log(map); |
|  |  |
|  |  |
|  | this.\_dataservice.getUsers() |
|  | .subscribe((data: any) => { |
|  |  |
|  | this.users = data; |
|  |  |
|  | for (let key in map) { |
|  | for (let da of this.users) { |
|  | if (da.name == key) |
|  | da.countmsg = map[key]; |
|  | } |
|  | } |
|  | }); |
|  | } |
|  |  |
|  | ngOnInit() { |
|  | let name = this.route.snapshot.params['name']; |
|  | this.\_dataservice.setsender(name); |
|  | this.sender = this.\_dataservice.getsender(); |
|  |  |
|  | this.\_dataservice.getUsers() |
|  | .subscribe((data: any) => this.users = data); |
|  |  |
|  |  |
|  | this.interval = setInterval(() => { |
|  | this.notifysenders = []; |
|  | this.recevier = this.\_dataservice.getrecevier(); |
|  |  |
|  | this.\_dataservice.getUsers() |
|  | .subscribe((data: any) => this.users = data); |
|  | this.msgservice.getMsgs().subscribe((data: any) => { |
|  | // console.log(data); |
|  | for (let msg of data) { |
|  | if (msg.recevier === this.sender && msg.isRead == false) { |
|  | this.notifysenders.push(msg.sender); |
|  | } |
|  | } |
|  | // console.log(this.notifysenders) |
|  | this.mapcount(); |
|  | }); |
|  | }, 10000); |
|  | } |
|  | } |