**CÂU HỎI AUDIT MODULE 5**

**Angular**

1. **TypeScript là gì?**

Typescript là 1 phiên bản nâng cấp của Javascript – 1 superset của js.

1. **Tại sao nên sử dụng Typescript, Ưu điểm của Typescript**

Typescript thừa kế nhiều định nghĩa, khái niệm của C#, Java và nó ràng buộc chặt chẻ hơn js, hỗ trợ hướng đối tượng.

* **Ưu điểm:**
* Suport nhiều tính năng nâng cao có trong bản ES
* Dễ dàng gợi ý code có trong tool editor/IDE
* Hỗ trợ debug tốt
* Tạo sự nhất quán coding style cho team nhiều người.

1. **Làm sao để cài đặt TypeScript**

**- Cài đặt NodeJs npm**

**-** npm install -g typescript

**4. Có thể chuyển mã TypeScript thành mã Javascript không, bằng cách nào?**

Sử dụng lệnh tsc src/name

**5. Kể tên các kiểu dữ liệu trong Typescript**

**-** Ngoài các kiểu có trong js, ts còn có 1 số kiểu dữ liệu mới:

+ Enum: 1 tập hợp hằng số

+ Any: Bất kỳ

+ Tuple: cho phép khai báo mảng với các kiểu dữ liệu đã biết

+ Interface

**6. Kể tên các vòng lặp được typescript hỗ trợ**

Ngoài các vòng lặp đc hỗ trợ trong js. Còn có:

+ Foreach: Không trả về giá trị

+ Map: Trả về giá trị

+ Filter: Trả về giá trị có thêm bộ lọc

7. **Phân biệt sự khác nhau giữa var, let, const**

+ var: Biến toàn cục, có thể sử dụng trước khai báo sau

+ let: phạm vi trong block scope, khai báo mới được sử dụng

+ const khai báo hằng số.

**8**. **Trình bày khái niệm Single page web application**

- Là ứng dụng web 1 trang, hỗ trợ tải trang mà ko cần load toàn bộ trang.

**9. Lazy-loading là gì?**

- Cho phép 1 trang angular ko cần tải toàn bộ trang mà tải từng phần để giúp cải thiện tốc độ web

**10. Để khởi tạo một dự án angular sử dụng câu lệnh nào**

Ng new project-name

**11. Để chạy một dự án angular sử dụng câu lệnh nào**

Ng serve

Ng serve –port 1000

**12. Tại sao nên sử dụng Angular (điểm mạnh)?**

- Hỗ trợ phát triển ứng dụng 1 trang

- Tăng tốc độ phát triển ứng dụng

- Cộng đồng người dùng lớn (Nhiều tài liệu, hô trợ nhiều về lỗi…)

**13. Nhược điểm của Angular là gì?**

- Viết bằng js nên không an toàn hoặc giả sử người dùng tắt hỗ trợ js/ts thì ứng dụng không hoạt động được.

- Cần phải học typescript

**14. Angular Cli là gì? Làm sao để sử dụng nó ?**

- Angular Cli là công cụ đc phát triển để chạy trên giao diện cửa sổ ứng dụng. Là công cụ để tạo và quản lý dự án, kiểm thử deloy…

- Sử dụng trực tiếp trên cửa sổ terminal westom or cmd

- Chạy câu lệnh cài đặt

$ npm install -g @angular/cli

1. **File angular.json và package.js sử dụng để làm gì?**

- **angular.json :** dùng để cấu hình lại Angular Cli, giúp build ứng dụng angular

- **package.js** file này chứa các thư viện cần thiết cho dự án angular, ngoài ra nếu ta thêm một thư viện bên thứ 3 vào thì khai báo trong này.

**16. Giải thích kiến trúc của Angular**

**Kiến trúc angular bao gồm 3 thành phần chính.**

**- Module:** Tập hợp các chức năng của ứng dụng. Mặc định angular là Root Module

**- Component:** Là thành phần hiển thị lên giao diện người dùng. Nó giống như 1 mô hình mvc thu nhỏ. Mỗi web chia thành nhiều khối, mỗi khối có 1 bussiness xử lý riêng -> component.

**- Inject service:** Là thành phần trung gian, chứa các đoạn code dùng chung cho toàn bộ ứng dụng.

+ Giúp xử lý nghiệm vụ cho ứng dụng

1. **Angular đi theo mô hình nào? Phân biệt mô hình MVVM và MVC**

* Angular đi theo mô hình MVVM: Model view viewmodel.
* Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực hiện data binding, command.

|  |  |
| --- | --- |
| * VM | * Controler |
| * Đón nhận các sự kiện | * Đón nhận request |

**18. Component là gì ? các thành phần của component**

- Componet là thành phần hiển thị giao diện cho ứng dụng. Mỗi wep có nhiều khối mỗi khối là 1 component-có 1 code bussiness riêng. Nó giống như 1 mô hình MVC thu nhỏ.

- Các thành phần:

**+ Template:** Là thành phần hiển thị giao diện

**+ Class:** file.ts chứa các phương thức và thuộc tính là nơi xử lý nghiệp vụ bussiness

**+ Metadata:** Khai báo các thông tin:

* Tên selector
* Url tới template của component hiện tại
* Url tới file css

**19. Mô tả vòng đời component?**

- NgOnchange

- NgOninit

- NgDocheck

-NgDestroy

**20. Trình bày được khái niệm template trong angular**

**-** Template trong angular là: 1 thành phần của component cho phép hiển thị view dạng html

**21. Biên dịch AOT là gì? Những lợi thế của AOT ? JST (Java)**

**- Angualar có 2 loại JIT – biên dịch lúc chạy**

**- AOT – biên dịch trước khi ct chạy**

- AOT (Ahead of time): Code tới đâu biên dịch tới đó

**Ưu điểm:**

**-** Kết xuất ứng dụng Angular của bạn nhanh hơn - Với AOT, mã được biên dịch trong quá trình xây dựng. Do đó, trình duyệt tải mã thực thi sẵn sàng được hiển thị ngay lập tức.

**-** Giảm kích thước ứng dụng (ko sử dụng trình biên dịch)

**-** Chất lượng mã tốt hơn

**-** Các ứng dụng an toàn và mạnh mẽ hơn - Đó là bởi vì các mẫu HTML và các thành phần TypeScript không được đánh giá động khi chạy trong trình duyệt. Điều này dẫn đến ít cơ hội cho các cuộc tấn công.

**22. Trình bày về Data binding trong component**

- Là kỹ thuật đồng bộ dữ liệu giữa tầng view và component Ví dụ khi người dùng cập nhật data ở tầng view thì Angular cũng cập nhật giá trị đó ở component.

- Data binding trong Angular có thể chia ra làm 2 nhóm. Đó là one way binding (binding 1 chiều) và two way binding (binding 2 chiều).

**23. Hãy mô tả khái niệm Databinding One Way & two way phân biệt sự khác nhau?**

- Databinding One Way: Truyền dữ liệu 1 chiều từ ts=>html or ngược lại.

+ **Có 3 cách**: - Interpolation: ts => html {{}}

- NgModel: ts => html [ngModel]

- eventBinding html => ts: sự kiện (onclick) or (change)

- Databinding Two Way: Truyền dữ liệu 2 chiều từ ts ⬄ html và ngược. Bất kỳ sự thay đổi của bên nào cũng dẫn đến sự thay đổi của giữ liệu 2 phía

+ Sử dụng [(ngModel)] – import FormModule đế sử dụng.

**24. Directive trong angular là gì ?**

- Directive trong angular là các thành phần thuộc tính mở rộng của html do angular định nghĩa, hỗ trợ cho CRUD.

**25 Kể tên các directive được hỗ trợ trong angular**

- Component Directive

- Structor Directive: ngFor/ngIf/[ngSwitch]

- Attribute Directive: ngClass: thêm xóa 1or nhiều css class

ngStyle: thêm xóa 1or nhiều css style

**26. Làm thế nào để lấy được dữ liệu từ component cha**

**27. Làm thế nào để lấy được dữ liệu từ component con**

**28. Pipe là gì? Kể tên 1 số pipe thường sử dụng**

- Pipe là tính năng được xây dựng sẳn trong angular giúp format dữ liệu: | namePipe

- Một số pipe thông dụng:

AsyncPipe

UpperCasePipe

CurrencyPipe

SlicePipe

DatePipe

LowerCasePipe

**29. ngFor là gì? Cách sử dụng**

Là 1 directive dùng để thực hiện lặp trong angular:

Cú pháp: \*ngFOr=”let temp of list”

**30 ngIf là gì? Cách sử dụng**

Là 1 directive dùng để thực hiện câu lệnh điều kiện trong angular:

\*ngIf=”biểu thức điều kiện”

**31. Các cách tạo form trong Angular**

**32. Phân biệt Template driven form và Reactive form**

**33.** **Trình bày 1 số validate thường dùng. Cách custom validate như thế nào?**

- Minlength, maxlength, required, min, max, pattern

- để xem lại

**34. router trong angular dùng để làm gì?**

**- Để điều hướng view của component nào sẽ được hiển thị**

**- Cách sử dụng:**

**+ Import vào AppMoudule: AppRoutingModule**

**+ Dùng <router-outet> - xác định vị trí hiển thị component**

**+ Dùng router-link để khai báo đường dẫn**

**35. Cách để lấy tham số trên URL trong angular**

- Inject ActivateRouter

- Sử dụng paramMap để lấy tham số

**36. làm sao để kiểm tra người dùng có quyền truy cập vào 1 route ?**

Có 5 loại guards khác nhau và mỗi loại được gọi theo một trình tự cụ thể. Hành vi của bộ định tuyến được điều chỉnh khác nhau tùy thuộc vào guard nào được sử dụng. Các loại đó là:

* CanActivate
* CanActivateChild
* CanDeactivate
* CanLoad
* Resolve

**37. Trình bày sự khác nhau giữa navigateByUrl và navigate**

|  |  |
| --- | --- |
| NavigateByUrl | Navigate |
| Chỉ truyền được url đạng string | Truyền được nhiều tham số |

38. Sự khác biệt giữa RouterModule.forChild và RouterModule.forRoot là gì?

|  |  |
| --- | --- |
| RouterModule.forChild | RouterModule.forRoot |
| forChild tạo một mô-đun chứa tất cả các khai báo sử dụng trong module đó | forRoot tạo một mô-đun chứa tất cả các chỉ thị, khai báo sử dụng trong toàn bộ dự án |

**39. Trình bày được ý nghĩa và khái niệm Observable**

- Đại diện cho 1 giá trị trả về trong tương lai, và khi có nhiệm vụ điều phối các gtri đó đến cho Observer

- Sử dụng để xử lý bất đồng bộ

**40. Trình bày được ý nghĩa và khái niệm Observer**

**Một Observer sẽ định nghĩa phương thức callback để xử lý 3 loại trạng thái mà một observable có thể gửi như sau:**

* next:  Bắt buộc phải có, được sử dụng để lấy ra dữ liệu từ Observable
* error: Không bắt buộc, được sử dụng để xử lý các lỗi xảy ra trong quá trình xử lý.
* complete: Không bắt buộc, đượ sử dụng để xử lý các nội dung mà chúng ta muốn được xử lý sau khi một Observers hoàn thành.

41 Phân biệt sự khác nhau giữa Promise và Observable

|  |  |
| --- | --- |
| **Promise** | **Observable** |
| – Chỉ trả về một giá trị duy nhất, đó có thể là object, array, number, … | – Trả về một hoặc nhiều giá trị |
| Không thể cancel được request | – Có thể cancel request |
| Được khởi tạo ngay lập tức mặc dù chưa có bất kỳ đăng ký nào. | – Chỉ được khởi tạo khỉ và chỉ khi có đăng ký đến observable đó(có listener) |
| Có kq mới xử lý đc | * Có hàm xử lý trước khi có kq trả về   – Có thể retry (tự động reconnect gọi lại)  – Có thể dùng với event  – Có các thao tác tiền xử lý dữ liệu như: filter, map, … |

**42. Trình bày được ý nghĩa và khái niệm subcribe**

Là 1 phương thức dùng để theo dõi sự thay đổi dữ liệu/nhận kết quả khi xử lý bất đồng bộ vs Obserable

Phương thức subcribe() sẽ nhận vào một Observer. **Một Observer sẽ định nghĩa phương thức callback để xử lý 3 loại trạng thái mà một observable có thể gửi như sau:**

* next:  Bắt buộc phải có, được sử dụng để lấy ra dữ liệu từ Observable
* error: Không bắt buộc, được sử dụng để xử lý các lỗi xảy ra trong quá trình xử lý.
* complete: Không bắt buộc, đượ sử dụng để xử lý các nội dung mà chúng ta muốn được xử lý sau khi một Observers hoàn thành.

43 ngModule là gì?

- Có trong note :D