

Truyền dữ liệu thông qua url (RouterLink)

Truyền tham số là gì ? Tại sao phải truyền tham số?

Cũng tương tự input trong component ta thiết kế 1 trang hay 1 component đều muốn tái sử dụng được cho nhiều nội dung khác nhau. Tương tự đối với các thành phần layout cũng vậy.

=> Ví dụ: Các bạn xây dựng 1 layout để hiển thị thông tin chi tiết của 1 phim hay 1 sản phẩm, bài viết ... như hình bên dưới. Đều muốn chỉ xây dựng duy nhất 1 layout thay vì mỗi sản phẩm dàn mỗi layout => 10 sản phẩm ta có 10 layout (Không ai làm như vậy cả). Giống như input trong component người ta dùng tham số để đưa vào các RouterLink các giá trị, ứng với giá trị nào sẽ show thông tin với phần nội dung tương ứng.

Truyền dữ liệu thông qua url (RouterLink)

Chân Ái - Orange, Khôi, C | x +

nhaccuatui.com/bai-hat/chan-ai-orange-ft-khoi-ft-chau-dang-khoa-lybwdIMjKeYD9.html

NCT Khám Phá Bài Hát Playlist Tuyển Tập Video BXH Chủ Đề Top 100 ... Tìm bài hát, video, playlist, ca sĩ

Nghệ nhạc > Bài hát Nhạc Trẻ > Orange, Khôi, Châu Đăng Khoa

Chân Ái - Orange, Khôi, Châu Đăng Khoa 7.609.296

Orange, Khôi, Châu Đăng Khoa

cần một người

00:03 / 04:23 128kbps

Cung cấp bởi:
NCT Music Distribution

Thêm vào Chia sẻ

Thích 7.4K Chia sẻ

Các khúc **Chân Ái** do ca sĩ **Orange, Khôi, Châu Đăng Khoa** thể hiện, thuộc thể loại **Nhạc Trẻ**. Các bạn có thể nghe, download (tải nhạc) bài hát **chan ai** mp3, playlist/album, MV/Video **chan ai** miễn phí tại **NhacCuaTui.com**.

Lời bài hát: Chân Ái
Nhạc sĩ: Châu Đăng Khoa
Lời đăng bởi: dauahih

Tự đứng buồn thế
Tự đứng buồn thế

GỢI Ý DÀNH CHO BẠN
Thường thức những ca khúc phù hợp nhất với bạn
Nghe bài hát

NGHE TIẾP Autoplay

- Tâm Trí Yêu Em 2.490
Kiến Vinh
- Chàng Và Nàng 320.824
Rum
- SAY 25.664
Lena Lena, T.R.I
- 6 Năm Không Gặp 957.054
T-Akayz
- Có Một Ngày Buồn Như T... 34.472
Nguyễn Hà

xem thêm

TÀI QUẢNG CÁO

Có 2 trường hợp truyền nhận dữ liệu thông qua link (routerlink) và sự kiện.

- ❖ **Truyền nhận 1 tham số**
- ❖ **Truyền nhận nhiều tham số**

Thực hành truyền tham số

Bước 1: Trong HomeModule cấu hình route cho trang chi tiết như hình bên dưới

```
16 const homeRoutes:Routes = [  
17   {  
18     path: '',component:LayoutComponent,children:  
19     [  
20       {path: '',component:TrangChuComponent},  
21       {path: 'trangchu',component:TrangChuComponent},  
22       {path: 'chitiet/:id',component:ChiTietComponent},  
23       {path: 'datve',component:DatVeComponent},  
24       {path: 'pipe',component:PipeComponent},  
25       {path: 'quanlynv',component:QuanLyNhanVienComponent},  
26       {path: 'quanlyphim',component:QuanLyPhimComponent},  
27     ],  
28   },  
29 ]  
30 }  
31 ]  
32 |  
33 @NgModule({  
34   imports: [  
35     CommonModule,RouterModule.forChild(homeRoutes),FormsModule  
36   ],  
37   declarations: [TrangChuComponent, ChiTietComponent, DatVeComponent,LayoutComponent, PipeComponent, Shortcu  
38 ]  
39 })  
40 export class HomeModule { }  
41
```

Thực hành truyền tham số

Bước 2: Tại QuanLyPhimComponent.html ta để đường link dẫn đến trang chi tiết như hình bên dưới.

```
TS quan-ly-phim.service.ts x TS quan-ly-phim.component.ts <> quan-ly-phim.component.html x TS home.module.ts
angularfe10 > src > app > home > quan-ly-phim > <> quan-ly-phim.component.html > ...
1  <div class="container mt-3">
2      <div class="row">
3          <div class="col-md-4 mb-3" *ngFor="let phim of dsPhim.MangPhim">
4              <div class="card text-white bg-info">
5                  <img class="card-img-top" height="300" [src]="phim.HinhAnh" alt="">
6                  <div class="card-body">
7                      <div style="height:150px; overflow: hidden;" class="card-text">{{phim.MoTa}}</div>
8                  </div>
9                  <a routerLink="/chitiet/{{phim.MaPhim}}" class="btn btn-success">Chi tiết phim</a>
10             </div>
11         </div>
12     </div>
13 </div>
```

Thực hành truyền tham số

Bước 3: Sử dụng đối tượng `activatedRoute` để lấy tham số và dùng service gửi về backend

```
TS quan-ly-phim.service.ts TS chi-tiet.component.ts x TS quan-ly-phim.component.ts quan-ly-phim.component.html TS home.module.ts
angular10 > src > app > home > chi-tiet > TS chi-tiet.component.ts > ...
1 import { Component, OnInit, OnDestroy } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3 import { QuanLyPhimService } from 'src/app/core/Services/quan-ly-phim.service';
4 import { Subscription } from 'rxjs';
5
6 @Component({
7   selector: 'app-chi-tiet',
8   templateUrl: './chi-tiet.component.html',
9   styleUrls: ['./chi-tiet.component.css']
10 })
11 export class ChiTietComponent implements OnInit, OnDestroy {
12   phim: any = {};
13   private MaPhim;
14   private subParam: Subscription;
15   constructor(
16     private activateRouter: ActivatedRoute, //Đối tượng dùng để lấy tham số từ url
17     private qlPhimService: QuanLyPhimService
18   ) {}
19
20   ngOnInit() {
21     this.subParam = this.activateRouter.params.subscribe((params) => {
22       this.MaPhim = params.MaPhim;
23     });
24
25     this.qlPhimService.LayChiTietPhim(this.MaPhim).subscribe((phim: any) => {
26       this.phim = phim
27       console.log(this.phim)
28     });
29   }
30
31   ngOnDestroy(){
32     this.subParam.unsubscribe();
33   }
34 }
35
```

Thông qua đối tượng `activatedRoute` dùng thuộc tính `params` để lấy tham số từ url

Từ tham số thông qua service `phim` ta gọi api để lấy dữ liệu từ backend về thể hiện ra giao diện

Truyền nhận nhiều tham số

Bước 1: Tạo 1 route bình thường không có tham số

```
const homeRoutes:Routes = [  
  {  
    path: '', component:LayoutComponent, children:  
    [  
      {path: '', component:TrangChuComponent},  
      {path: 'trangchu', component:TrangChuComponent},  
  
      // {path: 'chitiet/:id', component:ChiTietComponent},  
      {path: 'chitiet', component:ChiTietComponent},  
  
      {path: 'datve', component:DatVeComponent},  
  
      {path: 'pipe', component:PipeComponent},  
      {path: 'quanlynv', component:QuanLyNhanVienComponent},  
      {path: 'quanlyphim', component:QuanLyPhimComponent},  
    ],  
  },  
]
```

Truyền nhận nhiều tham số

Bước 2: Tại trang quản lý phim thay vì ta truyền routerlink dạng 1 tham số sau khi tự '/' cuối cùng. Thì ta có thể truyền thông qua thuộc tính queryParams. Hoặc có thể viết `link?thamso1=[giatri]&thamso2=[giatri]` => có 2 cách viết

```
angularfe10 ▸ src ▸ app ▸ home ▸ quan-ly-phim ▸ <> quan-ly-phim.component.html ▸ <> div.container.mt-3 ▸ <> div.row ▸ <> div.col-md-4.mb-3 ▸ <> di
1
2 <div class="container mt-3">
3   <div class="row">
4     <div class="col-md-4 mb-3" *ngFor="let phim of dsPhim.MangPhim">
5       <div class="card text-white bg-info">
6         <img class="card-img-top" height="300" [src]="phim.HinhAnh" alt="">
7         <div class="card-body">
8           <div style="height:150px; overflow: hidden;" class="card-text">{{phim.MoTa}}</div>
9         </div>
10        <!-- <a routerLink="/chitiet?MaPhim={{phim.MaPhim}}&TenPhim={{phim.TenPhim}}" class="btn
11         <a [routerLink]="['/chitiet']"
12            [queryParams]="{MaPhim:phim.MaPhim,TenPhim:phim.TenPhim}" |
13            class="btn btn-success">Chi tiết phim</a>
14        </div>
15      </div>
16    </div>
17  </div>
18 </div>
```


Truyền nhận nhiều tham số

Bước 3: Để lấy thông tin các tham số từ routink ta cũng dùng đối tượng activateRouter thông qua thuộc tính queryParams.

```
ngOnInit() {  
  
  this.subParam = this.activateRouter.queryParams.subscribe((params) => {  
    this.MaPhim = params.MaPhim;  
    this.TenPhim = params.TenPhim;  
  })  
}
```

Form – Form Validation (Form)

Dùng để hỗ trợ việc lấy giá trị từ Form người dùng nhập vào đưa vào biến **model** (thuộc tính kiểu dữ liệu object ta tạo trong class của component đó). Trong phần two-way binding chúng ta đã thao tác với FormsModule 1 lần rồi.

Form - validation

Email	<input type="text" value="Email"/>
Name	<input type="text" value="Name"/>
School	<input type="text"/>
	<input type="button" value="Submit"/>

```
37 export class FormvalidationComponent implements OnInit {
38   public schools:any = [{id:'1',name:'cybersoft'},{id:'1',name:'myclass'}];
39
40   constructor() { }
41   Submit(value:any)
42   {
43     console.log(value);
44   }
45   ngOnInit() {
46
47   }
48 }
```

```
<form #registerForm="ngForm" class="form-horizontal" (ngSubmit) = "Submit(registerForm.value)">
<div class="form-group">
  <label for="inputEmail3" class="col-sm-2 control-label">Email</label>
  <div class="col-sm-10">
    <input type="email" class="form-control" id="inputEmail3" name="email" placeholder="Email" ngModel>
  </div>
</div>
<div class="form-group">
  <label for="Name" class="col-sm-2 control-label">Name</label>
  <div class="col-sm-10">
    <input type="text" class="form-control" id="Name" name="name" placeholder="Name" ngModel>
  </div>
</div>
<div class="form-group">
  <label for="school" class="col-sm-2 control-label">School</label>
  <div class="col-sm-10">
    <select id="school" class="form-control" name="school" ngModel>
      <option *ngFor="let school of schools" [value]="school.id">{{school.name}} </option>
    </select>
  </div>
</div>
<div class="form-group">
  <div class="col-sm-offset-2 col-sm-10">
    <button type="submit" class="btn btn-default">Submit</button>
  </div>
</div>
</form>
```

- 1/ Đối tượng **ngForm** sẽ lấy các giá trị từ các **ngModel** thông qua thuộc tính name.
- 2/ Sau đó ta sẽ dùng sự kiện (**submit**) gọi đến phương thức xử lý do chúng ta định nghĩa: gửi vào tham số là đối tượng **ngForm** (biến **registerForm.value**)

Form – Form Validation (Form)

Trong angular 6 hỗ trợ ta 1 module dành cho việc nhập liệu Form và kiểm tra giá trị đó là: **FormsModule**.

```
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3  import { FormvalidationComponent } from './formvalidation/formvalidation.component';
4  //Import FormsModule
5  import {FormsModule} from '@angular/forms';
6  @NgModule({
7    imports: [
8      CommonModule,FormsModule
9    ],
10    exports:[FormvalidationComponent],
11    declarations: [FormvalidationComponent]
12  })
13  export class FormvalidationModule { }
14
```

import FormsModule tại module cần sử dụng FormValidation

Form – Form Validation (Form)

❖Trạng thái của giá trị đầu vào thẻ input

- : **untouched** : Trường này chưa được chạm vào.
- : **touched** : Trường này đã được chạm vào.
- : **pristine** : Trường này chưa được thay đổi.
- : **dirty** : Trường này đã được thay đổi.
- : **invalid** : Trường có nội dung không hợp lệ.
- : **valid** : Trường có nội dung hợp lệ.

Giá trị trả lại khi ta kiểm tra các trạng thái sẽ là **true**. Nếu ngược lại thì sẽ là **false**

Form – Form Validation (Form)

❖ Kiểm tra rỗng với **required**

```
<input type="email" class="form-control" id="email" name="email" placeholder="Email" #email='ngModel' required
<!-- Kiểm tra rỗng -->
<div *ngIf="email.errors && (email.dirty || email.touched)">
  <div *ngIf="email.errors.required">
    Email không được rỗng !
  </div>
</div>
```

Xét trong lỗi đó có **required** không nếu có thì hiển thị div báo lỗi này.

Nếu có sự thay đổi của thẻ **input#email** hoặc **#email** được chạm vào. **&& #email errors** (required, không đúng định dạng, quá ký tự ...) thì sẽ hiển thị div này. Ví dụ (true && true = true) nên hiển thị

Form – Form Validation (Form)

❖ Kiểm độ dài ký tự với `max-length` & `min-length`


```
<div class="col-sm-10">
  <input type="email" minlength='7' maxlength='15' #email='ngModel' required ngModel class="form-control"
  <!-- Kiểm tra rỗng -->
  <div *ngIf="email.errors && (email.dirty || email.touched)" class="alert alert-danger">
    <!-- Kiểm tra rỗng -->
    <div *ngIf="email.errors.required">
      Email không được rỗng !
    </div>
    <!-- Kiểm tra độ dài -->
    <div *ngIf = "(email.errors.minlength || email.errors.maxlength)">
      Độ dài từ 7 - 15 ký tự
    </div>
  </div>
</div>
```

Nếu độ dài nằm ngoài
khoảng
7<X<15 thì sẽ hiển thị
div này

Form – Form Validation (Form)

❖ Kiểm tra định dạng chuỗi nhập vào với **pattern**

```
<input type="email" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$" minlength='7' maxlength='15' #email=''  
<!-- Kiểm tra rỗng -->  
<div *ngIf="email.errors && (email.dirty || email.touched)" class="alert alert-danger">  
<!-- Kiểm tra rỗng -->  
    <div *ngIf="email.errors.required">  
        Email không được rỗng !  
    </div>  
<!-- Kiểm tra độ dài -->  
    <div *ngIf = "(email.errors.minlength || email.errors.maxlength)">  
        Độ dài từ 7 - 15 ký tự  
    </div>  
<!-- Kiểm tra định dạng chuỗi nhập liệu -->  
    <div *ngIf = "email.errors.pattern">  
        Email không đúng định dạng  
    </div>  
</div>  
</div>
```



Kiểm tra định dạng với email thông qua thuộc tính **pattern** nếu đúng sẽ hiển thị div báo lỗi


Tham khảo thêm các pattern:

https://www.w3schools.com/tags/att_input_pattern.asp

Form – Form Validation (Form)

❖ Gợi ý kiểm tra password trùng nhau

```
<div class="form-group">
  <label class="control-label col-sm-2" for="pass">pass:</label>
  <div class="col-sm-10">
    <input type="password" class="form-control" id="pass" name="pass" #pass placeholder="Enter pass" ngModel>
  </div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2" for="repass">re pass:</label>
  <div class="col-sm-10">
    <input type="password" #repass [ngClass]='{"ng-invalid":(pass.value!=repass.value), 'ng-valid':!(pass.value!=repass.value)}'>
    <div *ngIf = "repass.errors || pass.value!=repass.value" class="alert alert-danger">
      <div *ngIf = "pass.value!=repass.value">
        password không trùng
      </div>
    </div>
  </div>
</div>
```



Ta dùng 2 biến đại diện cho 2 input **#pass #repass**
+Kiểm tra 2 giá trị nó có khác (!=) nhau không nếu khác nhau thì hiển thị div báo lỗi.
+Đồng thời tự xét tay thuộc tính **ng-invalid** hay **ng-valid** cho input thông qua **property binding** thuộc tính **directive ngClass**.


Form – Form Validation (Form)


❖Trạng thái đầu vào của thẻ Form

pristine : Không có trường nào được thay đổi.
dirty : Có một trường hoặc nhiều trường đã được thay đổi.
invalid : Nội dung Form là không hợp lệ.
valid : Nội dung Form là hợp lệ.
submitted : Form đã được submit.

Tương tự như thẻ input thì các thuộc tính của Form cũng có giá trị trả lại là true hoặc false.

```
<div class="form-group">  
  <div class="col-sm-offset-2 col-sm-10">  
    <button type="submit" *ngIf='!registerForm.submitted' [disabled]='!registerForm.valid' class="btn btn-defau.  
  </div>  
</div>
```

 Form chưa submit hiển thị, submit rồi thì ẩn đi

 Form chưa nhập hợp lệ thì không cho click, nhập hợp lệ thì enabled lên

Form – Form Validation (Form)

❖ Ngoài ra angular còn cung cấp ta thêm 1 số class CSS để validate.

AngularJS tự động thêm các class CSS cho Form và các thẻ Input tùy thuộc vào trạng thái của chúng. Đồng thời nó sẽ tự xóa đi class đó khi dữ liệu hợp lệ.

➤ Đối với Input

`ng-untouched` : Trường này chưa được chạm vào.

`ng-touched` : Trường này đã được chạm vào.

`ng-pristine` : Trường này chưa được thay đổi.

`ng-dirty` : Trường này đã được thay đổi.

`ng-valid` : Trường có nội dung hợp lệ.

`ng-invalid` : Trường có nội dung không hợp lệ.

`ng-valid-key` : key tương ứng với mỗi validation. Ví dụ: `ng-valid-required` , trường đã có nội dung

`ng-invalid-key` : Ví dụ: `ng-invalid-required` , trường không có nội dung

➤ Đối với form

`ng-pristine` : Không có trường nào được thay đổi.

`ng-dirty` : Có một trường hoặc nhiều trường đã được thay đổi.

`ng-valid` : Nội dung Form là hợp lệ.

`ng-invalid` : Nội dung Form là không hợp lệ.

`ng-valid-key` : key tương ứng với mỗi validation. Ví dụ: `ng-valid-required` , có một hoặc nhiều hơn một trường đã có nội dung

`ng-invalid-key` : Ví dụ: `ng-invalid-required` , chưa trường nào có nội dung

Form – Form Validation (Form)

❓ Ví dụ:

```
styles: [`  
  input.ng-invalid {  
    border-left:5px solid red;  
  }  
  input.ng-valid {  
    border-left:5px solid green;  
  }  
`]
```

- Ta thêm vào khi input hợp lệ tất cả thì thẻ input có border left là màu xanh.
- Ngược lại nếu chưa hợp lệ thì thẻ input có border left là màu đỏ

➤ Kết quả:

Form - validation

Email:

Độ dài từ 7 - 15 ký tự
Email không đúng định dạng

Name:

Name:

Form – Form Validation (Form)

➤ Cách set giá trị cho form từ Modal

Bước 1: Dùng viewchild dom tới Form đăng ký ngoài html thông qua local reference “registerForm” đại diện cho thẻ form

Bước 2: Vì giá trị form trả về là 1 object, nên khi ta set giá trị cho form cũng phải đưa vào 1 object với key là name của các ô input, value là giá trị value của input

HTML

```
<form #registerForm="ngForm" class="form-horizontal" (ngSubmit) = "Submit(registerForm.value)">
  <div class="form-group">
    <label for="inputEmail3" class="col-sm-2 control-label">Email</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="inputEmail3" name="email" placeholder="Email" ngModel>
    </div>
  </div>
  <div class="form-group">
    <label for="Name" class="col-sm-2 control-label">Name</label>
    <div class="col-sm-10">
      <input type="text" class="form-control" id="Name" name="name" placeholder="Name" ngModel>
    </div>
  </div>
  <div class="form-group">
    <label for="school" class="col-sm-2 control-label">School</label>
    <div class="col-sm-10">
      <select id="school" class="form-control" name="school" ngModel>
        <option ngFor='let school of schools' [value]="school.id">{{school.name}} </option>
      </select>
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <button type="submit" class="btn btn-default">Submit</button>
    </div>
  </div>
</form>
```

1/ Đối tượng **ngForm** sẽ lấy các giá trị từ các **ngModel** thông qua thuộc tính name.
2/ Sau đó ta sẽ dùng sự kiện (**submit**) gọi đến phương thức xử lý do chúng ta định nghĩa: gửi vào tham số là đối tượng **ngForm** (biến **registerForm.value**)

Typescript

```
}
export class DangkyComponent implements OnInit {
  @ViewChild('registerForm') dangkyF:NgForm

  SetValueUser(){
    this.dangkyF.setValue({
      TaiKhoan:'hieu',
      MatKhai:'hieu',
      confirmPass:'hieu',
      HoTen:'hieu',
      Email:'hieu',
      SoDT:'hieu',
      MaLoaiNguoiDung:'KhachHang',
      gender:'famale'
    })
  }
}
```

Bài tập tổng hợp

Please Sign Up It's free and always will be.

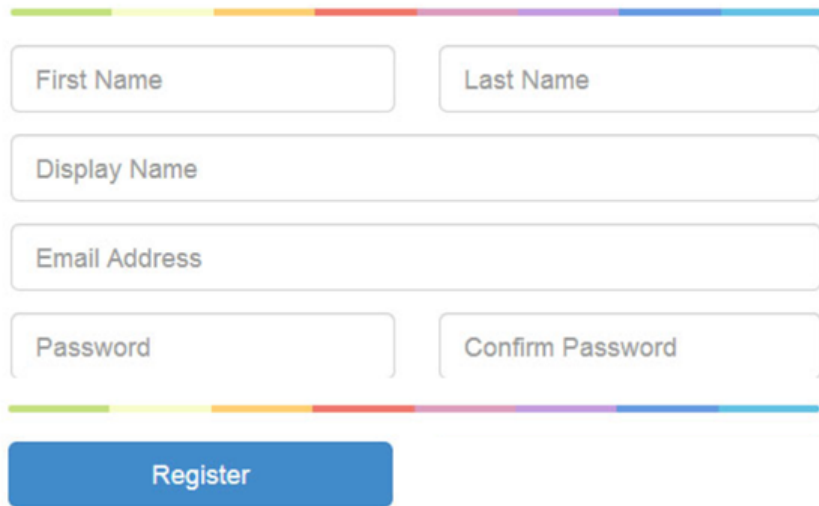
The form consists of several input fields and a button. At the top is a horizontal bar with segments of green, yellow, orange, red, pink, purple, and blue. Below this bar are two input fields for 'First Name' and 'Last Name'. Below those is a single input field for 'Display Name'. Below that is a single input field for 'Email Address'. Below that are two input fields for 'Password' and 'Confirm Password'. At the bottom is a blue button with the text 'Register'.

❖ Tạo form như hình:

- Tạo 1 component (trang) UserLogin thiết kế giao diện như hình
- Kiểm tra rỗng
- Kiểm tra tên từ 6 – 50 ký tự
- Kiểm tra định dạng email
- Kiểm tra password và confirm password
- Sau khi hợp lệ bấm register thì nó sẽ in ra table bên dưới. Với các cột tương ứng.
- Thêm 1 textbox để tìm kiếm theo họ tên hoặc email.

Bài tập tổng hợp

Please Sign Up It's free and always will be.



First Name

Last Name

Display Name

Email Address

Password

Confirm Password

Register

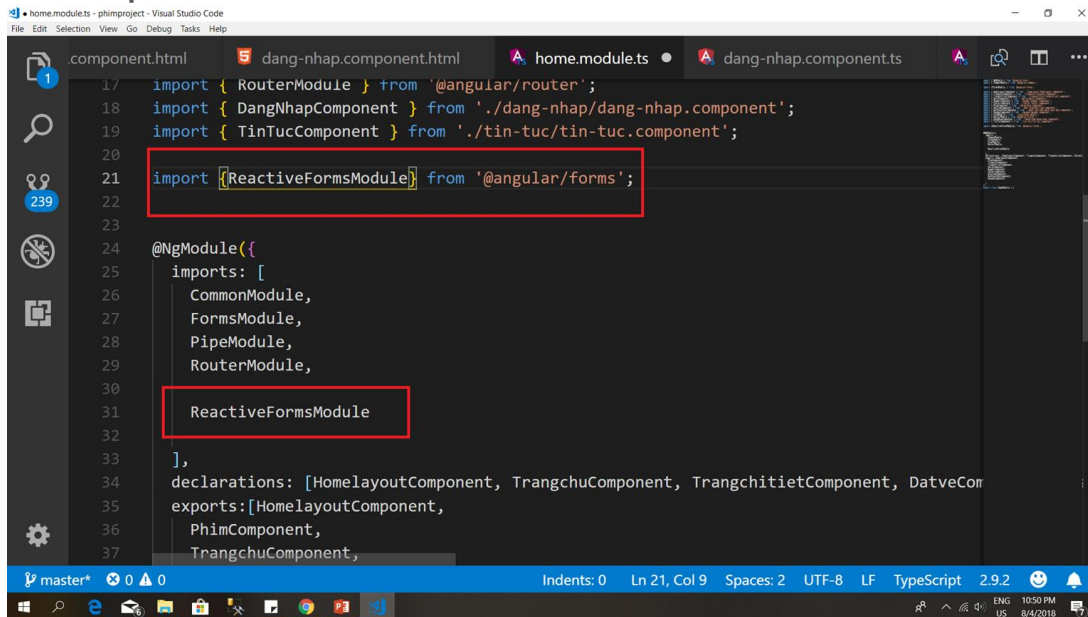
- Tạo thêm 1 component chứa Form đăng nhập với 2 input là email và password.
- Tạo 1 component hiển thị 2 link đăng ký, đăng nhập.
- Khi click vào link đăng ký thì form đăng ký hiển thị lên. Khi click vào link đăng nhập thì form đăng nhập hiển thị form đăng ký ẩn và ngược lại.
- Xây dựng chức năng cho form đăng nhập kiểm tra đăng nhập khi người dùng đăng nhập thành công xuất ra Xin chào email và ẩn form đó đi.

ReactiveForm

Ngoài FormsModule, Angular hỗ trợ thêm các module để handle form, trong đó có ReactiveForm.

Cách sử dụng ReactiveFormModule :

- Bước 1: import ReactiveFormModule vào module nào cần sử dụng



The screenshot shows the Visual Studio Code editor with a TypeScript file named `dang-nhap.component.ts` open. The file contains the following code:

```
17 import { RouterModule } from '@angular/router';
18 import { DangNhapComponent } from './dang-nhap/dang-nhap.component';
19 import { TinTucComponent } from './tin-tuc/tin-tuc.component';
20
21 import { ReactiveFormModule } from '@angular/forms';
22
23
24 @NgModule({
25   imports: [
26     CommonModule,
27     FormsModule,
28     PipeModule,
29     RouterModule,
30
31     ReactiveFormModule
32   ],
33   declarations: [HomelayoutComponent, TrangchuComponent, TrangchitietComponent, DatveComponent],
34   exports: [HomelayoutComponent,
35     PhimComponent,
36     TrangchuComponent,
```

The code is displayed in a dark-themed editor. The import statement on line 21, `import { ReactiveFormModule } from '@angular/forms';`, is highlighted with a red rectangular box. Similarly, the `ReactiveFormModule` entry within the `imports` array on line 31 is also highlighted with a red rectangular box. The status bar at the bottom indicates the file is using UTF-8 encoding, LF line endings, and TypeScript 2.9.2. The system tray shows the date as 6/4/2018 and the time as 10:50 PM.

ReactiveForm

- Bước 2: Xây dựng form Đăng Nhập ở giao diện HTML
- Bước 3: Xây dựng ReactiveForm Đăng Nhập ở Modal (typescript)
- Bước 4: Kết nối 2 form, dữ liệu người dùng nhập vào HTML được truyền vào ReactiveForm ở Modal
- Bước 5: Viết phương thức DangNhap, console.log ra value của form

```
<div class="container">
  <div class="row">
    <div class="col-5 mx-auto">
      <form [formGroup]="formDangNhap" (ngSubmit)="DangNhap()">
        <h4 class="display-4">Đăng Nhập</h4>
        <div class="form-group">
          <label for="">Tài Khoản</label>
          <input type="text" class="form-control" formControlName="TaiKhoan" >
        </div>
        <div class="form-group">
          <label for="">Mật Khẩu</label>
          <input type="text" class="form-control" formControlName="MatKhai">
        </div>
        <div class="form-group text-center">
          <button type="submit" class="btn btn-success">Đăng nhập</button>
        </div>
      </form>
    </div>
  </div>
</div>
```

```
dang-nhap.component.html  dangky.component.ts  home.module.ts  dang-nhap.component.ts
5 import { AuthService } from '../services/auth.service';
6
7 import { FormGroup, FormControl } from '@angular/forms';
8
9 @Component({
10   selector: 'app-dang-nhap',
11   templateUrl: './dang-nhap.component.html',
12   styleUrls: ['./dang-nhap.component.css']
13 })
14 export class DangNhapComponent implements OnInit {
15
16   public formDangNhap: FormGroup;
17
18   ngOnInit() {
19     this.formDangNhap = new FormGroup({
20       'TaiKhoan': new FormControl(null),
21       'MatKhai': new FormControl(null)
22     });
23   }
24
25 }
```

import từ angular forms

Khởi tạo ReactiveForm, kiểu dữ liệu là FormGroup

Các trường giá trị của reactiveForm, có thể hiểu các FormControl này giống như các input, có name là TaiKhoan và value ban đầu là null

ReactiveForm

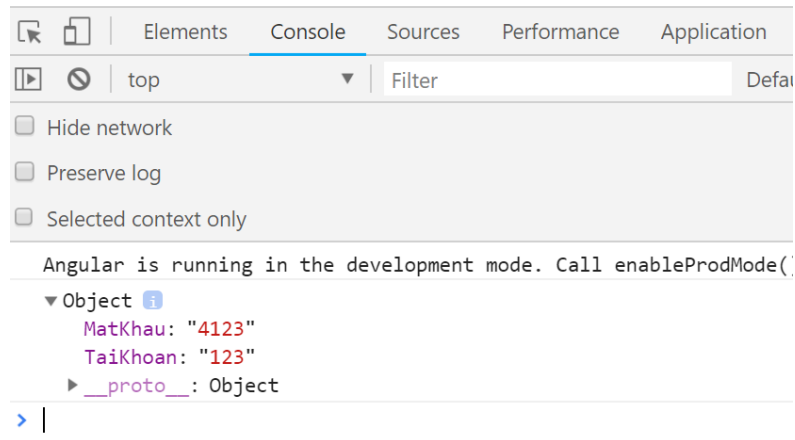
➤ Kết quả:

Đăng Nhập

Tài Khoản

Mật Khẩu

Đăng nhập



Validation với ReactiveForm

Ở HTML, kiểm tra không hợp lệ thì hiện alert

Gắn validators trực tiếp vào formControl

```
dang-nhap.component.html x dangky.component.ts home.module.ts dang-nhap.component.ts app.module.ts
6 <div class="form-group">
7   <label for="">Tai Khoản</label>
8   <input type="text" class="form-control" formControlName="TaiKhoan">
9
10   <div *ngIf="formDangNhap.get('TaiKhoan').invalid && formDangNhap.get('TaiKhoan').touched">
11     <div class="alert alert-danger" *ngIf="formDangNhap.get('TaiKhoan').errors['required']">
12       Vui lòng nhập tài khoản
13     </div>
14   </div>
15
16 </div>
17
18 <div class="form-group">
19   <label for="">Mật Khẩu</label>
20   <input type="text" class="form-control" formControlName="MatKhai">
21
```

```
public formDangNhap:FormGroup;

ngOnInit() {
  this.formDangNhap = new FormGroup({
    'TaiKhoan': new FormControl(null,Validators.required),
    'MatKhai': new FormControl(null,Validators.minLength(4))
  })
}
```

Gắn validation cho input tại đây, Kiểm tra TaiKhoan không được rỗng. MatKhai phải có độ dài tối thiểu là 4 kí tự

```
dang-nhap.component.html x dangky.component.ts home.module.ts dang-nhap.component.ts app.module.ts
15 </div>
16
17 </div>
18 <div class="form-group">
19   <label for="">Mật Khẩu</label>
20   <input type="text" class="form-control" formControlName="MatKhai">
21
22   <div *ngIf="formDangNhap.get('MatKhai').invalid && formDangNhap.get('MatKhai').touched">
23     <div class="alert alert-danger" *ngIf="formDangNhap.get('MatKhai').errors['minlength']">
24       Mật khẩu phải có tối thiểu 4 kí tự
25     </div>
26   </div>
27
28 </div>
29 <div class="form-group text-center">
30   <button type="submit" class="btn btn-success">Đăng nhập</button>
31 </div>
```

Kết hợp nhiều validators

Ở HTML, kiểm tra không hợp lệ thì hiện alert

```
dang-nhap.component.html x dangky.component.ts home.module.ts dang-nhap.component.ts app.module.ts
15 </div>
16 </div>
17 <div class="form-group">
18 <label for="">Mật Khẩu</label>
19 <input type="text" class="form-control" formControlName="MatKhai">
20
21 <div *ngIf="formDangNhap.get('MatKhai').invalid && formDangNhap.get('MatKhai').touched">
22 <div class="alert alert-danger" *ngIf="formDangNhap.get('MatKhai').errors['minlength']">
23   Mật khẩu phải có tối thiểu 4 kí tự
24 </div>
25 <div class="alert alert-danger" *ngIf="formDangNhap.get('MatKhai').errors['required']">
26   Vui lòng nhập mật khẩu
27 </div>
28 </div>
29 </div>
30
31 </div>
```

Gắn validators trực tiếp vào FormControl




```
export class DangNhapComponent implements OnInit {

  public formDangNhap: FormGroup;

  ngOnInit() {
    this.formDangNhap = new FormGroup({
      'TaiKhoan': new FormControl(null, Validators.required),
      'MatKhai': new FormControl(null, [Validators.required, Validators.minLength(4)])
    });
  }

  DangNhap(){
    console.log(this.formDangNhap.value);
  }
}
```

Bài tập tổng hợp

#	No.	Hình ảnh	Tên Phim	Sân xuất	Ngày công chiếu	Trailer
<input type="checkbox"/>	0		jkhakghajkhjklas	fgsdgdfg	2018-06-12T00:00:00	dsfgsdgdfg
<input type="checkbox"/>	1		Minions	Kyle Bald	2015-07-10T00:00:00	https://www.youtube.com/watch?v=WfQ_DoHRKc
						

ID	Title
Description	
Rating	Director
Release Date	Trailer
Image Url	
<button>ADD MOVIE</button>	

Tạo form add phim như hình:

- Kiểm tra rỗng
- Kiểm tra description trên 20 kí tự
- Kiểm tra Title phải là chữ
- Kiểm tra Rating phải là số
- Sau khi hợp lệ bấm Add Movie , tiến hành thêm phim vào mảng DanhSachPhim
- Tạo bảng động hiển thị danh sách phim
- Tên phim hiển thị ra bảng nhỏ hơn 10 ký tự, lớn hơn 10 kí tự thì cắt bớt và thêm dấu "..."
- Ngày chiếu hiển thị theo định dạng NgàyTháng-Năm Giờ : phút : giây
- ID hiển hiện ra bảng phải là chữ in hoa.
- Thêm button Cập nhật, sửa lại phim.