

Angular

Getting Started



- I) create input and <ps> inside the input write (ngModel) j="name" <p>ss name </p>
and also what needed in the controller. does it work? ✓ change ✓
- II) install bootstrap: npm install -- save bootstrap@3 - let angular know of that
run again the server. check in chrome: sources → styles.css ✓

The basics

- 1) Go to main.ts change the parameter value of the fn to something that doesn't exist ✓
- 2) create component manually. Inside the component define selector, templateUrl, ✓
- 3) in the component code <ps> it works! <ps> (HTML File) ✓
- 4) delete the html file and do everything in the ts file ✓
- 5) create a style in a css file for your template. ts. delete it and apply a different style inside the ts file ✓
- 6) change the selector to an array. it breaks the app. How can u fix it?
change it to look like a class. How can u fix the broken code? ✓

Data Binding \Rightarrow Communication between

communication between business logic (TS code) to view (template, HTML)

- 1) property binding: create a button that deactivate after 3 seconds, ✓
a timer that counts from 5 to zero and when it reach 0, the title disappear ✓
- 2) event binding: create a button. When a user click the button he change randomly
the color of the header X
- 3) Create an input: Inside the input create a function that update a variable in
the TS. present the content of the variable in the HTML. Result: with every
letter you type, you see the same letter displayed inside the <ps>. use the \$event
for this to work u need inside your method to cast the event <HTMLInputElement>
- 4) Do the same with data binding. ✓

Directives

- 1) have a button. When it's click \rightarrow an element is appearing ✓ use ngIf
- 2) the element appeared out of nowhere. Implement else ... if true \rightarrow show this element
else show another - use a local reference (#) \rightarrow use on container ✓
- 3) change an element style dynamically with ngStyle ✓
- 4) add margin border background color if a condition is fulfilled using ngClass ✓
- 5) display a component as many time as you have element in an array ✓

6) do the same using index

Create Course Project

- 1) create the components ✓
- 2) create and design your header using <nav> <a> ✓
- 3) create a model for the recipe with name, description, photoUrl . create a constructor ✓
- 4) in the Recipe-list create an array with 3 Recipes object . code the recipe list component with dumb data for the name and description ✓
in the recipe component let the recipe list col-md-5 and the details 7. put them inside the same row ✓
- 5) now get the data dynamically from the [] ✓
- 6) code the recipe details ✓
- 7) in the shopping list create a list of ingredients . Create an ingredient model under a shared folder with name, amount without creating an instance var. ✓
- 8) fill the array with 5 ingredients and display them in the html dynamically ✓
- 9) finish the shopping list future ✓
- 10) finish shopping details ✓

//Angular

Components & Data binding

- 1) DataBind an element from parent → daughter ✓
- 2) —→ daughter → parent ✓
- 3) in a parent element customize css property. is it applied to the child? change this with ViewEncapsulation.None ✓
- 4) pass input to the ts with one-way binding. get its value with local ref (#) using a button. try to use it without a button / work only with inside HTML
- 5) do the same with @ViewChild - what's the difference? use the attr nativeElement.value to get access to the value ✓
- 6) inside your app content put a <ps> nested inside one of your component. is it displayed? put inside this component <ng-content> in the beginning in the end in the middle. test ✓
- 7) create log(x has been fired) when x = method name. do it for: constructor, ngOnInit(), ngOnChanges(), ngDoCheck(), ngAfterContentInit(), ngAfterContentChecked, ngAfterViewInit, ngAfterViewChecked, ngOnDestroy() → copy paste them to different components
try to access element by @ViewChild with ngOnInt, try with all of them at which point you have access to it?
- 8) Implement the @ViewChild but in a child element does it work? Fix it if the <ng-content> must be present for it to work

Directives

- 1) create a Directive. DIY folder and a ts file - your directive file / make the necessary adjustments to make it directive - pass an config obj with selector do what it needs (constructor, injection, ngOnInt, telling angular that it exists). create a directive the highlight with green marker
- 2) create a directive (manually) call it highlight ✓. do the same with Renderer2 ✓
- 3) create a dynamic directive that changes when the mouse hovers. do it with @HostListener+method event name ✓

- 4) @HostBinding → do the same with host binding. Host Binding: I can bind to any element property that the directive is sitting on
- 5) create a dynamic directive that gets values from outside
- 6) create a structure directive to show something if false
- 7) use ngSwitch to present different <ps> according to a value in your ts
- 8) create a directive in the recipe-app to activate the drop-down menu →

you do it by attaching the class open and the dropdown will open

Services

1) Create an emit event in your service. One component will emit the event, another will catch it (subscribe)

Routes

1) Create a <a href> that redirect to a component. Use relative paths "... " or ".../x" and absolute path "/x" → routerLink = "..."

2) Play with routerLinkActive - set it to active on a class with routerLink="...". What happens when you navigate. Take it off. How do the tabs look like when you navigate? Solve the first problem with "/" . Fix it with [routerLinkActiveOptions] = "[exact: true]"

3) Snapshot: good for the first initialization but if the url will change I will never know cause I don't observe the change

route.params: it's an observable. I will subscribe to it and react

observable: get 3 parameters. At that point we will use the first

4) Define a path with a dynamic parameter in the app.module

5) Try to fetch it, print it to the console

6) Create a button. When you click it will take you to `www/example/edit?allow=yes#ok`

7) Get a parameter from the url. For example `user/1`, display it. Change the url. Do you see the change in the console? Fix it.

8) Create a service with `login=false`, a `login` method, `logout()`, and an `auth()` which create a promise and set timeout and return the `login` var.

9) Pass data (static) to a component and present it in your page. Do it in the `ngModule`

10) Difficult: Pass data from `app.component` in `ngModule` to a component

Routing

Angular

Routing in HTML

- Setting up:
 - an array with routes objects { path: component: ? }
 - imports: RouterModule, forRoot(carry name)
 - in app.component.html => <router-outlet>

1291 Navigation Bar routerLink = "/user" => user
[routerLink] = [" /user", 'x'] => user/x } absolute path
routerLink = "user" => /currentAddress/user } relative path
↓
file: inside nested routes

1301 .. /path => go back one level from the loaded component
'/path' = absolute. Will get appended to the root domain
'/path' & 'path' = will be added to the current path

1301 class = "active" to set it dynamically I will use routerLinkActive = "classname"
with the route "/" I need to add [routerLinkActiveOptions] = {exact: true}

Routing in TS

131 Navigation inside TS: 1) inject Router 2) use the method navigate (an array)
relative path: 1) inject ActivatedRoute route 2) pass an object to navigate()
=> this.router.navigate ([{"id": 1}], {relativeTo: this.route})

1331 passing parameters to routes: in the app.module, define the path as "user/:id"
V = user/try V = user/something user/1=V ↪ if a link is generated it will take the value

1341 Fetching route parameters: 1) inject ActivatedRoute 2) route.snapshot.params['id']
listening and Fetching route params: route.params.subscribe()

key = value #

1371 passing query parameters/Fragments in HTML [queryParams] = " { key: value } " ? key:value
fragment = "name" # name

the same in TS: navigate ([...], {queryParams: {key: 'value'}, fragment: 'name'})

1381 Fetching query/Fragments: 1) statically, only once: route.snapshot.queryParams → object
route.snapshot.fragment
2) reactively: route.queryParams.subscribe
route.fragment.subscribe

Children in the app module: in case I have a path with a children

1) I add a children: [{path: without the parent name, component}] to the parent

2) in the html I position <router-outlet> in the parent component where I want the children to be loaded.

How do I know that I need to nest paths as children? if I have repetition of the path name for ex: path: 'user' ... path: 'user/:id' path: 'user/:id/edit'

iii) WildCard routes: in the app.module, I can use the following wildcards

'*' : should be the last = everything that you don't recognise

iv) passing parameters between urls: route.navigate(['where'], {relativeTo: this.route, queryParamHandling: 'preserve'}); → 'merge' for adding more parameters

Guards

x = user who don't have what it takes to access a certain component

canActivate: preventing x from accessing a path

canActivateChild: preventing x from accessing child path // product/3 for ex

canDeactivate: preventing anyone from leaving a path (close the browser, changing address)

I want to ask the user: Are you sure?

Observable

rxjs

Observable: A data Source

Stream, time line → events/data emit by the observable

Observer: my code, the subscribe function. What I want to do with the data package
data package: 1) Data 2) Error 3) Completion

Working with observable

On one of your components in ngOnIt create an observable that emit an event every 1000 second, manage the data (seconds) that it emit by printing it to the console
navigate to another component. Did the observable stop? make it stop but before go back to your emitting component. look at the console

Custom observable

create a new observable with .create(), import Observable

create takes a function with an observer arg // or data - whatever you want to call it and the you should define your source for the data (mouse click, http request, interval...)
I do it by calling the method next()

- 2) if something happen (condition) throw and error with the error()
- 3) try to catch it in your observer
- 3) finish/end your observable after 5 seconds

operators pipe()

{ in his website → learn JS → understanding rxjs}

instead of doing the treat. in the subscribe, I do it before, in an operator.
it's done by calling the pipe() and inside it do the treatment and return it after the treatment.

- 1) use operators instead of subscribe to implement your treatment

Subjects: Replace event Emitter inside a service

replace event Emitter

take an older version of the app and replace event emitter with subjects

Subject won't replace an event Emitter that is attached to @output()

HTTP

Use In Place

1) Create a firebase project and open in intelliJ the code for this section

import HttpClientModule to the imports[] to use it I must inject HttpClient

Post

I call the method post() on the http instance

post() takes couple of args → 1. url 2. body

3: Object → Headers → Params

* if I use Firebase: the url 'http: ... com/products.json' → MUST firebaseRef db.ref()

body: I will put an obj, Angular will auto transform into json

subscribe: if I don't subscribe, the request won't be sent

1) Create a test data base. When you click on a button in Angular, a record is added to your database

Get

1) Get the data from the data base. do some treatment with operators

2) Modify your html to accept the data and display it in the HTML

3) Search google for loading indicator, or a css thing. If you find it implement in your code to show it while fetching data. Use slower connection to test it

Delete

1) delete all of your records

Setting Headers

1) To the post method add some headers. Header is the 3rd arg of the http.method and it's an object {

headers: new HttpHeaders({})

}

'key': 'value'

Adding Query Params

{ params: new HttpParams().set('...', '...')}

inside the object of the 3rd arg. Create a post method to the films api and add the params of the search

http

Response Object

- 1) get the response status. Do it by accessing the whole response obj;
So far you dealt only with the response Body
it is done in the 3rd param of the \$httpmethod. {observe: 'response'}
- 2) display your event from post/delete. You do it by observing the event
{observe: 'event'} instead of map
- 3) Angular will try to convert the json to an obj. if I want to tell it
don't do it, keep it as a txt, blob etc => {responseType: 'text'}

Forms

- No submitting with the button in the old-fashion way. Instead: I will use Angular powerful toolkit - an object^{**} that represent the form template! Inside the template, angular analyze, give me the input reactive: inside the typescript and template, and link them
- * old fashion way: <form action="..."> / having action attribute
 - Misplaced import FormsModule
 - * <form>: it tells Angular to create the Form Object but it doesn't detect the inputs auto I need to tell angular what part of my form I want to work with => if Model <input ngModel> tell to angular - this input is a control of my form, but what is its name? <input ngModel name="..."/>
- Create + Submit**
- ① Create a form with name, email, family name, telephone. register ✓ p.30 n.
 - ② Create an ^{onSubmit()} method and represent it in the HTML (not on the button), test with console.log ✓
 - ③ console log the form object, use local ref in the html. the object is of type NgForm ✓
 - ④ Select the form with local ref (without a value) print it to the console while declaring its type as ElementRef and by HTMLElement - is there a difference? - No ✓

Form object

- 1) Submit the form untouched without input, check for dirty and touched attr
enter 1 input, check them again, leave it empty again, click on one of the inputs but keep it empty, check now touched, untouched ✓

Validation & State

must be filled: required
for emails: email

- 1) enter invalid email address ^{chks} obj. in the template under the email validation directive, enter invalid email address, click the object.valid. enter a valid address do the same, but check the "elements" in the developer tool ✓
- 2) disable the form button if the form is invalid. Write CSS to color all the invalid input in red(border attr). B) Show a message under the email input in case it's invalid, do it only after the user has tried to enter email ✓
- 3) Create a select with 2 options. let 1 of the option being shown as a default (for this I need that every option will have its value ✓)

- With the same logic prepopulate the email input with default email ✓
- 4) add a user name (choose a username) and display a cps that says the username xxxx is already taken / or it's too short ✓
 - 5) Group the user name, family and email under the same group in the template with ngModelGroup. Create a cps that will indicate if all the group is invalid ✓
 - 6) add an error "email is not valid" in case the email is not valid & touched ✓

ngModel, [ngModel], [ngModel]

property binding: 1 way binding

- 1) create a drop down menu and make the 2nd option the default by using [ngModel] ✓

2 way binding

- 1) create a text area for an answer, and display it as the user type it ✓

also: No value: to tell to angular that this is a form control

1 way binding:

Setting Form Values: Set value, form.patchValue using myForm: NgForm

- 1) create a button, suggest user name, in its click method set the username input to the first name in an array, clicking again array[1], again..array[2]
Do it with the method setValue() - * the type is NgForm, not Form ✓

Set Value: receive an object that represent the form completely, you define all controls ✓

- 2) Do the same with patchValue. You can define only 1 control. form

Summary: setValue: to set value to all of the controls in the form ✓

patchValue: to set only to certain controls. this method run: formObj.form.patchValue()

Submit

- 1) create a user object with the needed att. on submit, fill this object with all the user input. b) create a parameter/variable that will be false (name → submitted). in the html create ngIf and show all the object values in case that submitted = true / (this.form.value['nameCont'])
- 2) RESET: reset the form. there is a method for it, by passing to the object you created in 1)

Reactive Form

setup, submit, Access, Group, Custom Validators, FormArray, Reacting to change
Setting/Patching

Setup

- 1) Create a var that will hold the form. type: FormGroup ✓
- 1B) in the app.module import ReactiveForm ✓
- 2) Define your form in your ngOnInit By creating a new FormGroup Obj.
As a parameter - the architecture of the form ✓
- 3) Create inside the form config - All your controllers → new FormControl()
 - 1arg: initial state/value
 - 2arg: Validator
 - 3arg: Async Validator ✓
- 4) for the sex, set a default (male) ✓

Connect

- 5) connect your TS form to your HTML ✓

Submit

- 1) submit the form and print the form object ✓
- 2) create an object and print it ✓

Validators - built in

- 1) Add one validator (validators.required), delete it, add to html. does it work? yes
- 2) add 2 validators (required+email) ✓

Custom

a validator is just a function that is exe when Angular check the validation of a control, so creating a validator - is creating a function

returns
function name(control: FormControl): { [s: string]: boolean } {
if logic... return { 'string': true }
else return null }
Error code that I write
or omit it (not if control -> sk : obj, it's null, it returns obj)

- 1) create a validator that check if a user name is not a part of user name already exist, inspect the element in the browser to see that it's invalid ✓
- 2) add it to the control and test it. if it doesn't work use bind(this) ✓
- 3) create a cps with error message in case u get Error: 'string' is not a valid value

Access Controls from HTML \ TS

- 1) Create a message please enter a valid email
- 2) Do the Grouping control exo and come back to here after. Create a message that will be shown if the user has its form group invalid

Form Group

Our Form is of type form Group, but it can have another form group inside of it, a nested form Group

- 1) Create a nested form group that will include 2 controls (TS)
- 1B) take care of the HTML. You need to modify and define something
- 2) return to the previous exo and do 2)

Adding Controls Dynamically: Reactive via FormArray

- 1) Create a reactive control, the user push a button and he can add a control, an input for example. instead of FormControl it's the FormArray object that I need
- 2) addControl method: add the control. it is done by accessing the control and since it's an array, pushing the new control. Add also validator required
- 3) in the HTML position it inside your code A) on a parent div
· declare it `<formArrayName="...">`
B) create a form element
C) above it create a `<div FormGroup>` and loop the control*, extract the index
D) in the form element define `[formControlName] = "index"`
* you may have trouble here, go see the last mins of the lecture

Status / Value Change

On the form variable I have 2 observables : valueChanges, statusChanges
I can apply them on the all form or on a controller

- 1) Subscribe to value changes, create a var and assign the value in the subscribe to this variable. display in html
- 2) Subscribe to status change and find something to do with it

setValue(), patchValue(), reset()

work the same as the template form