BÁO CÁO

Lab 8: Angular và những điều đã được học





Sinh viên thực hiện:Nguyễn Minh Hiếu

Mã số sinh viên**:** PS16000

**Phần 1. Những điều đã được học về Angular Framework**

**I**. **Tổng quan về Angular Framework**

* Angular là một framework được dùng để viết front-end
* Angular framework sử dụng ngôn ngữ Typescript
* Do Google phát triển

# Các ưu điểm của Angular

* Cấu trúc code khá tường minh, chia làm module, service, 1 component cho 1 chức năng riêng, template, style được chia ra cho từng module
* Rất nhiều framework hổ trợ từ routing, validation, tới API call
* Một component có Input/Output, thuận tiện cho viêc đưa dữ liệu từ component cha ra component con và ngược lại
* Truyền dữ liệu từ component ra view thông qua data binding

# Các khái niệm cơ bản

|  |  |
| --- | --- |
| Component | * Thực hiện một chức năng nào đó * Gồm 3 phần: * View (file HTML, CSS) * Class (code Typescript) * Meta data (khai báo thông tin bổ sung) |
| Data binding | * Truyền dữ liệu từ component ra view |
| Module | * Angular cung cấp rất nhiều module để chúng ta sử dụng * Ví dụ: FormsModule, RouterModule, HttpClientModule,… * Mỗi ứng dụng sẽ có một module gốc để chạy ứng dụng là AppModule |
| Service | * Là một đoạn code để phục vụ một số chức năng nào đó trong ứng dụng * Ví dụ: lấy dữ liệu từ server, validate dữ liệu * Mỗi service được lưu trong file tên là Service.service.ts |

**II**. **Chi tiết về Angular Framework**

# Component trong Angular

Cấu trúc file của **app** component:

* app.component.css
* app.component.html
* app.component.spect.ts
* app.component.ts
* app.module.ts

# File app.module.ts

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | import { BrowserModule } from '@angular/platform-browser';  import { NgModule } from '@angular/core';    import { AppRoutingModule } from './app-routing.module';  import { AppComponent } from './app.component';    @NgModule({    declarations: [      AppComponent    ],    imports: [      BrowserModule,      AppRoutingModule    ],    providers: [],    bootstrap: [AppComponent]  })  export class AppModule { } |

# Truyền dữ liệu giữa các component

* @Input(): Nhận giá trị từ ngoài truyền từ cha vào component con
* @Output(): Đưa giá trị từ component con ra component cha

Trong @Output: dùng EventEmitter để thực hiện chuyển dữ liệu

# Truyền dữ liệu giữa các component



*Nguồn: https://angular.io/guide/binding-syntax*

# Binding dữ liệu ra view

* Cú pháp: {{ tên biến / tên hàm }}
* Ví dụ: <p> {{ myInfo.fullName }} </p>

# Binding dữ liệu ra thuộc tính của tag HTML

* Cú pháp: <tag [tên thuộc tính HTML]=”tên biến”> </tag>
* Ví dụ: <a **[href]**=”myLink.url”> Xem tại đây </a>

# Binding event trong DOM với tên hàm

* Cú pháp: <tag (tên sự kiện)=”tên hàm()”></tag>
* Ví dụ: <input **(keyup)**=”**handleInput($Event)**”>

# Định dạng dữ liệu

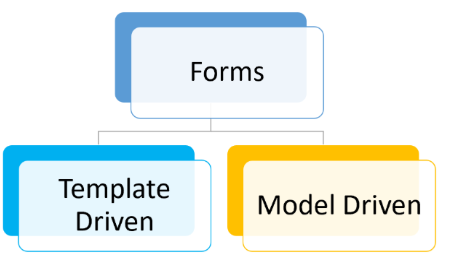
* <p> {{ myInfo.fullName **|** **uppercase** }} <p>
* <p> {{ myInfo.dayOfBirth **|** **date:’dd/MM/Y’**}} <p>
* <p> {{ myBill.total **|** **currency:’VND’: ‘đồng’** }} <p>

# Routing

* Giúp ứng dụng trở thành Single Page Application
* Thông qua việc khai báo đường route, routing giúp chuyển qua lại từng component một cách dễ dàng, không phải load trang

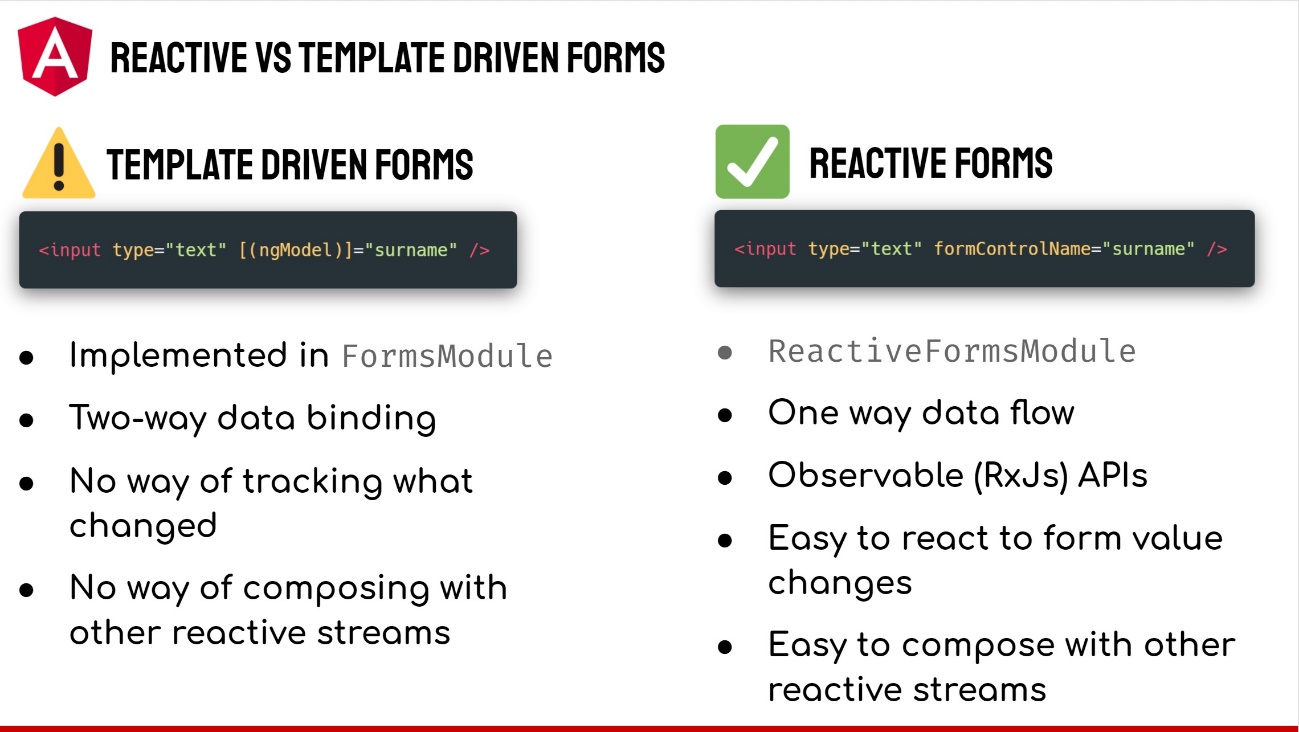


# Làm việc với Forms



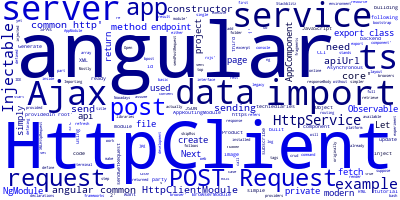
Có 2 cách:

* Template-Driven Forms
* Reactive Forms



|  |  |
| --- | --- |
| Sử dụng FormsModule | Sử dụng ReactiveFormsModule |
| Chuyển dự liệu hai chiều | Chuyển dữ liệu một chiều |
| Không thể theo dõi những gì đã thay đổi | Dễ dàng nhận biết giá trị của form bị thay đổi |
| Không thể composing với các luồng reactives khác | Dễ dàng compose với các luồng reactives khác |

# Http service trong Angular



**Http Serive** giúp chúng ta thực hiện các phương thức **GET**, **POST**, **DELETE**, **PUT** với dữ liệu thông qua giao thức **HTTP**

Để sử dụng được Http Serive chúng ta cần import **HttpClientModule** vào file module và khai báo sử dụng **HttpClient** trong component

# Xác thực người dùng trong Angular



Angular có các loại route guard như CanActive, CanActiveChild, CanActiveDeactive, CanLoad.

**Phần 2. Những điều chưa học về Angular Framework**

* Scope
* Filter
* MVC & MVVM
* Deep link

Hết.