**Step -1**

Create new Angular App

And add following dependencies

@angular/cdk

@angular/http

@angular/material

core-js

rxjs-compat

**Step -2**

App.component.html

<div class="container">

<div class="header">

<mat-form-field>

<input matInput placeholder="Your name here" type="text" [(ngModel)]="sender" [ngModelOptions]="{updateOn: 'blur'}">

</mat-form-field>

</div>

<div \*ngIf="!!sender" #viewer class="body">

<div class="viewer">

<div class="message" \*ngFor="let msg of serverMessages" [ngClass]="{'bold': msg.isBroadcast, 'others': !isMine(msg)}">

<div class="badge" [style.backgroundColor]="getSenderColor(msg.sender)">{{ getSenderInitials(msg.sender) }}</div>

<div class="content" [ngClass]="{'bold': msg.isBroadcast, 'others': !isMine(msg)}">

<i \*ngIf="msg.isBroadcast" class="material-icons">speaker\_phone</i>

<span>{{ msg.content }}</span>

</div>

</div>

</div>

</div>

<div \*ngIf="!!sender" class="footer">

<form (submit)="send()">

<mat-form-field>

<input name="clientMessage" matInput placeholder="Write a message" type="text" [(ngModel)]="clientMessage">

</mat-form-field>

<button mat-fab type="submit">

<i class="material-icons">send</i>

</button>

<button mat-fab (click)="toggleIsBroadcast()" type="button" [ngClass]="{'disabled': !isBroadcast}">

<i class="material-icons">speaker\_phone</i>

</button>

</form>

</div>

</div>

app.component.ts

import { Component, ViewChild, ElementRef, OnInit, AfterViewInit } from '@angular/core';

import { WebSocketSubject } from 'rxjs/observable/dom/WebSocketSubject';

export class Message {

constructor(

public sender: string,

public content: string,

public isBroadcast = false,

) { }

}

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.scss']

})

export class AppComponent implements AfterViewInit {

@ViewChild('viewer', null) private viewer: ElementRef;

public serverMessages = new Array<Message>();

public clientMessage = '';

public isBroadcast = false;

public sender = '';

private socket$: WebSocketSubject<Message>;

constructor() {

this.socket$ = new WebSocketSubject('ws://localhost:8999');

this.socket$

.subscribe(

(message) => this.serverMessages.push(message) && this.scroll(),

(err) => console.error(err),

() => console.warn('Completed!')

);

}

ngAfterViewInit(): void {

this.scroll();

}

public toggleIsBroadcast(): void {

this.isBroadcast = !this.isBroadcast;

}

public send(): void {

const message = new Message(this.sender, this.clientMessage, this.isBroadcast);

this.serverMessages.push(message);

this.socket$.next(message);

this.clientMessage = '';

this.scroll();

}

public isMine(message: Message): boolean {

return message && message.sender === this.sender;

}

public getSenderInitials(sender: string): string {

return sender && sender.substring(0, 2).toLocaleUpperCase();

}

public getSenderColor(sender: string): string {

const alpha = '0123456789ABCDEFGHIJKLMNOPQRSTUVXYZ';

const initials = this.getSenderInitials(sender);

const value = Math.ceil((alpha.indexOf(initials[0]) + alpha.indexOf(initials[1])) \* 255 \* 255 \* 255 / 70);

return '#' + value.toString(16).padEnd(6, '0');

}

private scroll(): void {

setTimeout(() => {

this.scrollToBottom();

}, 100);

}

private getDiff(): number {

if (!this.viewer) {

return -1;

}

const nativeElement = this.viewer.nativeElement;

return nativeElement.scrollHeight - (nativeElement.scrollTop + nativeElement.clientHeight);

}

private scrollToBottom(t = 1, b = 0): void {

if (b < 1) {

b = this.getDiff();

}

if (b > 0 && t <= 120) {

setTimeout(() => {

const diff = this.easeInOutSin(t / 120) \* this.getDiff();

this.viewer.nativeElement.scrollTop += diff;

this.scrollToBottom(++t, b);

}, 1 / 60);

}

}

private easeInOutSin(t): number {

return (1 + Math.sin(Math.PI \* t - Math.PI / 2)) / 2;

}

}

App.module.ts

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { MatButtonModule, MatCheckboxModule, MatInputModule, MatButtonToggleModule, MatIconModule } from '@angular/material';

import { FormsModule } from '@angular/forms';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

BrowserAnimationsModule,

FormsModule,

MatButtonModule,

MatButtonToggleModule,

MatIconModule,

MatInputModule,

MatCheckboxModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }