

Progetto pratico

Simone Giamboni Daniel Ibrahim I3A

Semestre primaverile 2023/2024

Applicazioni web 2

Angular

- Framework open source sviluppato da Google
- Linguaggio: Typescript
- Applicazioni web a Single Page Application
- Data binding bidirezionale
- Creazione di molteplici componenti con relazione padre-figlio



Data binding bidirezionale

- Modifiche del modello si riflettono immediatamente sulla view e viceversa

Model

```
onEdit: boolean = false;  
editedMessage: string = "";
```

View

```
<div *ngIf="onEdit">  
  <input type="text" class="form-control" [(ngModel)]="editedMessage">  
</div>
```

Componenti

- Unità di interfaccia utente riutilizzabile
- 3 file: template HTML, file di stile CSS, file di codice in Typescript

```
@Component({
  selector: 'app-search-message',
  standalone: true,
  imports: [CommonModule, FormsModule],
  templateUrl: './search-message.component.html',
  styleUrls: ['./search-message.component.css']
})
export class SearchMessageComponent implements OnChanges {
  searchInput: string = "";
  @Output() searchMessagesEvent = new EventEmitter<string>();
  @Input() channel: Channel;
```

▼ messages-list

messages-list.component.css

<> messages-list.component.html

TS messages-list.component.spec.ts

TS messages-list.component.ts

```
<div id="searchMessageContainer" class="m-3">
  <div class="row">
    <div class="col">
      <input type="text" class="form-control" placeholder="Search."
        #inputSearch>
    </div>
    <div class="col-2">
      <button class="btn btn-primary" (click)="clearSearchInput()"
    </div>
  </div>
</div>
```

```
#ChannelHeader {
  border-bottom: 3px solid #0d6efd;
}

#messagesList {
  height: 69vh;
  overflow-y: auto;
}
```

Relazione componenti: da padre a figlio

- Input

Componente figlio

```
export class MessageComponent {  
  
  @Input() url: string;  
  @Input() message : Message;  
  @Input() authorRegistered: string;
```

Componente padre (message-list component)

```
<app-message [url]="url" [message] = message [authorRegistered]="authorRegistered"  
  (messageClickedEvent)="onMessageFromListClick($event)">  
</app-message>
```

Relazione componenti: da figlio a padre

- Output

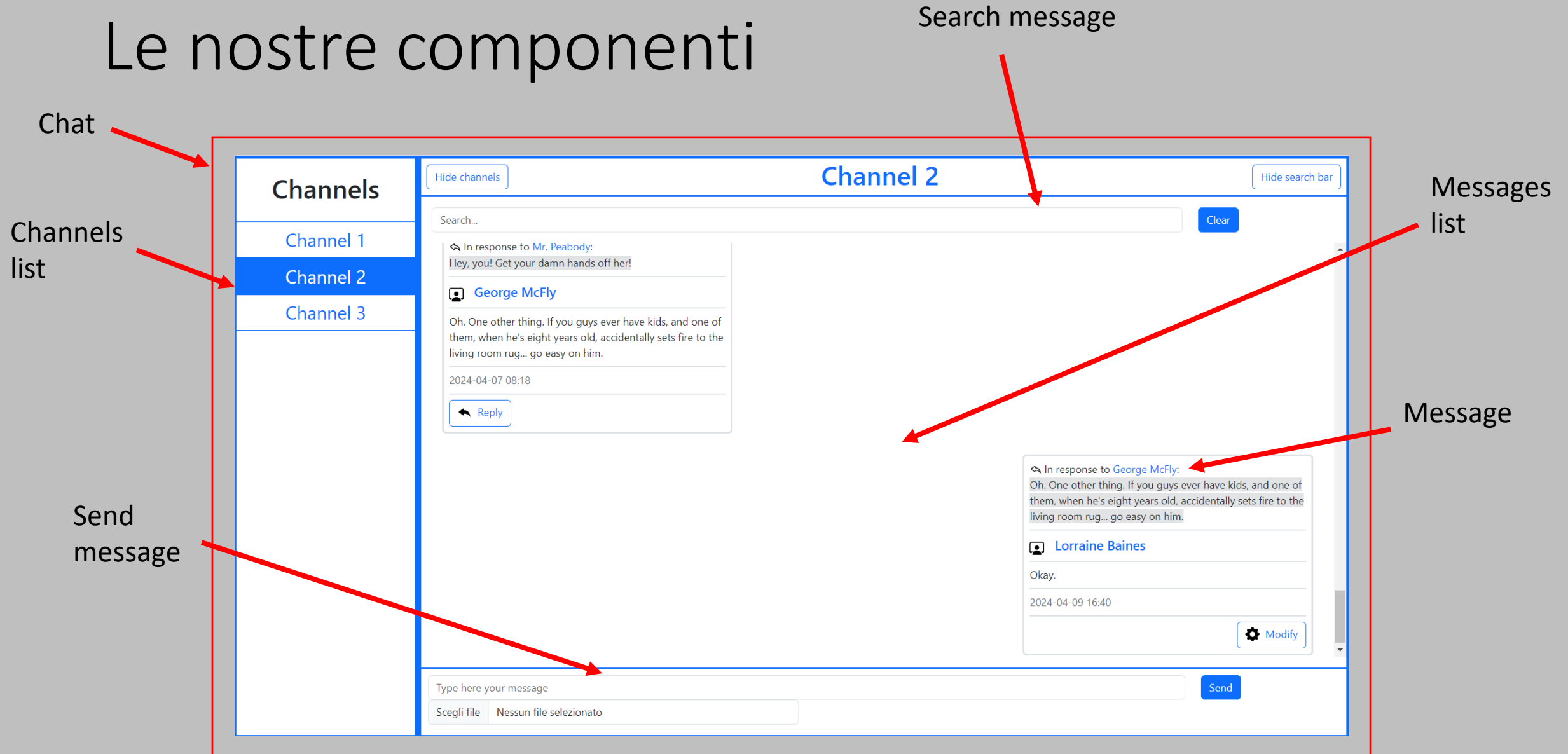
Componente figlio

```
export class MessageComponent {  
  
    @Output() messageClickedEvent = new EventEmitter<Message>();  
  
    onMessageClick(message: Message) {  
        this.messageClickedEvent.emit(message);  
    }  
}
```

Componente padre (messages-list)

```
<app-message [url]="url" [message] = message [authorRegistered]="authorRegistered"  
    (messageClickedEvent)="onMessageFromListClick($event)">  
</app-message>
```

Le nostre componenti



Chat

- Padre di tutte le componenti
- Possibilità di configurazione:
 - l'url delle chiamate http
 - il nome dell'autore
 - la scelta di mostrare/non mostrare la barra di ricerca
 - la scelta di mostrare/non mostrare la lista dei canali

```
<app-chat  
  [url]="'https://supsi-ticket.cloudns.org/supsi-chat/bff/'"  
  [author]="'Lorraine Baines'"  
  [showChannelsList]="true"  
  [showSearchBar]="true"  
  (messageInChatClickedEvent)="onMessageClick($event)">  
</app-chat>
```


Channels list

- Selezione dei vari canali della chat
- Mostra i messaggi non letti di un canale

```
onSelect(channel : Channel) {  
  if(this.selectedChannel != null) {  
    this.selectedChannel.selected = false;  
    this.selectedChannel = channel;  
    this.selectedChannel.selected = true;  
    this.resetNumberOfUnreadMessagesInChannel(this.selectedChannel.id);  
    this.selectChannelEvent.emit(this.selectedChannel);  
  }  
}
```

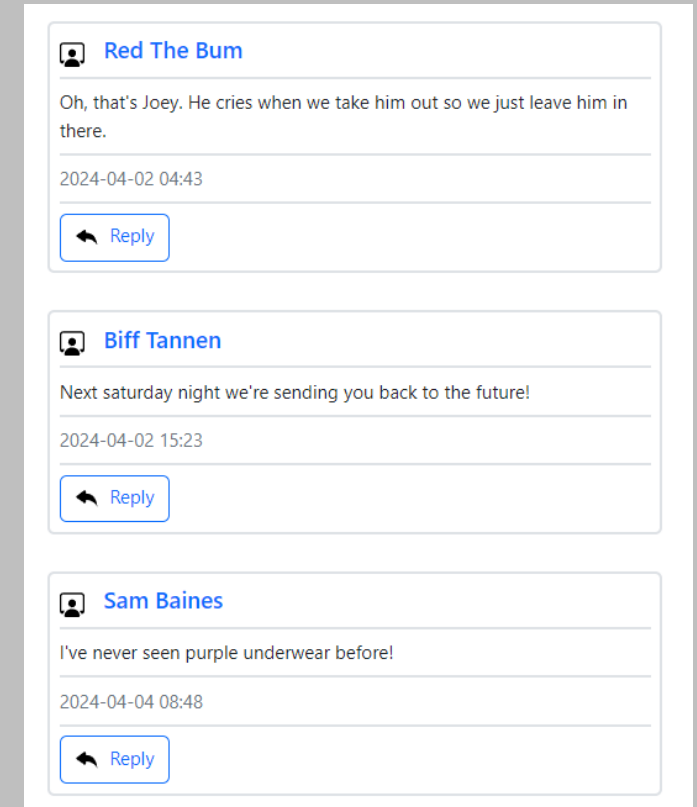
```
<li *ngFor="let channel of channels">  
  <button (click)="onSelect(channel)"  
    class="btn btn-outline-primary rounded-0 border-start-0"  
    [ngClass]="channel.selected ? 'active' : ''">
```

Channels	
Channel 1	
Channel 2	2
Channel 3	1

Messages list

- Recupero dei messaggi del canale selezionato
- Gestisce il layout dei messaggi del canale
- Gestisce lo scorrimento verticale della lista

```
ngOnChanges(changes: SimpleChanges) {  
  if (changes.channel) {  
    if (this.channel) {  
      this.initMessages();  
    }  
  }  
}
```



Send message e Search message

- Invio messaggio con possibile allegato

```
@Input() url: string;  
@Input() authorRegistered: string;  
@Input() channel: Channel;
```

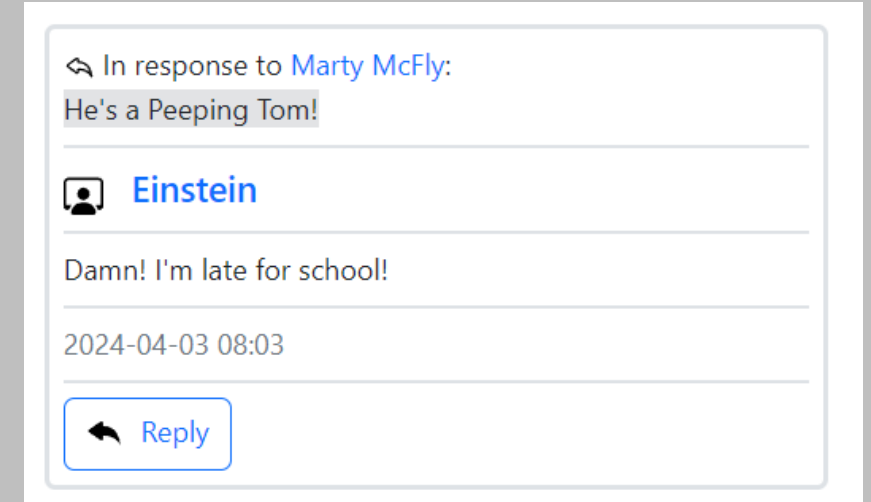
```
<app-send-message [url]="url" [channel] = "selectedChannel" [authorRegistered]="author"></app-send-message>
```

- Ricerca di un messaggio in base all'autore e al contenuto del messaggio

```
searchInput = searchInput.toLowerCase();  
  
this.filteredMessages = this.messages  
  .filter(message => message.author != null && message.author.toLowerCase().includes(searchInput)  
    || message.body != null && message.body.toLowerCase().includes(searchInput));
```

Message

- Gestione del layout di un singolo messaggio
- Gestione della modifica di un testo del messaggio
- Gestione della preview di un messaggio
- Gestione del messaggio di risposta



```
export class MessageComponent {  
  apiKey: string;  
  @Input() url: string;  
  @Input() message : Message;  
  @Input() authorRegistered: string;  
  onEdit: boolean = false;  
  editedMessage: string = "";  
  onReply: boolean = false;  
  repliedMessage: string = "";  
  @Input() parentMessage: Message | null = null;
```

Demo