



FPT POLYTECHNIC



THỰC HỌC – THỰC NGHIỆP



Conceive Design Implement Operate

LẬP TRÌNH FRONT-END FRAMEWORK 1

FORM TRONG ANGULAR

- ⊙ Biết được form và hỗ trợ Angular với form
- ⊙ Biết xây dựng form dạng template và Reactive
- ⊙ Validation form với Angular



📖 Form là gì? Angular hỗ trợ như thế nào

📖 Xây dựng form dạng Template

📖 Xây dựng form dạng Reactive

📖 Xử lý validation trên form với dạng template và reactive





PHẦN 1: TEMPLATE FORM

FORM TRONG ANGULAR

```
<form>
  <div class="form-group">
    <label>Tài khoản</label>
    <input type="email" class="form-control" name="user">
  </div>
  <div class="form-group">
    <label>Mật khẩu</label>
    <input type="password" class="form-control" name="pass">
  </div>
  <button type="submit">Đăng Nhập</button>
</form>
```

```
{
  value:{
    user:'Fpoly',
    pass:'Technic'
  },
  valid:true
}
```

Tài khoản

Fpoly

Mật khẩu

.....

Đăng Nhập

Hai cách tiếp cận

Template-drive

Angular sinh ra form object từ
DOM

Reactive

Form được tạo trong chương
trình và đồng bộ với DOM

- Tạo form

```
<form>
<div class="form-group">
<label>Tài khoản</label>
<input type="email" class="form-control" name="user">
</div>
<div class="form-group">
<label>Mật khẩu</label>
<input type="password" class="form-control" name="pass">
</div>
<button type="submit" class="btn btn-primary">Đăng Nhập</button>
</form>
```

- Đăng ký các control cho angular, sử dụng thuộc tính ngModel

```
<form>
<div class="form-group">
<label>Tài khoản</label>
<input type="email" class="form-control" ngModel name="user">
</div>
<div class="form-group">
<label>Mật khẩu</label>
<input type="password" class="form-control" ngModel name="pass">
</div>
<button type="submit" class="btn btn-primary">Đăng Nhập</button>
</form>
```

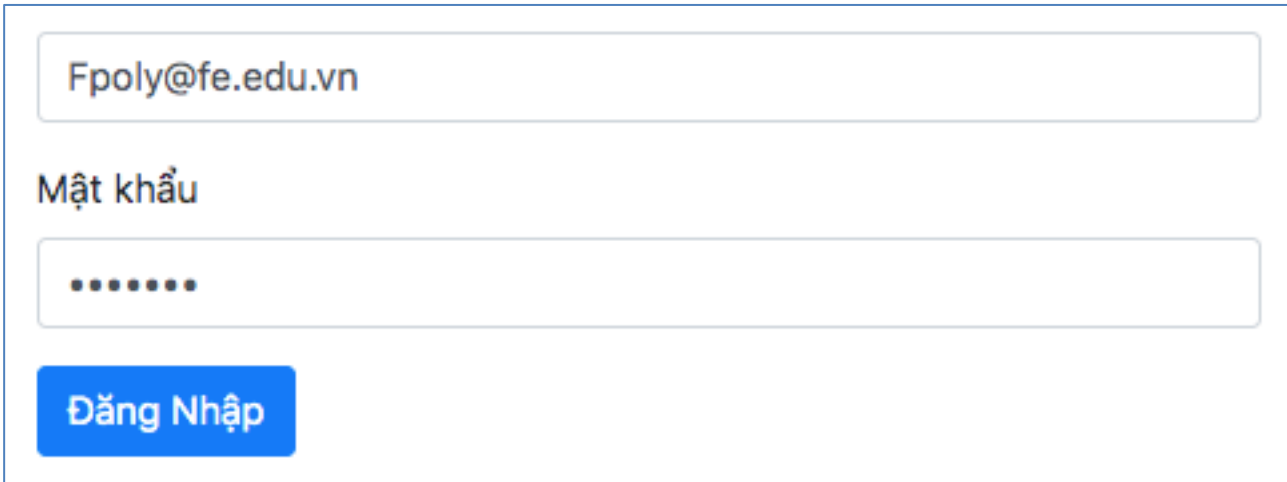

- Form submit

```

<form (ngSubmit)="onSubmit(loginForm)" #loginForm="ngForm" >
  <div class="form-group">
    <label>Tài khoản</label>
    <input type="email" class="form-control" ngModel name="user">
  </div>
  <div class="form-group">
    <label>Mật khẩu</label>
    <input type="password" class="form-control" ngModel name="pass">
  </div>
  <button type="submit" class="btn btn-primary">Đăng Nhập</button>
</form>
  
```

- Form submit và sử dụng control trên form

```
import { NgForm } from '@angular/forms';  
  
onSubmit(contactForm: NgForm) {  
  console.log(contactForm.value);  
}
```



The image shows a login form with a light blue border. It contains two input fields: the first is for an email address, containing the text 'Fpoly@fe.edu.vn'; the second is for a password, labeled 'Mật khẩu' and filled with dots. Below the password field is a blue button with the text 'Đăng Nhập' in white.

```
▶ {user: "Fpoly@fe.edu.vn", pass: "Technic"}
```

- Validation form

```
<input type="email" class="form-control" ngModel name="user"
```

```
  required minlength="6" #user="ngModel">
```

```
<span *ngIf="!user?.valid && (user?.dirty | | user?.touched)"> Email
```

```
Không hợp lệ </span>
```

```
</div>
```

- Dùng ngModel với gắn kết hai chiều

```
<label>Tài khoản</label>
```

```
<input type="email" class="form-control" [(ngModel)]="userData.tenDN" name="user"
  required minlength="6" #user="ngModel">
```

```
<a class="btn btn-secondary" (click)="suggest()">Tên đề xuất</a>
```

```
<span *ngIf="!user?.valid && (user?.dirty | user?.touched)">Email Không hợp lệ</span>
```

```
</div>
```

```
<div class="form-group">
```

```
<label>Mật khẩu</label>
```

```
<input type="password" class="form-control" [(ngModel)]="userData.matKhau"
  name="pass">
```

```
</div>
```

- Dùng ngModel với gắn kết hai chiều, khai báo và xử lý dữ liệu gắn kết

```
userData={  
  tenDN:'',  
  matKhau:''  
}
```

```
constructor() { }
```

```
ngOnInit(): void {  
}
```

```
onSubmit(contactForm:NgForm) {  
  console.log(contactForm.value);  
}
```

```
suggest(){  
  this.userData.tenDN='Fply@fe.edu.vn'  
}
```

- Truy cập form với @ViewChild

component.html

```
<form (ngSubmit)="onSubmit()" #f="ngForm" >
```

```
export class LoginComponent implements OnInit {
```

```
  @ViewChild('f') loginForm:NgForm;
```

```
  constructor() { }
```

```
  ngOnInit(): void {
```

```
  }
```

```
  onSubmit(){
```

```
    console.log(this.loginForm.value);
```

```
  }
```

```
}
```

component.ts



THỰC HIỆN FORM



PHẦN 2: REACTIVE FORM

- Phương pháp này tránh việc sử dụng các directive ví dụ như ngModel, required, etc,
- Tạo các Object Model ở trong các Component, rồi tạo ra form từ chúng.

Cài đặt

App.module.ts

```
import {FormsModule, ReactiveFormsModule} from '@angular/forms';
```

Manager-product.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormGroup } from '@angular/forms';
export class ManageProductComponent implements OnInit {
  productForm: FormGroup

  constructor() {}

  ngOnInit(): void {
  }
}
```

Tạo form trong code

```
formGroupObject=new FormGroup({  
  'controlName': new FormControl(value),  
  
  ...  
})
```

```
this.productForm=new FormGroup({  
  'productName': new FormControl(null),  
  'productCode': new FormControl('123-321'),  
  'description': new FormControl(null),  
  'price': new FormControl(null),  
  'starRating': new FormControl(null),  
  'imageUrl': new FormControl('https://lh3.googleusercontent.com/ab')  
})
```

Ví dụ tạo form thêm sản phẩm

Đồng bộ giữa HTML và form code: sử dụng directive formGroup và thuộc tính formControlName

```
<form [formGroup]="productForm">
```

```
<div class="form-group">
```

```
<label>Code sản phẩm</label>
```

```
<input type="number" formControlName="productCode" class="form-control">
```

```
</div>
```

```
<div class="form-group">
```

```
<label>Tên sản phẩm</label>
```

```
<input type="text" formControlName="productName" class="form-control">
```

```
</div>
```

Xử lý submit form

```
<form [formGroup]="productForm" (ngSubmit)="onSubmit()">
```

```
onSubmit(){
  console.log(this.productForm);
}
```

```
▼ FormGroup {_hasOwnPendingAsyncValidator: false, _parent: null, pristine: true,
  ► controls: {productName: FormControl, productCode: FormControl, description: FormControl,
    errors: null
    pristine: false
    status: "VALID"
  }
  ► statusChanges: EventEmitter_ {_isScalar: false, observers: Array(0),
    touched: true
  }
  ▼ value:
    description: "Ngôi trường đáng theo học"
    imageUrl: "https://lh3.googleusercontent.com/proxy/CI1q-1l3wrj4uAc1"
    price: 700000
    productCode: 123
    productName: "Fpoly"
    starRating: 5
  ► __proto__: Object
```

Code sản phẩm

Tên sản phẩm

Mô tả

Giá sản phẩm

Đánh giá

Ảnh sản phẩm

Thêm Sản Phẩm

Bỏ Qua

Thêm validation vào form: sử dụng đối tượng Validator được cung cấp trong module @angular/Core

```
'productName': new FormControl(null,Validators.required),  
'productCode': new FormControl('123-321',  
[Validators.required,Validators.minLength(10)]),
```

Truy cập control và xử lý validation: sử dụng hàm get của form code

```
<div class="form-group">
```

```
<label>Tên sản phẩm</label>
```

```
<input type="text" formControlName="productName" class="form-control">
```

```
<span *ngIf="!productForm.get('productName').valid &&  
productForm.get('productName').touched">
```

Bạn chưa nhập tên

```
</span>
```

```
</div>
```

FormArray: Thường dùng để tạo danh sách control động trên form reactive.

Ví dụ: Khi thêm một sản phẩm có kèm bộ sưu tập ảnh chưa biết số lượng ảnh là bao nhiêu?

Code sản phẩm

123-321

Tên sản phẩm

Mô tả

Giá sản phẩm

Đánh giá

Ảnh sản phẩm

https://lh3.googleusercontent.com/proxy/CI1q-1l3wrj4uAcfZWYYIQGscVLNJ3_KvDOW2sq

Album ảnh

Thêm ảnh

F.jpg

P.jpg

Thêm Sản Phẩm

Bỏ Qua

Cách thực hiện với FormArray

```
ngOnInit(): void {
  this.productForm=new FormGroup({
    'productName': new FormControl(null,Validators.required),
    'productCode': new FormControl('123-321',[Validators.required,Valida
    "description": new FormControl(null),
    "price": new FormControl(null),
    "starRating": new FormControl(null),
    "imageUrl": new FormControl('https://lh3.googleusercontent.com/proxy
    "listImg":new FormArray([])
  })
}

onSubmit(){
  console.log(this.productForm);
}

addImageUrl(){
  const control=new FormControl(null,Validators.required);
  (<FormArray>this.productForm.get('listImg')).push(control)
}
```

Tự định nghĩa validation

```
functionName(control:FormControl):{[s:string]:boolean}{  
    if(valid){  
        return value  
    }  
    return null  
}
```

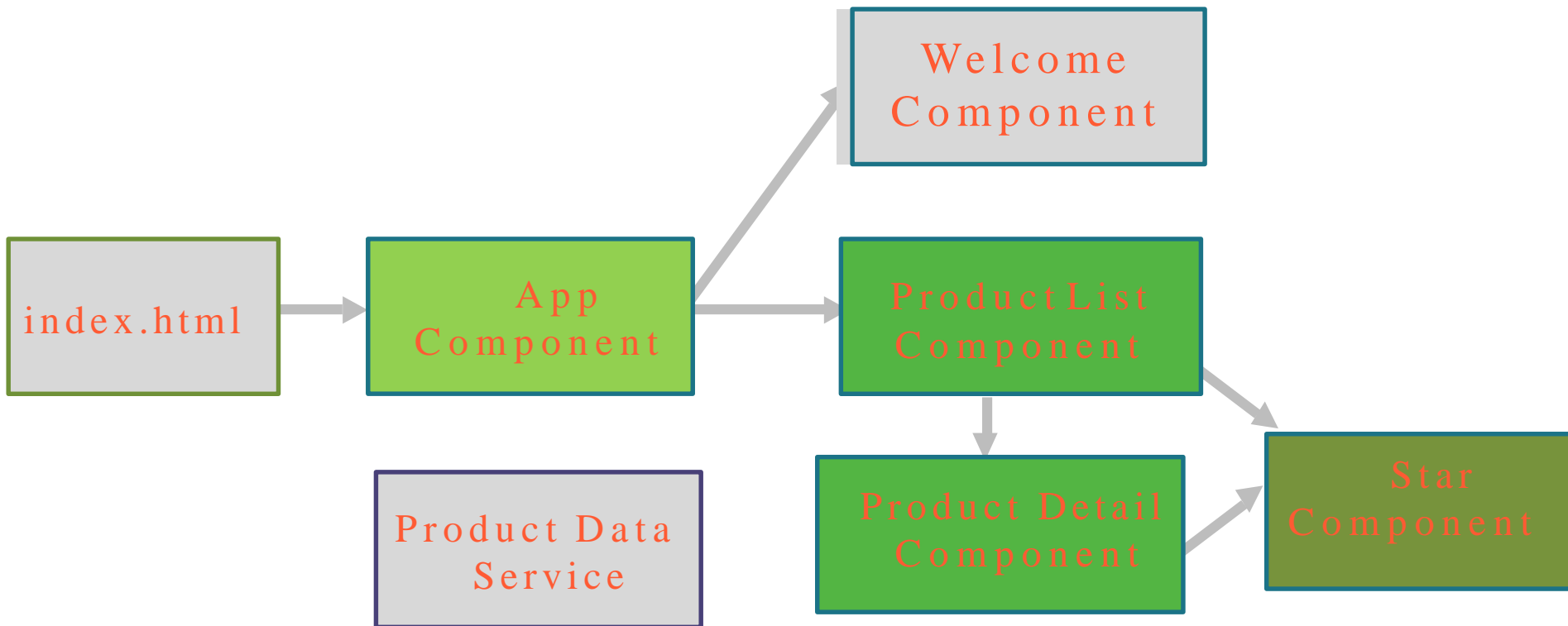
Sử dụng

```
new FormControl(null,[Validators.required, functionName.bind(this)])
```



XỬ LÝ FORM THÊM SẢN PHẨM

KIẾN TRÚC CỦA ANGULAR



- ☑ Form là gì? Angular hỗ trợ như thế nào
- ☑ Xây dựng form dạng Template
- ☑ Xây dựng form dạng Reactive
- ☑ Xử lý validation trên form với dạng template và reactive



thank
you!