# angular59道面试题(https://github.com/minsion)

# 🛅 面试题 1. 简述 AngularJS 的数据双向绑定是怎么实现的?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题 ▶

### 试题回答参考思路:

# 一、什么是AngularJS

AngularJS是一个JavaScript框架,它诞生于2009年,由Misko Hevery 等人创建,后为Google所收购。是一款优秀的前端JS框架,已经被用于Google的多款产品当中。

AngularJS有着诸多特性,最为核心的是: MVVM、模块化、自动化双向数据绑定、语义化标签、依赖注入等等。

理解什么是自动化双向数据绑定以及其实现的方式。

### 二、什么是数据绑定

首先我们要理解什么是数据绑定。目前我们所看到的网站页面中,基本是由数据和设计两部分组成。将设计转换成浏览器能理解的语言,便是html和css主要做的工作。而将数据显示在页面上,并且有一定的交互效果(比如点击等用户操作及对应的页面反应)则是js主要完成的工作。出于用户体验、数据安全等方面考虑,很多时候我们不会一提交数据就刷新页面(get请求),而是通过向后端请求相关数据,然后通过无刷新加载(ajax)的方式进行页面更新(post请求)。当数据进行更新后,页面上相应的位置也能自动做出对应的修改,这就是数据绑定。可以看出,数据绑定是M(model,数据)通过VM(model-view,数据与页面之间的变换规则)向V(view)的一个修改。

#### 三、什么是双向绑定

所谓的双向绑定,就是从界面的操作能实时反映到数据,也就是数据的变更能够实时展现到 界面

在新的框架中(angualr, react, vue等),通过对数据的监视,一旦数据发生变化,就根据已经写好的规则对页面进行修改,这样便实现了V——M——VM——V的一个双向绑定。它是相当于增加了一条反向的路,在用户操作页面(比如在Input中输入值)的时候,数据能及时发生变化,并且根据数据的变化,页面的另一处也做出对应的修改。有一个常见的例子就是淘宝中的购物车,在商品数量发生变化的时候,商品价格也能及时变化。

### 四、双向绑定的实现

AngularJs主要通过scopes模型实现数据双向绑定。AngularJS的scopes包括以下四个主要部分:

- (1) digest循环以及dirty-checking, 包括\$watch,\$digest,和\$apply。
- (2) Scope继承 这项机制使得我们可以创建scope继承来分享数据和事件。
- (3) 对集合 数组和对象 的有效dirty-checking。
- (4) 事件系统 \$on, \$emit, 以及\$broadcast。

AngularJs 为 scope 模型上设置了一个 监听队列\$watch, 用来监听数据变化并更新 view。每次绑定一个东西到 view(html) 上时 AngularJs 就会往 \$watch 队列里插入一条 \$watch, 用来检测它监视的 model 里是否有变化的东西。当浏览器接收到可以被 angular

context 处理的事件时,\$digest 循环就会触发。\$digest 会遍历所有的 \$watch。从而更新DOM。

# 🛅 面试题 2. 详述 Angular 的数据绑定采用什么机制? 详述原理

推荐指数: ★★★★★ 试题难度: 初级 试题类型: 原理题▶

## 试题回答参考思路:

双向数据绑定是 AngularJS 的核心机制之一。当 view 中有任何数据变化时,会更新到 model ,当 model 中数据有变化时,view 也会同步更新,显然,这需要一个监控。【相 关教程推荐: 《angular教程》】

原理就是,Angular 在 scope 模型上设置了一个监听队列,用来监听数据变化并更新 view 。每次绑定一个东西到 view 上时 AngularJS 就会往 \$watch 队列里插入一条 \$watch,用来检测它监视的 model 里是否有变化的东西。当浏览器接收到可以被 angular context 处理的事件时, \$digest 循环就会触发,遍历所有的 \$watch, 最后更新 dom

# 🛅 面试题 3. ng-if与ng-show/hide的区别有哪些?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

# 试题回答参考思路:

ng-if与ng-show/hide区别有以下几方面:

第一个区别是, ng-if在后面表达式为true时才创建DOM节点, ng-show则在初始时就创建了, 可以通过切换ng-hide类控制元素的显示或隐藏。

第二个区别是, ng-if会(隐式地)产生新作用域。这样在ng-if中通过ng- model绑定的数据改变时,不会影响外面元素中插入的数据。下列代码说明ng-if指令内外的name属性变量已经是两个变量了。

```
<br />
{ name } }<br />
<div ng-if="true"><br />
<nput type="text" ng-model="name"><br />
</div><br />
```

ng-show不存在此问题。而避免这类问题出现的办法是,始终为页面中的元素绑定对象的属性( data.msg) ,而不是直接绑定到基本变量(msg)上。

```
<br />
{ { data.msg } }<br />
<div ng-if="true"><br />
<input type="text" ng-model="data.msg"><br />
</div><br />
```

# 🖹 面试题 4. 当使用 ng -repeat指令迭代数组时,如果数组中有相同值,会有什么问题? 如何解决?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

一般会提示 Duplicates in a repeater are not allowed, 通过加 track by \$index表达式可解决。当然, 也可以通过 track by跟踪任何一个普通的值, 只要能唯一地标识数组中的每一项即可(建立DOM和数据之间的关联)

{ {item } }

🛅 面试题 5. ng-cick中写的表达式,能使用 JavaScript原生对象上的方法吗?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题▶

## 试题回答参考思路:

不只是ng-click中的表达式,只要是在页面中,都不能直接调用原生的 JavaScript方法,因为这些并不存在于与页面对应的 Controller的 \$scope中。

举个例子,执行以下代码会发现,什么也没有显示。

```
{ { parseInt (100.5) } }
但如果在$scope中定义了以下 parseInt函数,就没有问题了。
$scope. parseInt =function (num) {
return parseInt (num)
}
对于这种需求,使用一个 filter是不错的选择。
{ {12.34 | icktParseInt)

.filter ('icktParseInt', function() {
return function (value) {
return parseInt (value)
}
})
```

## 🛅 面试题 6. 简述Angular filter的含义是什么?

推荐指数: ★★★★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

filter表示格式化数据,接受一个输入,按某规则处理,返回处理结果。

内置 filter有以下9种。

1: date (日期)。

```
2: currency (货币)
3: limitTo (限制数组或字符串长度)
4: orderBy (排序)
5: lowercase (小写)。
6: uppercase (大写)
7: number (格式化数字,加上千位分隔符,并接受参数以限定小数点位数)
8: filter (处理一个数组,过滤出含有某个子串的元素)
9: json (格式化json对象)
```

# 🖿 面试题 7. 请说出 Augluar filter的两种使用方法?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

```
试题回答参考思路:

两种使用方法如下。
在模板中使用

{ { time | date : 'yyyy-MM-dd' }}

在Javascript t中使用。
.controller ('main', function ($scope, $filter) {
  console.log ($filter ('date') (Date. now () , 'HH:mm:ss'))
```

# 🖿 面试题 8. 简述factory、 service和 provider是什么关系?

推荐指数: ★★★★★ 试题难度: 高难 试题类型: 原理题 ▶

```
试题回答参考思路:
```

```
它们都是用来实现自定义服务的方法。但是定义服务的方式不同,factory通过return暴露
服务的接口对象。
factory('ickt', function(){
return {
color: 'red '
}
})
service通过this暴露服务的接口对象
.service ('ickt', function() {
this .color = 'green '
})
provider创建一个可通过 config配置的服务,通过$get方法的返回值暴露接口。
.provider ('ickt', function () {
this .$get = function () {
return {
color: 'blue'
}
}
```

})

从底层实现上来看, service调用了 factory,返回其实例;factory调用了 provider,返回其\$get中定义的内容。factory和 service的功能类似,只不过 factory是普通function,可以返回任何数据。service是构造器,不需要返回;provider是加强版factory,返回一个可配置的 factory。

## 🛅 面试题 9. 对于两个平级界面模块a和b,如果a中触发一个事件,有哪些方式能让b知道?详述原理

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题 ▶

### 试题回答参考思路:

这个问题换一种说法就是,如何在平级界面模块间进行通信。有两种方法种是共用服务,一种是基于事件。

共用服务。

在 Angular中,通过 factory可以生成一个单例对象,在需要通信的模块a和b中注入这个对象即可。

基于事件。

这个又分两种方式。

第一种是借助父控制器。在子 controller中向父控制器发射 (\$emit)一个事件,然后在父 控制器中监听(\$on)事件,再广播(\$ broadcast)给其他子控制器,这样通过事件携带 的参数,实现了数据经过父控制器,向同级控制器传播。

第二种是借助 \$rootScope。每个 Angular应用默认有一个根作用域 \$rootScope,根作用域位于最顶层,从它向下挂着各级作用域。所以,如果子控制器直接使用 \$rootScope广播和接收事件,那么就可实现平级之间的通信

## 🛅 面试题 10. 简述Angular应用应当如何进行目录结构的划分?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

对于小型项目,可以按照文件类型划分,比如以下形式。

CSS

JS

controllers

models

services

filters

templates

但是,对于规模较大的项目,最好按业务模块划分,比如以下形式。

CSS

modules

user

controllers

models

services

```
filters
templates
activity
controllers
models
services
filters
templates
modules下最好再有一个 common目录来存放公共数据
```

# 🖿 面试题 11. Angular应用通常使用哪些路由库? 各自的区别是什么?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题 ▶

# 试题回答参考思路:

Angular应用通常使用 neRoute和 ui.router。

无论是 ngRoute还是ui. router,作为框架额外的附加功能,都必须以模块依赖的形式引入。

区别如下。

- (1) ngRoute模块是 Angular自带的路由模块,而 ui.router模块是基于 ngRoute模块开发的第三方模块,属于UI指令规范。
- (2) ui.router是基于状态 (state)的, ngRoute是基于URL的, ui.router模块具有更强大的功能,主要体现在视图的嵌套方面。
- (3) 使用 ui.router能够定义有明确父子关系的路由,并通过ui-view指令将子路由模板插入父路由模板的中,从而实现视图嵌套。而在 neRoute中不能这样定义,如果同时在父子视图中使用了会陷入死循环。

```
示例如下:
ngRoute
angular .module ('ickt', [' ngRoute'])
//在 config中定义路由
.config (function ($routeProvider) {
// console. log ('config', this, arguments)
$routeprovider
//首页
.when (' / home', {
template: 'home { { color } }',
controller:function ($scope) {
$scope. color ='red'
}
})
//配置默认路由
.otherwise ({
redirectTo: '/home'
})
ui .router
```

```
angular .module ('ickt', [ 'ui .router' ] )
//配置路由
.config ( function($stateProvider ) {
// console. log ($stateProvider)
//配置首页
$stateProvider
.state (' home',{
/*定义规则*/
url: '/home/: id? a&b,
template: '
home page { { color } }
//定义控制器
controller:function ($scope) {
$scope. color ='red'
}
})
})
```

🖿 面试题 12. 对于不同团队开发的 Angular应用,如果要做整合,可能会遇到哪些问题?如何解决?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题 ▶

```
试题回答参考思路:
可能会遇到不同模块之间的冲突比如一个团队的开发在 moduleA下进行,另一团队的开发在 moduleB下进行。
angular .module ('myApp.moduleA',[])
.factory ('serviceA, function(){
...
})
angular .module ('myApp. moduleB', [])
.factory ('serviceA', function(){
...
})
angular .module ('myApp. moduleB', [])
.cfactory ('serviceA', function() {
...
})
angular .module ('myApp, [' myApp. moduleA', 'myApp. moduleB'])
这会导致两个 module下面的 serviceA发生覆盖。
在 Angular 1.x中并没有很好的解决办法,所以最好在前期进行统一规划,做好约定,严格按照约定开发,每个开发人员只写特定的区块代码
```

## 🖿 面试题 13. Angular的缺点有哪些? 如何克服这些缺点?

推荐指数: ★★★★★ 试题难度: 初级 试题类型: 原理题▶

强约束,导致学习成本较高,对前端不友好。当遵守 Angular的约定时,效率会提高。但不利于SEO,因为所有内容都是动态获取并渲染生成的,搜索引擎无法爬取种解决办法是,对于正常用户的访问,服务器响应 Angular应用的内容;对于搜索引擎的访问,则响应专门针对SEO的HTML页面。作为MVVM框架,因为实现了数据的双向绑定,对于大数组、复杂对象会存在性能问题

## 🛅 面试题 14. 简述如何优化 Angular应用的性能?

推荐指数: ★★★★★ 试题难度: 高难 试题类型: 原理题 ▶

## 试题回答参考思路:

优化方式如下。

- (1) 减少监控项(比如对不会变化的数据采用单向绑定)。
- (2) 主动设置索引(指定 track by, 简单类型默认用自身当索引, 对象默认使用 \$\$hashKey, 比如改为 track by item id)。
- (3) 降低渲染数据量(比如分页,或者每次取一小部分数据,根据需要再取)。
- (4) 数据扁平化(比如对于树状结构,使用扁平化结构,构建map和树状数据。当操作树时,由于跟扁平数据是同一个引用,因此树状数据变更会同步到原始的扁平数据)

## 🛅 面试题 15. 如何看待 Angular1.2中引入的 controllerAs语法?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

```
试题回答参考思路:
在 Angular1.2以前, 在view上的任何绑定都是直接绑定在 $scope上的。
function myCtrl ($scope) {
$scope.a=='aaa';
$scope.foo = function(){
}
使用 controllerAs,不需要再注入 $scope服务了, controller变成了一个很简单的
JavaScript对象,一个更纯粹的ViewModel。
function myCtr1 () {
//使用vm捕获this可避免内部的函数在使用this时导致上下文改变。
var vm = this;
vm.a = 'aaa';
从源码实现上来看, controllerAs语法只是把 controller这个对象的实例用 controllerAs
别名表示,并在 $scope上创建了一个属性。
if (directive .controllerAs) {
locals. $scope [ directive .controllerAs ]= controllerInstance;
}
```

# 🛅 面试题 16. 简述如何理解 Angular2?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

## 试题回答参考思路:

与 Angular 1.x相比, Angular 2的改动很大,几乎是一个全新的框架。

Angular2基于 TypeScript语法实现,在大型项目团队协作时,使用强语言类型更有利优点是组件化,提升开发和维护的效率。其他优点还有 module支持动态加载,以及new router、 promise的原生支持等。它迎合未来标准,吸纳其他框架的优点,值得期待

## 

推荐指数: ★★★★ 试题难度: 高难 试题类型: 原理题 ▶

### 试题回答参考思路:

它依赖于 \$interpolation服务,在初始化页面HTML后,它会找到这些表达式,并进行标记,于是每遇见一个{{}},就会设置一个\$ watch。而 \$interpolation会返回一个带有上下文参数的函数,最后执行该函数,渲染数据

## 🛅 面试题 18. Angular中的 digest周期是如何触发的?

推荐指数: ★★★★ 试题难度: 高难 试题类型: 原理题 ▶

### 试题回答参考思路:

在每个 digest周期中, Angular总会对比 scope上 model的值,一般 digest周期都是自动触发的,也可以使用 \$apply进行手动触发

## 🖿 面试题 19. Augluar 如何取消\$timeout?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

要停止\$timeout,可以用 cancel方法。示例如下。
var timebar =\$timeout (function (){
\$scope,msg='有课前端网'
}, 1000)
\$timeout.cancel (timebar)

### 面试题 20. Augluar 如何注销一个\$watch()?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题▶

#### 试题回答参考思路:

```
要注销一个 Swatch,方法如下:
var watchFn = $scope. $watch ('msg', function (newValue) {
if (newValue ==='icketang) {
    // $watch () 会返回一个停止注册的函数,可以执行该函数进行注销
    watchFn ()
}
console. log (newValue)
})
```

# 🛅 面试题 21. Augluar 自定义指令中 restrict有几种类型?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

## 试题回答参考思路:

restrict有4种类型。

A: 匹配属性。 E: 匹配标签。

C: 匹配 class。

M: 匹配注释。

当然,可以设置多个值,比如AEC,进行多个匹配

# 🛅 面试题 22. Augluar 自定义指令中, scope配置中的@、=和&修饰符有什么区别?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题 ▶

### 试题回答参考思路:

在 scope中, @、=和&在进行值绑定时分别表示如下含义。

@表示单向绑定,数据只能由父作用域流入子作用域。

=表示双向绑定,数据可以在父子作用域中双向传递。

&表示父作用域传递的属性或方法等数据,在子作用域中作为方法获取

# 🛅 面试题 23. 请列出Augluar 至少3种实现不同模块之间通信的方式?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题 ▶

### 试题回答参考思路:

不同模块之间的通信方式如下。

通过自定义服务实现作用域之间的通信。

通过\$on注册事件,在其他作用域之间通过使用\$emit或\$broadcast发布消息实现通信。

借助 \$rootScope实现作用域间的通信。

通过 \$scope的\$parent等属性,访问父子或者兄弟作用域,实现通信。

通过自定义指令实现通信。

### 🛅 面试题 24. 简述有哪些措施可以改善 Angular的性能?

推荐指数: ★★★★★ 试题难度: 高难 试题类型: 原理题 ▶

## 试题回答参考思路:

改善措施有以下几个。

- (1) 关闭 debug, 方法是使用 \$compileProvider。myApp. config (function (\$compileProvider) {\$compileProvider .debugInfoEnabled (false);});
  - (2) 使用一次绑定表达式,即{{: msg}。
  - (3) 减少 watcher数量
  - (4) 在无限滚动加载中避免使用 ng-repeat
  - (5) 使用性能测试小工具挖掘 Angular的性能问题

# 🛅 面试题 25. Angular中使用 jQuery好吗?

推荐指数: ★★★★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

普遍认为在 Angular中使用 jQuery并不是一个很好的尝试。其实当我们学习

Angular的时候,应该做到从零去接受 Angular的思想,例如,数据绑定、使用 Angular自带的一些API、合理地组织路由、写相关指令和服务等。Angular自带的很多API完全可以替代 jQuery中常用的API,可以使用 angular.element、\$http、 \$timeout、ng-init等。或许引入 jQuery可以在解决问题(比如使用插件)时有更多的选择。当然,这通过影响代码组织来提高工作效率。随着对 Angular的理解深入,在重构时会逐渐摒弃当初引入 jQuery时的一些代码。

所以我们还是应该尽力遵循 Angular的设计理念

## 🖿 面试题 26. Angular中的\$http服务有什么作用?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题♪

## 试题回答参考思路:

\$http是 Angular中的一个核心服务,用于读取远程服务器的数据。

可以使用内置的 \$http服务直接与服务器端进行通信。\$http服务只是简单地封装了浏览器原生的 XMLHttpRequEst对象

# 🖿 面试题 27. 简述在Augluar 写 controller逻辑时,需要注意什么?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题▶

#### 试题回答参考思路:

#### 要注意以下几点

(1) 简化代码

(2) 坚决不能操作DOM节点。

DOM操作只能出现在指令(directive)中。不应该出现在服务(service)中。Angular 倡导以测试驱动开发,在服务或者控制器中出现了DOM操作,也就意味着无法通过测试。当然,这只是一方面。重要的是,使用 Angular中的数据双向绑定技术,就能专注于处理业务逻辑,无须关心各种DOM操作。如果在 Angular的代码中还有各种DOM操作,还不如直接使用 jQuery去开发

# ■ 面试题 28. 列举Augluar 常见的自定义指令参数?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

## 试题回答参考思路:

常见的自定义指令参数如下。

restrict: 定义指令在DOM中的声明形式,如E(元素)、A(属性)、C(类名)、 M(注 聚)

template: 定义字符串模板。

templateUrl: 定义模板文件路径。

compile:可以返回一个对象或函数。如果设置了 compile函数,说明我们希望在指令将数据绑定到DOM之前,进行DOM操作,在这个函数中进行诸如添加和删除节点等DOM操作是

安全的。

link:链接函数,可以实现指令的一些功能

# 🛅 面试题 29. 简述Angular和 jQuery的区别是什么?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题▶

# 试题回答参考思路:

Angular基于数据驱动,所以 Angular适合做数据操作比较繁杂的项目。 jQuery基于DOM驱动, jQuery适合做DOM操作多的项目

## 🛅 面试题 30. Augluar 什么是作用域数据丢失? 如何解决作用域数据丢失问题?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题▶

#### 试题回答参考思路:

当在作用域中修改数据的时候,更新的数据没有渲染到页面中,或者没有触发相应的业务逻辑执行,我们就说这部分数据丢失了。解决数据丢失问题有4种方式。

使用相应的服务,如 setTimeout方法可以用 \$timeout服务完美代替。

使用\$digest方法进行检测,但只能检测该方法执行前修改的数据,后面修改的数据无法检测。

使用\$apply方法进行检测,但只能检测该方法执行前修改的数据,后面修改的数据无法检测,本质上,\$apply在内部调用\$digest方法实现检测在 Sapply方法的回调函数中修改数据,\$apply方法可以屏蔽错误中断应用程序的执行,但只能检测前面以及回调函数内部修改的数据,在该方法后面修改的数据无法检测

# 🖿 面试题 31. Angular2应用程序的生命周期 hooks是什么?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题 ▶

## 试题回答参考思路:

Angular2组件/指令具有生命周期事件,是由@ angular/core管理的。@ Angular/core会创建组件,渲染它,创建并呈现它的后代。当@ angular/core的数据绑定属性更改时,处理就会更改,在从DOM中删除其模板之前,就会销毁它。Angular提供了一组生命周期hooks(特殊事件)函数,并在需要时执行操作。构造函数会在所有生命周期事件之前执行。每个接口都有一个前缀为ng的hook方法。

组件特定 hooks的意义如下。

ngAfterContentInit: 组件内容初始化完成之后。

ngAfterContentChecked: 在 Angular检查投影到其视图中绑定的数据之后。

ngAfterViewInitAngular: 创建组件的视图后。

ngAfterViewChecked: 在 Angular检查组件视图的绑定之后

# 🖿 面试题 32. 和使用 Angular1相比,使用 Angular2有什么优势?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

#### 试题回答参考思路:

Angular2是一个平台,它不仅是一种语言,还具有更快的速度、更好的性能和更简单的依赖注入。它支持模块化开发,跨平台,具备了 EMAScript6和 Typescript的优点,如路由灵活,具备延迟加载功能等

## 🖿 面试题 33. 简述Angular2中路由的工作原理?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题▶

#### 试题回答参考思路:

路由是能够让用户在视图纽组件之间导航的机制。Angular2简化了路由,并提供了在模抉级的(延迟加载)配置,定义更灵活。

Angular应用程序具有路由器服务的单个实例,并且每当URL改变时,相应的路由就与路由配置数组进行匹配。在成功匹配时、它会应用重定向。

此时,路由器会构建 ActivatedRoute对象的树,同时包含路由器的当前状态。在重定向之前,路由器将通过运行保护(CanActivate)来检查是否允许新的状态。

Route Guard只是用来检查路由授权的接口方法。保护运行后,它将解析路由数据并通过将所需的组件实例化到< router-outlet>< router- outlet>中来激活路由器状态

# 🖿 面试题 34. 简述什么是事件发射器?它是如何在 Angular2中工作的?

推荐指数: ★★★★★ 试题难度: 高难 试题类型: 原理题▶

## 试题回答参考思路:

Angular2不具有双向检测系统,这与 Angular1不同。在 Angular2中,组件中发生的任何改变总是从当前组件传播到其所有的子组件中。

如果一个子组件的更改需要反映到其父组件的层次结构中,就可以使用事件发射器API(吞吐器)来发出事件简而言之, EventEmitter是在@ angular/core模块中定义的类,由组件和指令使用,用来发出自定义事件。

使用emit (value)方法发出事件。通过模块的任何一个组件,可以使用订阅方法来实现事件发射的订阅,例如以下代码

```
@Directive ({
selector: '[app Demo]',
//定义发送的数据
outputs: [ 'sendMessage' ]
})
export class DemoDirective {
sendMessage = new EventEmitter ( );
constructor ( ) {
setInterval (o=>
//发布消息this sendMessage emit ('hello)
}, 2000)
//父组件
```

# 🖿 面试题 35. Angular2应用中,应该注意哪些安全威胁?

推荐指数: ★★★★★ 试题难度: 高难 试题类型: 原理题 ▶

## 试题回答参考思路:

就像任何其他客户端或Web应用程序一样,Angular2应用程序也应该遵循一些基本准则来减轻安全风险、例如以下基本准则。

避免为组件使用/注入动态HTML内容。

如果使用外部HTML,也就是来自数据库或应用程序之外的HTML,就需要清理它。

不要将外部网址放在应用程序中,除非它是受信任的。

避免网址重定向,除非它是可信的。

考虑使用AOT编译(运行前编译)或离线编译。

通过限制API,选择使用已知或安全环境/浏览器的App来防止XSRF攻击

# 🖿 面试题 36. 简述什么是 Shadow dom? 它是如何帮助 Angular2更好地执行的?

推荐指数: ★★★ 试题难度: 中级 试题类型: 原理题 ▶

#### 试题回答参考思路:

Shadow DOM是HTML规范的一部分,它允许开发人员封装自己的HTML标记、 CSS和 JavaScript代码。Shadow DOM以及其他一些技术,使开发人员能够像< audio>标签样构 建自己的一级标签、Web组件和API。总的来说,这些新的标签和API称为Web组件。 Shadow dom提供了更好的关注分离,通过其他的 HTML DOM元素实现了更少的样式与脚本的冲突。

因为 Shadow DOM本质上是静态的,同时也是开发人员无法访问的,所以它是一个很好的候选对象。它缓存的DOM将在浏览器中呈现得更快,并提供更好的性能。此外,还可以较好地管理 Shadow dom,同时检测 Angular2应用的改变,并且可以有效地管理视图的重新绘制

### 🖿 面试题 37. 简述什么是Augluar AOT编译? 它有什么优缺点?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题 ▶

#### 试题回答参考思路:

AOT编译,即 Ahead Of Time编译(运行前编译)。在构建时, Angular编译器会将 Angular组件和模板编译为原生 JavaScript和HTML。编译好的HTML和 JavaScript将会 部署到Web服务器中,以便浏览器节省编译和渲染时间。

优点如下。

更快的下载速度。因为应用程序已经编译,许多 Angular编译器相关库就不再需要捆绑,应用程序包变得更小,所以该应用程序可以更快地下载。

更少的HTTP请求数。如果没有捆绑应用程序来支持延迟加载,对于每个关联的HTML和 CSS,都会有一个单独的服务器请求。但是预编译的应用程序会将所有模板和样式与组件对 齐,因此到达服务器的HTTP请求数量会更少。

更快的渲染速度。如果应用程序不使用AOT编译,那么应用程序完全加载时,编译过程会发生在浏览器中。这需要等待下载所有必需的组件,然后等待编译器花费大量时间来编译应用程序。使用AOT编译,就能实现优化。

在构建时检测错误。由于预先编译可以检测到许多编译时错误,因此能够为应用程序提供更好的稳定性。

缺点如下。

仅适用于HTML和CSS、其他文件类型需要前面的构建步骤。

没有 watch模式, 必须手动执行并编译所有文件。

需要维护AOT版本的 bootstrap文件。

在编译之前,需要清理步骤。

## 🛅 面试题 38. 简述如何评价 Backbone和 Angular?

推荐指数: ★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

Backbone具有依赖性,依赖于 underscore.js。Backbone+ Underscore+ jQuery(或 Zepto)比一个Angular应用多出了两次HTTP请求。

Backbone的模型没有与UI视图数据绑定,而是需要在视图中自行操作DOM来更新或读取UI数据。Angular与此相反,模型直接与UI视图绑定。模型与U视图的关系,通过指令封装。Angular内置的通用指令就能实现大部分操作了。也就是说,基本不必关心视图与UI视图的关系,直接操作视图即可,UI视图自动更新。

在你输入特定数据时, Angular的指令就能渲染相应的U视图。它是一个比较完善的前端 MVW框架,包含模板、数据双向绑定、路由、模块化、服务、依赖注入等功能模板的功能 强大,并且是声明式的,自带丰富的Aη gular指令

# 🛅 面试题 39. 浅谈ionic和 Angular的区别?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题 ▶

### 试题回答参考思路:

ionic是一个用来开发混合手机应用的、开源的、免费的代码库。可以优化HTML、 CSS和 JavaScript的性能,构建高效的应用程序。

Angular通过新的属性和表达式扩展了HTML. Angular可以构建一个单一页面应用程序(Single Page Application,SPA)。

ionic是一个混合App开发工具,它以 Angular为中间脚本工具,所以如果要使用 ionic开发 App,就必须了解 Angular

# 🛅 面试题 40. Augluar 在用双大括号绑定元素时,如何解决内容闪烁的问题?

推荐指数: ★★★★ 试题难度: 高难 试题类型: 原理题 ▶

### 试题回答参考思路:

在 Angular内部,可以向元素中添加ng- clock属性来实现元素的隐藏效果。

{ {message } }

如果绑定纯文字的内容,建议使用ng-bind的方式,而非双大括号。

# ■ 面试题 41. 如何理解Augluar ng-repeat指令中的作用域继承关系?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题▶

#### 试题回答参考思路:

在调用 ng-repeat指令显示数据时,在新建DOM元素时, ng-repeat也为每个新建的 ng-repeat也为每个新建的 ng-repeat也为每个新建的

尽管如此,它们的父级作用域是相同的,都是构建控制器时注入的\$scope对象,调用 angular.element( element).scope方法可以获取某个DOM元素对应的作用域。通过某个 DOM元素对应的作用域又可以访问它的父级作用域,从而修改绑定的数据源

## 🛅 面试题 42. 简述Angular 4.0的优点 ?

推荐指数: ★★ 试题难度: 中级 试题类型: 原理题 ▶

# 试题回答参考思路:

Angular是由谷歌维护的一个开源JavaScript框架。

目前有两个大的版本,一个是Angular 1.5,一个是Angular 4.0。两个版本的差别非常大,4.0 是一个完全重写的版本。Angular1.5叫AngularJS,Angular 2.0之后的都只叫Angular。

Angular新特性

全新的命令行工具AngularCLI

服务器端渲染

移动和桌面的兼容

组件:是Angular应用的基本构建块,可以把一个组件理解为一段带有业务逻辑和数据的HTML。

服务: 用来封装可重用的业务逻辑。

指令:运行你向HTML元素添加自定义行为。

组件、服务、指令是用来完成功能的。模块是用来打包这些东西的。

模块用来将应用中不同部分组织成一个Angular框架可以理解的单元。比如这几个组件,服务和指令可以完成登陆注册的功能,我就把他们归到一个模块中。当开发另一个项目的时候也需要登陆注册,我直接把这个模块引入到另一个项目中就可以了

组件之间可以有父子关系

组件可以去调用服务

# 🛅 面试题 43. 简述使用 Angularjs 项目开发中 你使用过那些第三方的插件?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

## 试题回答参考思路:

AngularUi ui-router oclazyload 等等

## 🖿 面试题 44. Augluar 表达式 {{yourModel}} 是如何工作的?

推荐指数: ★★★★★ 试题难度: 高难 试题类型: 原理题 ▶

### 试题回答参考思路:

它依赖于 \$interpolation服务,在初始化页面html后,它会找到这些表达式,并且进行标记,于是每遇见一个 {{}},则会设置一个 \$watch。而 \$interpolation会返回一个带有上下文参数的函数,最后该函数执行,则算是表达式 \$parse 到那个作用域上

# 🛅 面试题 45. ng-repeat 迭代数组的时候,如果数组中有相同值,会有什么问题,如何解决?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题▶

### 试题回答参考思路:

会提示 Duplicates in a repeater are not allowed. 加 track by \$index 可解决。当然,也可以 trace by 任何一个普通的值,只要能唯一性标识数组中的每一项即可(建立 dom 和数据之间的关联)

# 🖿 面试题 46. 请问Angular.js 是 mvc 还是 mvvm 框架?

推荐指数: ★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

mvvm

#### 解析:

首先阐述下你对 mvc 和 mvvm 的理解:

首先为什么我们会需要 MVC? 因为随着代码规模越来越大,切分职责是大势所趋,还有为了后期维护方便,修改一块功能不影响其他功能。还有为了复用,因为很多逻辑是一样的。而 MVC 只是手段,终极目标是模块化和复用。

mvvm 的优点

低耦合: View 可以独立于 Model 变化和修改,同一个 ViewModel 可以被多个 View 复用;并且可以做到 View 和 Model 的变化互不影响;

可重用性:可以把一些视图的逻辑放在 ViewModel, 让多个 View 复用;

独立开发: 开发人员可以专注与业务逻辑和数据的开发 (ViewModemvvmdi 计人员可以专注于 UI(View)的设计;

可测试性:清晰的 View 分层,使得针对表现层业务逻辑的测试更容易,更简单。

在 angular 中 MVVM 模式主要分为四部分:

View: 它专注于界面的显示和渲染,在 angular 中则是包含一堆声明式 Directive 的视图 模板。

ViewModel: 它是 View 和 Model 的粘合体,负责 View 和 Model 的交互和协作,它负责给 View 提供显示的数据,以及提供了 View 中 Command 事件操作 Model 的途径;在 angular 中\$scope 对象充当了这个 ViewModel 的角色;

Model: 它是与应用程序的业务逻辑相关的数据的封装载体,它是业务领域的对象, Model并不关心会被如何显示或操作,所以模型也不会包含任何界面显示相关的逻辑。在 web 页面中,大部分 Model 都是来自 Ajax 的服务端返回数据或者是全局的配置对象;而 angular 中的 service 则是封装和处理这些与 Model 相关的业务逻辑的场所,这类的业务

服务是可以被多个 Controller 或者其他 service 复用的领域服务。

Controller: 这并不是 MVVM 模式的核心元素,但它负责 ViewModel 对象的初始化,它将组合一个或者多个 service 来获取业务领域 Model 放在 ViewModel 对象上,使得应用界面在启动加载的时候达到一种可用的状态。

mvc 的界面和逻辑关联紧密,数据直接从数据库读取。mvvm 的界面与 viewmode 是松耦合, 界面数据从 viewmodel 中获取。所以 angularjs 更倾向于 mvvm

# 🖿 面试题 47. 简述Angular.JS的controller之间如何正确的通信?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题▶

## 试题回答参考思路:

在 Angular 中, Controller 之间通信的方式主要有三种:

- 1) 作用域继承。利用子 Controller 控制父 Controller 上的数据。(父 Controller 中的数据要为引用类型,不能是基本类型,原因参见 AngularJS中的作用域 一文)
- 2) 注入服务。把需要共享的数据注册为一个 service, 在需要的 Controller 中注入。
- 3) 基于事件。利用 Angular 的事件机制,使用 \$on、\$emit 和 \$boardcast 其中,作用域继承仅限于上下级之间的通信,注入服务和基于事件的机制可以实现任意级别 的 Controller 通信。

## 🛅 面试题 48. 简述Angular CLI ?

推荐指数: ★★★★★ 试题难度: 初级 试题类型: 原理题▶

## 试题回答参考思路:

Angular CLI又称 Angular脚手架,用于快速生成项目或者组件的框架以提高效率。可以方便的生成angular app、component、service 等等, 并且可以通过参数,按照自己的需求去创建。可以说是angular开发必不可少的利器。

参考: https://cli.angular.io/

ng generate: 新建component、service、pipe, class 等

ng new: 新建angular app

ng update: 升级angular自身, 以及依赖

ng version: 显示anuglar cli全局版本、以及本地的angular cli、angular code等的版

本

ng add: 新增第三方库。会做2件事, 1) 基于npm安装node\_modules, 2) 自动更改配置

文件, 保证新的依赖正常工作

# 🛅 面试题 49. 简述关于Angular的依赖注入(dependency injection)?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题 ▶

## 试题回答参考思路:

依赖注入是Angular实现的一种应用程序设计模式,是Angular的核心概念之一。

依赖就是具有一系列功能的服务(service),应用程序中的各种组件和指令(derictives)可能需要服务的功能。 Angular提供了一种平滑的机制,通过它我们可以将这些依赖项注入我们的组件和指令中。因此,我们只是在构建依赖关系,这些依赖关系可以在应用程序的所有组件之间注入。

### 使用依赖注入还有以下好处:

不需要实例化,(new 实例)。不需要关心class的构造函数里需要什么参数
一次注入(app module通过Providers注入),所有组件都可以使用。而且是用同一个
service实例(Singleton),也就是说一个service里的数据是共分享的,可以用于组件间
数据传递

# 🛅 面试题 50. 解释关于Angular的编译,AOT和JIT的区别?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题 ▶

## 试题回答参考思路:

每个Angular应用程序都包含浏览器无法理解的组件和模板。 因此,在浏览器内部运行之前,需要先编译所有Angular应用程序。

#### Angular提供两种编译类型:

JIT(Just-in-Time) compilation

AOT(Ahead-of-Time) compilation

区别在于,在JIT编译中,应用程序在运行时在浏览器内部进行编译;而在AOT编译中,应用程序在构建期间进行编译。

显而易见,AOT编译好处多多,因而是Angular的默认编译方式。主要优点

由于应用程序是在浏览器内部运行之前进行编译的,因此浏览器会加载可执行代码并立即呈 现应用程序,从而加快了呈现速度。

在AOT编译中,编译器将与应用程序一起发送外部HTML和CSS文件,从而消除了对那些源文件的单独AJAX请求,从而减少了ajax请求。

开发人员可以在构建阶段检测并处理错误,这有助于最大程度地减少错误。

AOT编译器将HTML和模板添加到JS文件中,然后再在浏览器中运行。 因此,没有多余的 HTML文件可读取,从而为应用程序提供了更好的安全性。

### 🖿 面试题 51. 简述Angular双向绑定效率 ?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题▶

## 试题回答参考思路:

对于页面中需要绑定DOM元素极其多的情况(成百上千),必然会遇到效率问题。(具体还取决于PC、浏览器性能)。另外,脏检查超过10次(经验值?),就认为程序有问题,不

再进行检查。

可以采用如下方式避免

对于只用于展示的数据,使用单向绑定,而不是双向绑定;

Angular的数据流是自顶而下,从父组件到子组件单向流动。单向数据流向保证了高效、可 预测的变化检测。因而组件化也是提高性能的一种手段。

表达式(以及表达式所调用的函数)中少写太过复杂的逻辑

不要连接太长的 pipe(往往 pipe里都会遍历并且生成新数组, pipe 在anglarJS (v1) 中叫做filter)

变化检测策略onPush. Angular有两种变化检测策略。Default是Angular默认的变化检测策略,也就是上述提到的脏检查(只要有值发生变化,就全部检查)。开发者可以根据场景来设置更加高效的变化检测方式: onPush。onPush策略,就是只有当输入数据的引用发生变化或者有事件触发时,组件才进行变化检测。

NgFor应该伴随trackBy方程使用。否则,每次脏值检测过程中,NgFor会把列表里每一项都执行更新DOM操作

# 🛅 面试题 52. 解释什么是Angular的Module?

推荐指数: ★★★ 试题难度: 初级 试题类型: 原理题 ▶

#### 试题回答参考思路:

模块(Module)是一个我们可以对组件(Component),服务(service)和管道(pipe)进行分组的地方。模块通过导出或隐藏这些元素来决定其他模块是否可以使用组件,指令等。每个模块都使用@NgModule装饰器定义

## 🖿 面试题 53. 请描述Root Module和Feature Module的区别?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题 ▶

# 试题回答参考思路:

每个Angular应用程序只能有一个根模块(Root Module),而它可以有一个或多个功能模块 (Feature Module)。根模块导入BrowserModule,而功能模块导入CommonModule

# 🛅 面试题 54. 详细说明什么是Module 延迟加载(Lazy–loading)?

推荐指数: ★★★★★ 试题难度: 高难 试题类型: 原理题 ▶

### 试题回答参考思路:

当一个项目做得很大后,为了提高首屏加载速度,可以通过Lazy-loading,当访问到某些具体的url时,才加载那些不常用的feature module。

实现:正常创建feature module,修改路由配置。例如:

const routes: Routes = [

```
{
    path: 'customers',
    loadChildren: () => import('./customers/customers.module').then(m => m.CustomersModule)
    }
    l;
    这样,编译后,这个feature module就会是一个独立的js,只有当用户访问url (~/customers)时,才会向server端请求这个独立的js,然后加载、执行。
```

# ■ 面试题 55. 简述什么是Augluar 指令(Directive)?

推荐指数: ★★ 试题难度: 初级 试题类型: 原理题 ▶

## 试题回答参考思路:

指令(Directive)用于添加行为到已有元素(DOM)或者组件(Component)。同时, 一个元素或组件,可以应用多个指令

## 🖿 面试题 56. 简述Promise 和 Observable的区别?

推荐指数: ★★★★ 试题难度: 高难 试题类型: 原理题 ▶

## 试题回答参考思路:

首先新版本的anuglar是推荐使用Observable的(属于RxJS), 其次, 对于Observable对象,可以使用.toPromise()转化为Promise对象。

Promise,无论是否调用then。promise都会立即执行;而observables只是被创建,当调用(subscribe)的时候才会被执行。

Promise返回一个值; Observable返回0至N个值。所以Promise对应的操作符是.then(),

Observable对应的是.subscribe

Observable, 还额外支持map, filter, reduce和相似的操作符

Observable 可以取消, Promise不可以

# 🖹 面试题 57. 请问Angular 版本如何升级 ?

推荐指数: ★★★★ 试题难度: 中级 试题类型: 原理题▶

#### 试题回答参考思路:

Angular CLI提供了升级命令(ng update),同时,官网(https://update.angular.io/)也有升级指南。选择从哪个版本升级到哪个版本后,会给出一步一步的升级命令,直接执行就好

## 🖿 面试题 58. 简述Angular和Vue框架有什么区别?

推荐指数: ★★★★ 试题难度: 初级 试题类型: 原理题▶

### 试题回答参考思路:

Angular和Vue都是现代前端开发中最流行的JavaScript框架之一,它们有一些相似之处,例如都使用了组件化的思想和单向数据流等。但是,它们也有许多不同之处。

### 区别:

轻量级:相比于Angular, Vue是一个轻量级的框架,不需要很多其他东西就可以快速上手。

学习曲线: Vue具有更加简单的API,其学习曲线更加平缓,适用于初学者和小团队。而 Angular则较为复杂,学习曲线较陡峭。

性能: Vue在性能方面表现良好,在DOM操作时会采用虚拟DOM,这意味着它的渲染速度更快。而Angular的性能表现通常较差。

工具集成: Angular自带大量工具集成,包括绑定、路由、指令等, Vue当然也支持这些功能,但是依赖第三方库来完成。

#### 适用场景:

Angular更适合大型企业级项目,因为其拥有强大的工具集成和抽象层次,可扩展性非常强;

对于中小型应用程序或未来要做跨平台的web与app应用则更适合vue,由于vue更加轻量级和灵活,使用起来比较容易。

这两个框架都有自己的优缺点,选择哪个完全取决于项目需求和个人偏好。

## 🛅 面试题 59. Angular CLI如何实现Build?

推荐指数: ★★★★★ 试题难度: 中级 试题类型: 原理题▶

# 试题回答参考思路:

Build主要会做以下动作:

编译项目文件并输出到某个目录

Build targets决定了输出的结果

bundling 打包

生产环境的build还会进行uglify和tree-shaking(把没用的代码去掉)

ng build.

可以先看帮助:

ng build --help

针对开发环境, 就是用命令 ng build.

默认情况下,它的输出目录在.angular-cli.json文件里ourDir属性配置的,默认是/dist目录.build之后会看见dist里面有这些文件:

inline.bundle.js 这是webpack的运行时.

main.bundle.js 就是程序代码.

pollyfills.bundle.js 就是浏览器的Pollyfills.

styles.bundle.js 样式

vendor.bundle.js 是angular和第三方库

可以使用source-map-explorer来分析依赖,并且查看哪些模块和类在bundle里面.