Create project on git bash go to the directory you want it to be created and run

ng new project\_name

Import project to IntelliJ

File -> New Project from existing sources -> select the path from created project

Add Project to Github

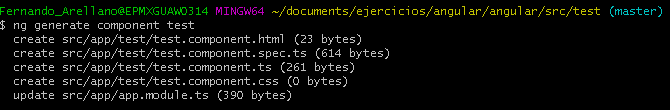
On Intellij VCS-> Import into Version Control -> share project on Github -> fill data and share

Add project into SourceTree

New Tab clicking -> Add -> enter the path where the project is locally

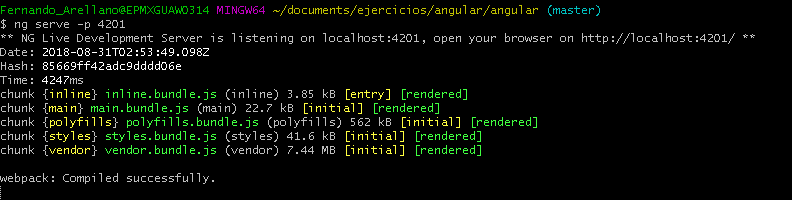
New component

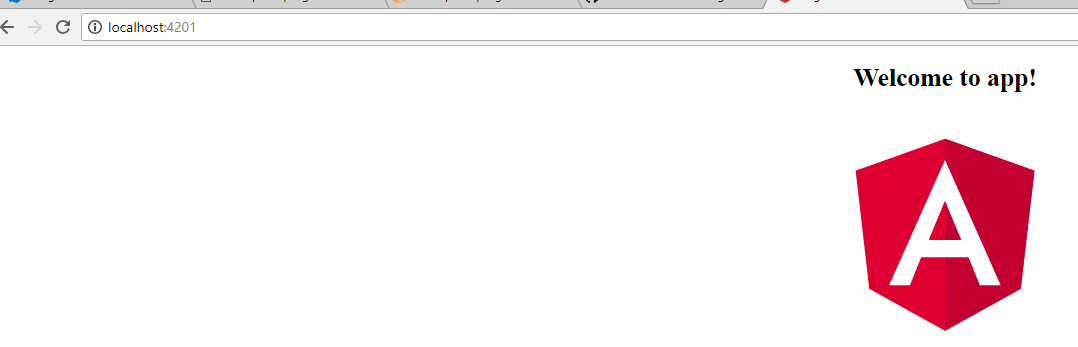
On git bash where the project is located ng generate component component\_name



Run Project

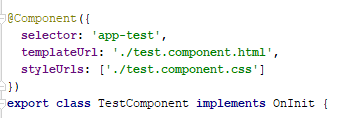
Ng serve -p port\_number





Test created component on app

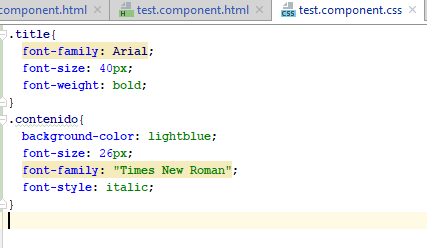
Open app component.html, tak the selector from the created component on the ts file and add it on the app html

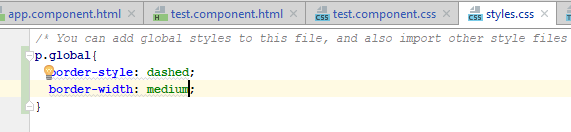


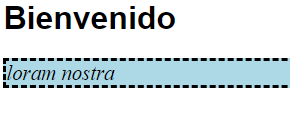


Add Css style to the component

On the css file, normally on the one for the component or there is also another global css







PRACTICA 2

Declare array on Ts

Arreglo: Array<string> = new Array();

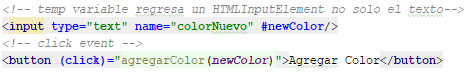
ngFor:



ngIf:

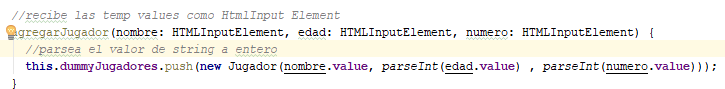


Temp Var:

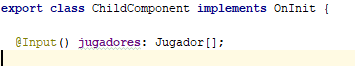




Parse values on tss:

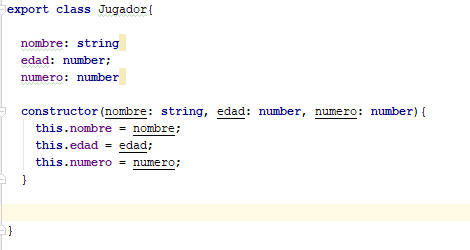


Pass value from parent component to child:



When a new jugador is added its updated and send to the child

Model example:



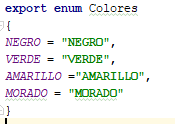
Writing property on String:

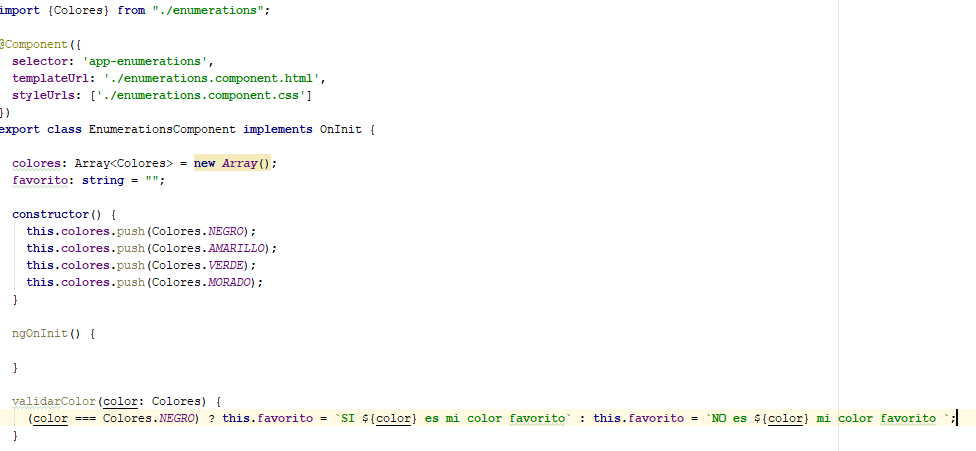




Enum

Import enum class.



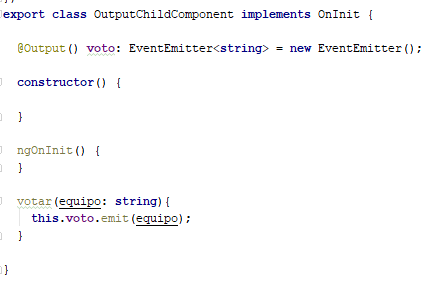


@Output, Event Emitter.

Sends an event from the child to the parent to pass or send values from child to parent



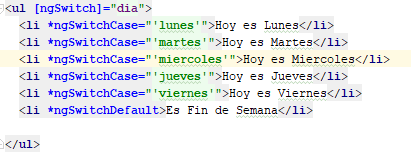
“When there is an event emitted for the voto var on the child component, execute the sumarVoto() method from the parent, pass the value to the parent



Voto property is set as event emitter and when it emits, then the sumarVoto event is executed.

ngSwitch

Choose from few options according from a var





ngClass

adds a class to an element If the condition is true



Adds class resaltarGanador if resaltar is true

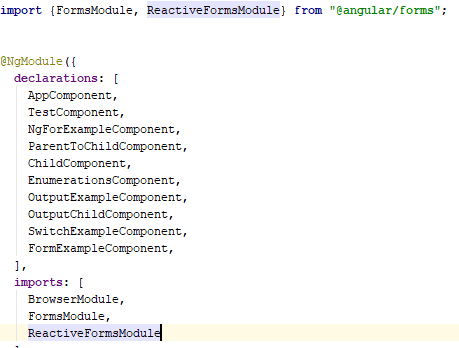
Forms

A FormControl represents a single input field - it is the smallest unit of an Angular form.





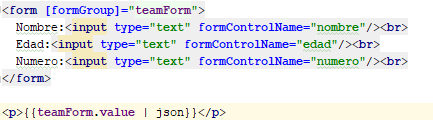
Added formcontrol dayControl and linked It to the input. dayControl is a formControl, to see its value use the value property



Needs to add the import of FormsModule and ReactiveFormsModule in order to use forms in the project

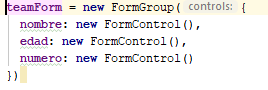
Form Group

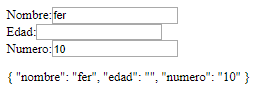
Most forms have more than one field, so we need a way to manage multiple FormControls



Printing team form value as json, if not entered will be null

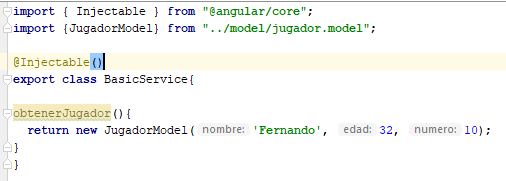
formGroup is linked to the teamForm FormGroup from ts file





Basic Service

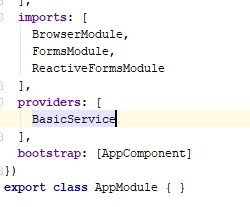
Create ts file for the Service and annotate it with @Inject()



Inject service on the class where its going to be used on the constructor.

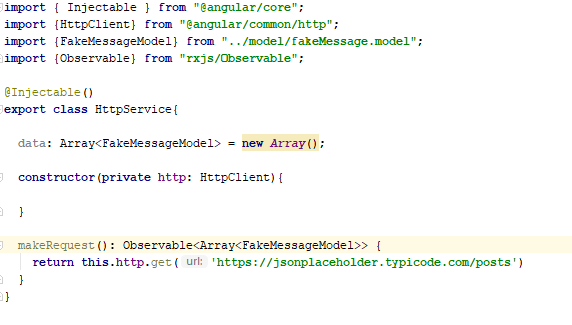


Add service on the providers on app module.

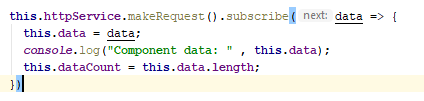


Http Service

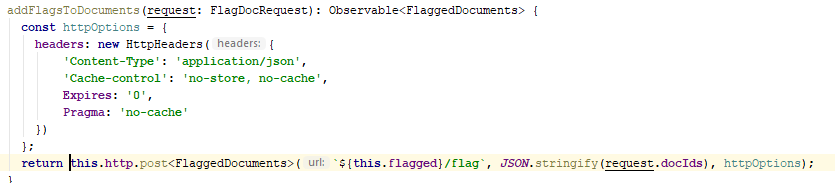
Using the HttpClient from angular/common/http you can make http calls which return and observable. (async calls).



Subscribing to observable and get the data



Call adding header params and body

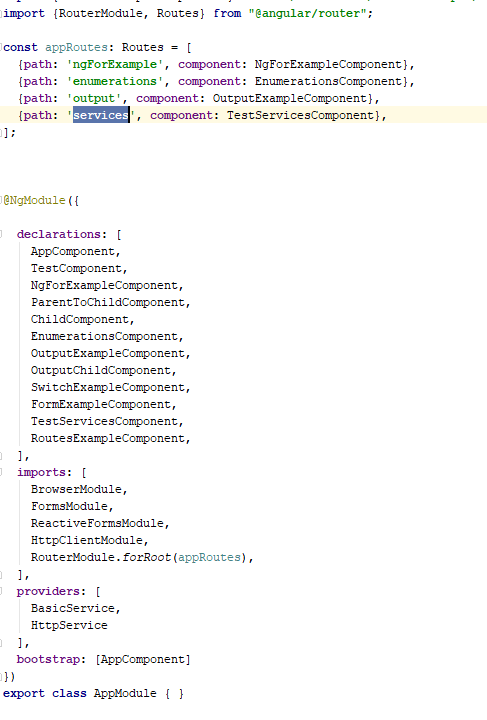


Routes:

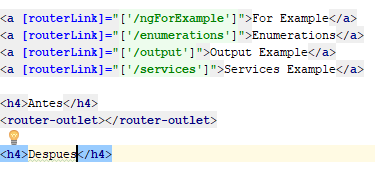
Allows to move among components on different links.

Config.

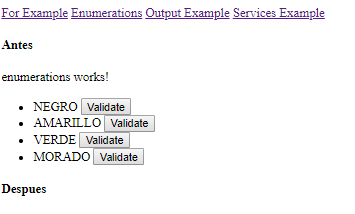
Needs to create the path routes and which component it goes to. Then import the routes.



Then create links which go to the created routes.



<**a [routerLink]="['/ngForExample']"**>For Example</**a**> creates a link that goes to the path ngForExample, from routes and goes to the corresponding component, then it replaces the <**router-outlet**></**router-outlet**> with the content of the component



Path Param,

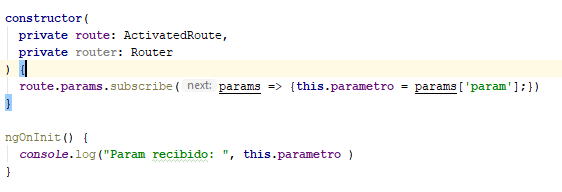
You can get path params from the link and use it on the component.





Declare the path param in the <a> with router link and modify the path of the route to specify it might have a param.

Then on the component you can go through the params received.



Print on console dummyParam (from the routerlink). Router type allows you to be redirected to a component and we could send the params we received.

39 this.router.navigate(['search'], { queryParams: { query: query } })

Observers

Allows to listen on a variable and execute a function when there are changes on it.



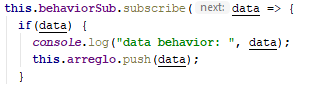
Subscribe to the service will return an observer, we will subscribe to it and when we get the data from the service then it will execute the function which is assigning the data received to the local var

Behavior Subject

Similar to observer allows to trigger the function sending a new value for the subject with the next value (it will replace the actual value of the behavior subject for the one sent on the parameter of next).



Initial value of the subject will be null.



Start the subscription on ng oninit so it will listen for changes.



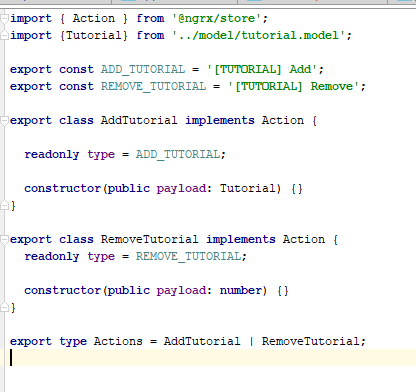
When fillSubject method is called then a new value for behaviorSub is sent which triggers the function on the observer.

Ngrx Example

Action: An action in Ngrx/store is two things:

* A type in the form of a string. It describes what's happening.
* It contains an optional payload of data.

What is going to happen with it depends on the reducer.



Consts used as strings for the action type.

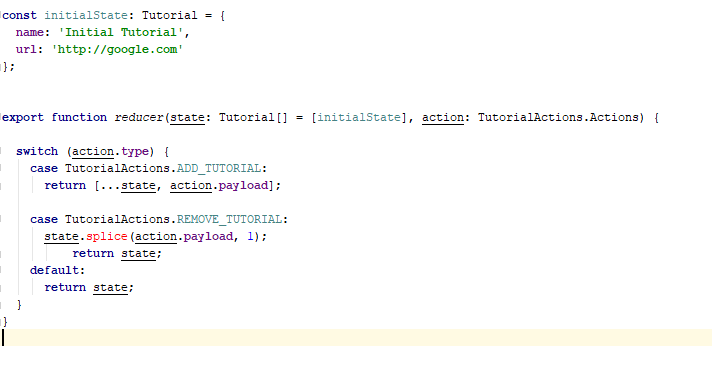
Export available actions which have to specify its type and a constructor (args are optional)

Export type actions then when imported in other file both actions will be available



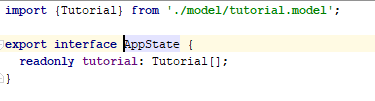


Reducer: A reducer is what takes the incoming action and decides what to do with it. It takes the previous state and returns a new state based on the given action.



Can have an initial state which will be return in case that reducer is selected before any action is executed.

State: state of the application, this will be return by the reducer and made available on the store.

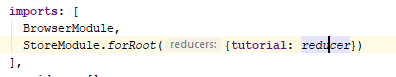


Store is injected to the component with generic type of the wanted state



Config:

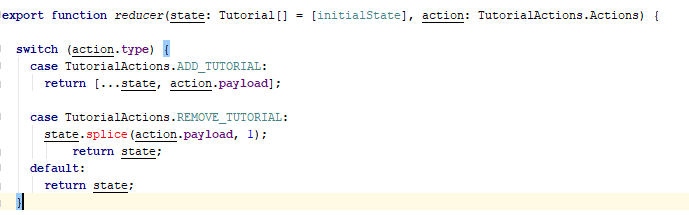




Import the reducers on app module. (name convention, tutorial.reducer.ts would be the file for the tutorial reducer added on the store module for root above.) tutorial should be selected on the store in order to call that reducer and get the reducer from the store:



Read from the store flow:



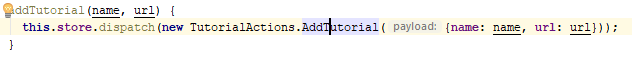
At start reducer is called, it will return the default value of the reducer (will send back the init value of the state)



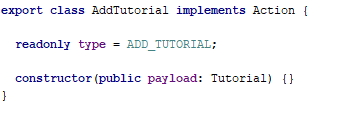
Component constructor calls the reducer tutorial with select, to retrieve info from the store.

Write to the store flow:

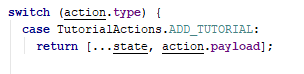
Component method called and it dispatch to the store



On the argument an new action is sent as argument, so it goes to actions to create that action type AddTutorial:

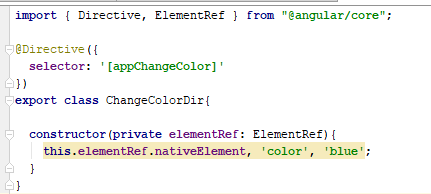


Then the reducer is called and will execute the according method depending on the action type:



New Directive

Annotate it with the @Directive and add a selector (this will be the property used in the components to use the directive ex: \*ngIf)

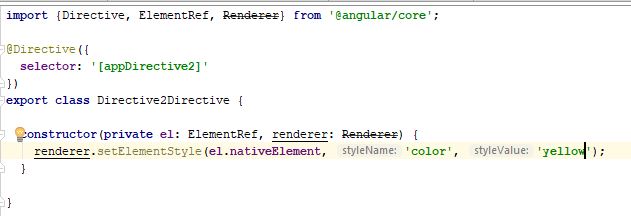


Use the selector on the component’s html in order to use it



Generate directive with angular cli

ng generate directive [name] generates a directive







@HostListener

Allows to listen to an event on the host element and then execute a function on it.





With this code, when inside the host element (the div using the selector of the directive in this cause) is mouseover onMouseOver function will execute. If click inside the div, onClick function will be executed.

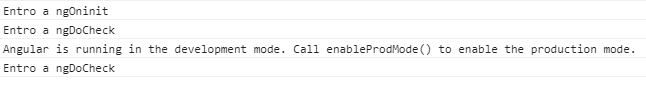
@HostBinding

Allows to set properties from the directive on the host element



Here when the mouse is over on the host element, then the style.border property from the element is modified with the value of border.

Lifecycle.



Keyup event

When a user presses and releases a key, the keyup event occurs, and Angular provides a corresponding DOM event object in the $event variable which this code passes as a parameter to the component's





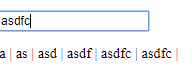
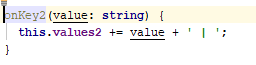
onKeyUp

passing the $event might be bad practice, other way to do it with template vars:



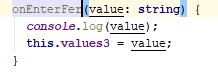
After each key is pressed box has a new value which is displayed (this will not be tied to a component, it will not work there unless you tie it up.

onKeyUp teeing to a component



onKeyUp.enter pseudo-event

only enter matters sometimes, bind to Angular's keyup.enter pseudo-event. Then Angular calls the event handler only when the user presses Enter.



After enter the value of box3 (everything inside the input) is sent and the function is executed, after that if you continue entering keys, they will be ignored until another enter is pressed

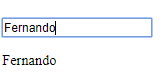
Blur

Executes the function when you click or move from the input



ngModel

The two-way binding is a combination of these two bindings; it gets the data from the component object to the view and sets the data from view to the component object.



Testing

The TestBed is the most important of the Angular testing utilities. The TestBed creates a dynamically-constructed Angular test module that emulates an Angular @NgModule.

The TestBed.configureTestingModule() method takes a metadata object that can have most of the properties of an @NgModule.

To test a service, you set the providers metadata property with an array of the services that you'll test or mock.



Most test suites in this guide call beforeEach() to set the preconditions for each it() test and rely on the TestBed to create classes and inject services.

Component test



Interview Questions

What is Angular.

Angular is a front-end or client-side Framework

Most important on Angular

Modules

Component

Template

Directives

Data Binding

Services

DI

Routing

What is transpiling in Angular

Transpiling is the process of converting the typescript into javascript

Which of the Angular life cycle component execution happens when a data-bound input value updates?

ngOnChanges is the life cycle hook that gets executed whenever a change happens to the data that was bound to an input.

Differentiate between Components and Directives in Angular 5.

Components break up the application into smaller parts (view and code); whereas, Directives add behavior to an existing DOM element. Component has template and styles Directives no

Structural Directives:

like \*ngFor and \*ngIf used for changes the DOM layout by adding and removing DOM elements.

Attribute Directives:

are used to give custom behaviour or style to the existing elements by applying some functions/logics. like ngStyle

What is the use of @Input and @Output?

communication of Angular Components, which are in Parent-Child Relationship; we use @Input in Child Component when we are passing data from Parent to Child Component and @Output is used in Child Component to receive an event from Child to Parent Component.

What is ng-content Directive?

The HTML elements like p (paragraph) or h1 (heading) have some content between the tags. For example, <p>this is a paragraph</p> and <h1>this is a heading</h1>. Now, similar to this, what if we want to have some custom text or content between the angular tags like <app-tax>some tax-related content</app-tax> This will not work the way it worked for HTML elements. Now, in such cases, the <ng-content> tag directive is used.

What does a router.navigate do?

When we want to route to a component we use router.navigate. Syntax: this.router.navigate([‘/component\_name’]);

What is ViewEncapsulation?

ViewEncapsulation decides whether the styles defined in a component can affect the entire application or not. There are three ways to do this in Angular:

Emulated: styles from other HTML spread to the component.

Native: styles from other HTML do not spread to the component.

None: styles defined in a component are visible to all components.

What are Services in Angular and what command is used to create a service?

Services help us in not repeating the code. With the creation of services, we can use the same code from different components. the command to create a service in angular, ng g service ServiceName

What is Routing in Angular 5?

Routing helps a user in navigating to different pages using links.

How to handle Events in Angular 5?

Any activity (button click, mouse click, mouse hover, mouse move, etc) of a user on a frontend/web screen is termed as an event. Such events are passed from the view (.HTML) page to a typescript component (.ts).

What is a RouterOutlet?

RouterOutlet is a substitution for templates rendering the components. In other words, it represents or renders the components on a template at a particular location.

Explain the usage of {{}}?

The set of brackets {{}} when used with an HTML tag, represent data from a component. This entire concept is called String Interpolation.

In how many ways the Data Binding can be done?

Data Binding happens between the HTML (template) and typescript (component). Data binding can be done in 3 ways:

Property binding in simple term is defined as updating the value of a certain variable in component (model) and displaying it in view (presentation layer) one way mechanism.

Event binding in simple term is defined as updating/sending the value/information of a certain variable from the presentation layer (view) to the component (model). This is just opposite from property binding.

Two-way binding is a combination of both property binding and event binding as it is a continuous synchronization of data/values from presentation layer to component and from component to the presentation layer.

What is the purpose of using package.json in the angular project?

With the existence of package.json, it will be easy to manage the dependencies of the project.

How is SPA (Single Page Application) technology different from the traditional web technology?

In traditional web technology, the client requests for a web page (HTML/JSP/asp) and the server sends the resource (or HTML page), and the client again requests for another page and the server responds with another resource. The problem here is a lot of time is consumed in the requesting/responding or due to a lot of reloading. Whereas, in the SPA technology, we maintain only one page (index.HTML) even though the URL keeps on changing.

ngModel

The ngModel directive binds an input,select, textarea (or custom form control) to a property on the scope

One-way Data Binding: [Component to View]

It will bind the data from Component to View using the following different ways.

Event Binding

Event binding flows or binds the data from an HTML element to a component.

Within parentheses on the left of the equal sign, we have the target event ("click" in this case) and on the right side, we have the template statements such as component properties and methods.

<button (click)="onClick()">Click me</button>

Two-way Data Binding

In simple words, two-way data binding is a combination of both Property Binding and Event Binding.

What does a Subscribe method do in Angular 4?

It is a method which is subscribed to an observable. Whenever the subscribe method is called, an independent execution of the observable happens.

Differentiate between Observables and Promises.

Observables are lazy, which means nothing happens until a subscription is made. Whereas Promises are eager; which means as soon as a promise is created, the execution takes place. Observable is a stream in which passing of zero or more events is possible and the callback is called for each event. Whereas, promise handles a single event.

What Is Bootstrapping in Angular?

main.ts is the entry point of your application, compiles the application with just-in-time and bootstrap the application.The Bootstrap is the root AppComponent that Angular creates and inserts into the “index.html” host web page.The bootstrapping process creates the components listed in the bootstrap array and inserts each one into the browser (DOM).

The bootstrapping process sets up the execution environment, digs the root AppComponent out of the module’s bootstrap array, creates an instance of the component and inserts it within the element tag identified by the component ’s.selector

Templates

A template is a form of HTML that tells Angular how to render the component

Data Binding:

a mechanism for coordinating parts of a template with parts of a component. Add binding markup to the template HTML to tell Angular how to connect both sides.

What Is the Angular Compiler? Why we need Compilation in Angular?

The Angular compiler converts our applications code (TypeScript) into JavaScript code + HTML before browser downloads and runs that code.

Explain the life cycle hooks of Angular 5 application

Angular 5 component/directive has lifecycle events, managed by @angular/core. It creates the component, renders it, creates and renders its children, processes changes when its data-bound properties change, and then destroys it before removing its template from the DOM. Angular provides a set of lifecycle hooks(special events) which can be tapped into this lifecycle and perform operations when required. The constructor executes prior to all lifecycle events. Each interface has a single hook method prefixed with ng. For example, ngOnint interface has Oninit method that must be implemented in the component.

Some of the events are applicable for both component/directives while few are specific to components.

ngOnChanges: Responds when angular sets its data-bound property which receives the current and previous object values.

ngOnInit: Initializes the component/directive after first ngOnChange triggers. This is most frequently used method to retrieve the data for the template from a back-end service.

ngDoCheck: Detect and act upon changes occurring outside Angular context. It is called when every change detection run.

ngOnDestroy: Cleanup just before Angular destroys the directive/component. Unsubscribe observables and detach event handlers to avoid memory leaks.

Component-specific hooks:

ngAfterContentInit: Component content has been initialized

ngAfterContentChecked: After Angular checks the bindings of the external content that it projected into its view.

ngAfterViewInit: After Angular creates the component’s view.

ngAfterViewChecked: After Angular checks the bindings of the component’s view.

The Angular offers two ways to compile our application code-

Just-in-Time (JIT) – JIT compiles our app in the browser at runtime (compiles before running).

Ahead-of-Time (AOT) – AOT compiles our app at build-time (compiles while running).

What are differences between Constructors and OnInit?

Constructors:-

1. The constructor is a default method runs when a component is being constructed.

2. The constructor is a typescript feature and it is used only for a class instantiations.

3. The constructor called first time before the ngOnInit().

ngOnInit:-

1. The ngOnInit event is an Angular 5 life-cycle event method that is called after the first ngOnChanges and the ngOnInit method is use to parameters defined with @Input otherwise the constructor is OK.

2. The ngOnInit is called after the constructor and ngOnInit is called after the first ngOnChanges.

3. The ngOnChanges is called when an input or output binding value changes.

What are Event Emitters and how it works in Angular 5?

Angular 5 doesn’t have bi-directional digest cycle, unlike angular 1. In angular 5, any change occurred in the component always gets propagated from the current component to all its children in the hierarchy. If the change from one component needs to be reflected to any of its parent component in a hierarchy, we can emit the event by using Event Emitter API.

In short, EventEmitter is class defined in @angular/core module which can be used by components and directives to emit custom events.

@output() somethingChanged = new EventEmitter();

We use somethingChanged.emit(value) method to emit the event. This is usually done in setter when the value is being changed in the class.

This event emit can be subscribed by any component of the module by using subscribe method.

myObj.somethingChanged.subscribe(val) => this.myLocalMethod(val));

Explain local reference variables, ViewChild, and ContentChild.

Local template variables in angular5 are used to refer HTML elements and use their properties to access siblings or children.

Let’s consider you have an input field named username.

<input type="text" required ... />

This HTMLInputField can be made available to the template using # symbol with a variable name say username.

<input type="text" #username required ... />

Now, this HTMLInputElement can be accessed from anywhere in the current template, for example, checking validation and showing appropriate message based on the validation rule. But, username HTML reference is not accessible in the component/directive.

To access this in the component, angular 5 provides @ViewChild decorator which accepts the local reference variable.

@ViewChild('username') username: HTMLInputElement;

ViewChild an element can be read after the view is initialized (ngAfterViewInit).

ContentChild is used to query the reference of the DOM within ng-content. Content Child are set before the ngAfterContentInit lifecycle hook.

Explain tsconfig.json file.

The tsconfig.json file corresponds to the configuration of the TypeScript compiler (tsc).

Explain package.json file.

All npm packages contain a file, usually in the project root, called package.json – this file holds various metadata relevant to the project.

This file is used to give information to npm that allows it to identify the project as well as handle the project’s dependencies.

Explain app.module.ts file.

This is root module that tells Angular how to assemble the application. Every Angular app has a root module class.

@NgModule — takes a metadata object that tells Angular how to compile and launch the application.

Imports — the BrowserModule that this and every application needs to run in a browser.

Declarations — the application’s component.

Bootstrap — this is the root component tells which component to run first.

How will you handle errors in Angular 2 applications?

Angular 5 applications have the option of error handling. This is done by including the ReactJS catch library and then using the catch function.

The catch function contains a link to the Error Handler function.

In the error handler function, we send the error to the console. We also throw the error back to the main program so that the execution can continue.

What is the difference between ActivatedRoute and RouterState in Angular 5?

ActivateRoute and RouterState both refer to Routing in Angular 5.

ActivatedRoute consists of the information about a route associated with the component loaded in outlet whereas RouterState represents the state.

ActivatedRouteSnapshort has old data When Route changes, ActivateRouteSnapshort has data from the previous route whereas the RouteState care about the arrangements and application components.

ActivatedRouteSnapchat to traverse all the activated routes whereas RouterState maintains the states.

What is a module?

Every application has a root module conventionally named as appModule. which provides bootstrap mechanism that launches the application. Typically It contains many functional module. Ngmodule is a container. An ngmodule defined as class decorated with @NgModule.

What are the router events ?

• NavigationStart

• NativationEnd

• NavigationCancel

• NavigationError

• RoutesRecongized

• RouteConfigLoadStart

• RouteConfigLoadEnd

What is NPM?

Node Package Manager NPM helps in installing packages for an Angular application.

Explain what a digest cycle is in AngularJS?

During every digest cycle, all new scope model values are compared against the previous values. This is called dirty checking. If change is detected, watches set on the new model are fired and another digest cycle executes. This goes on until all models are stable.

The digest cycle is triggered automatically but it can be called manually using “.$apply()”.

How can SPA be implemented in AngularJS?

SPA can be implemented with Angular by using Angular routes

Controller

In AngularJS, a Controller is defined by a JavaScript constructor function that is used to augment the AngularJS Scope.

Explain Directive scopes

Parent Scope: is default scope

Child Scope: If the properties and functions you set on the scope are not relevant to other directives and the parent, you should probably create a new child scope.

Isolated Scope: Isolated Scope is used if the directive you are going to build is self-contained and reusable. Does not inherit from parent scope, used for private/internal use.

What is $rootscope? How is it different from the scope

In Angular JS $rootscope and $scope both are an object which is used for sharing data from the controller to view.

The main difference between $rootscope and $scope is that $rootscope is available globally to across all the controllers whereas $scope is available only in controllers that have created it along with its children controllers.

What is the difference between $scope and scope

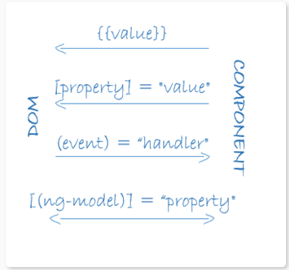
In Angular js $scope is used whenever we have to use dependency injection (D.I) whereas as the scope is used for directive linking.

How to access parent scope from child controller in Angular JS

In angular there is a scope variable called $parent (i.e. $scope.$parent). $parent is used to access parent scope from child controller in Angular JS.

Angular Docs

When you create a component, it's associated directly with a single view, called the host view. The host view can be the root of a view hierarchy, which can contain embedded views, which are in turn the host views of other components. Those components can be in the same NgModule, or can be imported from other NgModules.



Angular pipes let you declare display-value transformations in your template HTML.

<p>The date is {{today | date:'fullDate'}}</p>

Structural directives alter layout by adding, removing, and replacing elements in the DOM

Attribute directives alter the appearance or behavior of an existing element.

Ideally, a component's job is to enable the user experience and nothing more. A component should present properties and methods for data binding, in order to mediate between the view (rendered by the template) and the application logic (which often includes some notion of a model).

The injector is the main mechanism. Angular creates an application-wide injector for you during the bootstrap process, and additional injectors as needed. You don't have to create injectors.

A provider is an object that tell an injector how to obtain or create a dependency.

When Angular creates a new instance of a component class, it determines which services or other dependencies that component needs by looking at the constructor parameter types

When Angular discovers that a component depends on a service, it first checks if the injector has any existing instances of that service. If a requested service instance doesn't yet exist, the injector makes one using the registered provider, and adds it to the injector before returning the service to Angular.

When all requested services have been resolved and returned, Angular can call the component's constructor with those services as arguments.

When you register a provider with a specific NgModule, the same instance of a service is available to all components in that NgModule

When you register a provider at the component level, you get a new instance of the service with each new instance of that component. At the component level, register a service provider in the providers property of the @Component() metadata

Ej:

@Component({

selector: 'app-hero-list',

templateUrl: './hero-list.component.html',

providers: [ HeroService ]

})

In Angular, the component plays the part of the controller/viewmodel, and the template represents the view.

The <script> element is a notable exception; it is forbidden, eliminating the risk of script injection attacks. In practice, <script> is ignored and a warning appears in the browser console.

Prohibited in interpolation:

assignments (=, +=, -=, ...)

new

chaining expressions with ; or ,

increment and decrement operators (++ and --)

In an event binding, Angular sets up an event handler for the target event.

When the event is raised, the handler executes the template statement. The template statement typically involves a receiver, which performs an action in response to the event, such as storing a value from the HTML control into a model.

The binding conveys information about the event, including data values, through an event object named $event.

As of Typescript 2.0, you can enforce strict null checking with the --strictNullChecks flag. TypeScript then ensures that no variable is unintentionally null or undefined.

The type checker throws an error if you leave a variable unassigned or try to assign null or undefined to a variable whose type disallows null and undefined.

The hero's name is {{hero!.name}}

When the Angular compiler turns your template into TypeScript code, it prevents TypeScript from reporting that hero.name might be null or undefined.