

The EventEmitter is our Observable

The Plot-Component is only visible if movie is not null (a movie is selected)

Event-binding

Registration: Event-binding

Notification: The call of the emit method triggers the callback function to be invoked

Property-binding

The Callback function update the selectedMovie property in the AppComponent (parent)

Callback function, is invoked when the EventEmitter’s emit method is fired (the selected movie is passed as argument ($event)

Notice, the property movieSelected is annotated with @Output

The ngOnInit() are using the getMovies() which is a Service method injected via the constructor to the MovieList-Component

The property movies in MovieList is initialized by the ngOnInit() hook method

C/ Henrik Høltzer, Zealand March 31. 2020

Communication between child-components Overview (MovieInputOutput) overview

11a

11b

9b

9a

10b

10a

8b

8a

7

6

5

4

3

2

1

export class **AppComponent** {

  private selectedMovie: Movie;

  public movieSelected(movie: Movie) { this.selectedMovie = movie; }

}

<div class="jumbotron">

      <h1>Angular Movies <img src='assets/Image/movie\_img.png'> </h1>

      This small App demonstrates communication between child-components …

</div>

<div class="row">

….

     <app-movie-list **(movie-selected)="movieSelected($event)">** </app-movie-list>

      ….

     <app-movie-poster**[movie]="selectedMovie">** </app-movie-poster>

      ….

     <app-movie-plot **[movie]="selectedMovie">** </app-movie-plot>

 </div>

<section class="menu">  
 ….

        <h2>Movie list:</h2>

        <ul class="movies">

        <li **\*ngFor**="let **movie**of **movies**"

**[class.selected]="movie** === selectedMovie**"(click)="onSelect(movie)"** >

         <span >{{movie.Title}} </span>{{movie.Year}}

        </li>

….  
  </section>

export class **MoviePosterComponent {**

@Input()

  private movie: Movie;

  constructor() { }

  ngOnInit() {

  }

}

export class **MoviePlotComponent**  {

**@Input()**

  private movie: Movie;

  constructor() { }

}

<div **\*ngIf**="movie">

<div className="movieplot">

  <h3>Movie Plot: </h3>

  <div>{{movie.Plot}}</div>

</div>

</div>

<div \*ngIf="movie">

  <h3>Poster:</h3>

  <div><img class="poster" src={{movie.Poster}}></div>

</div>

export class **MovieListComponent** implements OnInit {

  private movies: Movie[];

**@Output('movie-selected')**

  private movieSelected = new **EventEmitter<Movie>();**

  constructor(public movieService: MovieService) {}

  ngOnInit() { this.movies = this.movieService.getMovies(); }

   public **onSelect**(**movie**: Movie) {

    this.movieSelected**.emit(movie);**

  }

}

@Injectable()

export class **MovieService** {

  public getMovies(): Movie[] { return MOCK\_MOVIES; }

}

