

SignupAdmin  $\rightarrow$  role = "admin"

1 --

1 --

✓ --

Signup

app.js

let admin = { --, ---, -- }

admin.save ( )

res. user  $\left\{ \begin{array}{l} p_{N, \dots} \\ p_{W, \dots} \end{array} \right. \text{role: Admin}$

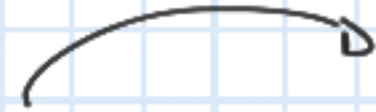
comb. log (res. user)

if (res. user. role == 'admin')

h

(path == '---')? 'admin' : 'user';

3r  $\frac{Q}{\checkmark}$



Test praghiye : 40 min

Validat<sup>o</sup> Praghiye  
12 min

Service : Post (@, obj)

"http://localhost:3000/..." {id: ...}

Suppress: —

Modificational

get All

Search by

get by ID

Add ✓

→ Delete (id) --- / \${id}

app.post("/", (req, res) => {

let id = req.body.id

}

Modifical<sup>o</sup>

Print(@, obj)

Post(@, obj)

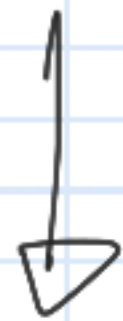
app.post('\_', (req, res) => {  
 let newobj = req.body;

})



Get by name & Age  
"Ali" 30

http.get(-- /\${name}/\${age})



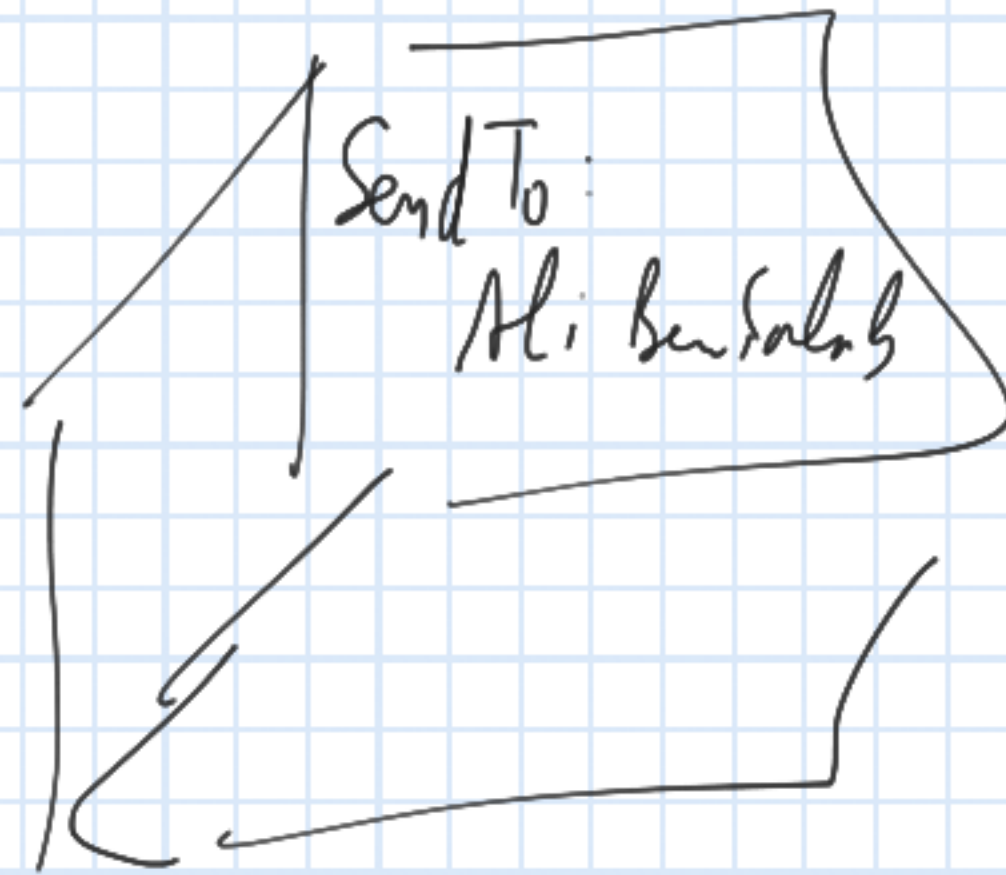
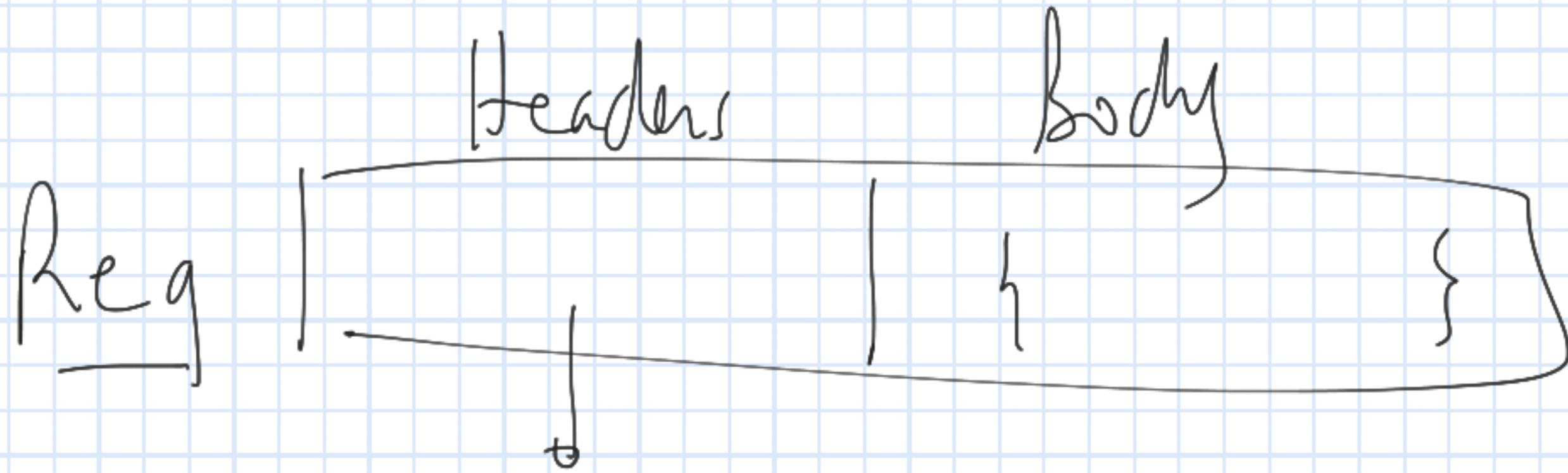
let x = ref.params.name  
let y = ref.params.age

http.post

(@, {name: "Ali",  
age: 30})

User.find({name: ref.body.  
name,  
age: ref.body.  
age})

User.find({name: x, age: y})  
den x attributs de la collect.



@ Input ( )

Passage des données  
du Parent  $\rightarrow$  Child



@ Output ( )

Child  $\rightarrow$  Parent

@ Component  
@ Pipe

Décorateur



$\rightarrow$  f spécifique qui  
donne des caractéristiques

@ — • classe  
• attribut  
• méthode



@Input()

matchInput: any;



attribut

---

@Component {

selector: - -

templateUrl: - -

styleUrls: - - -

}

export

class

MatchComponent

✓

@Pipe ( { name: ----- } )

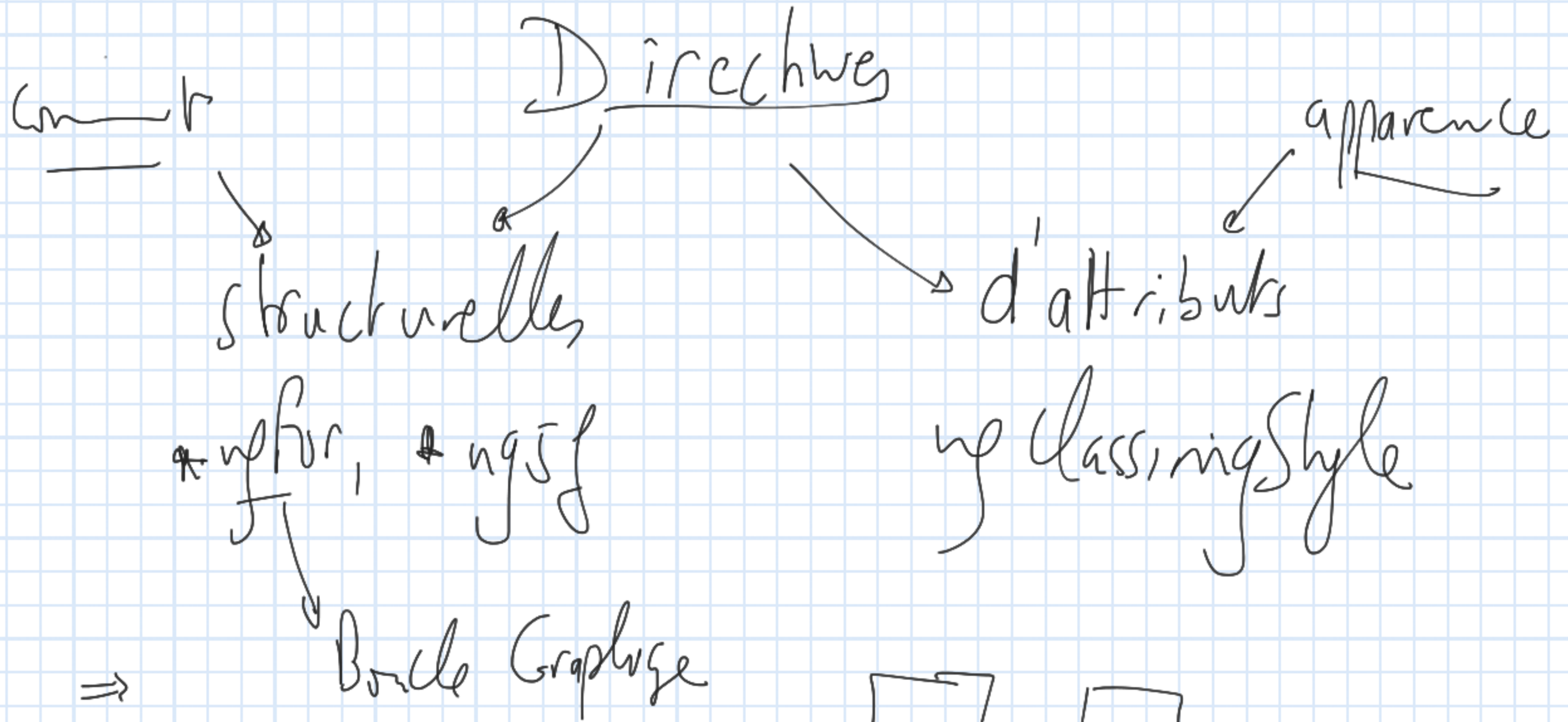
export class XPipe {  
 transform ( ) {

ngOnInit ( ) {

}

}

}



classe qui permet de  
ou le compt des élts du DOM

modifier l'apparence

$\langle hr \rangle$      $\dots$      $\langle /hr \rangle$

app. module.rs  
  declarations  
  imports  
  providers  
  bookgraph

$T = [ \hookrightarrow ]$

$V = [ \hookrightarrow, \hookrightarrow, \hookrightarrow ]$

app-routing.module.ts

↓  
Déclaratif des paths.

---

router-outlet: — balise < injecte

dynamiquement les composants

selon le path



index.h

$\langle \text{app-root} \rangle$ : selector du G-r

récupérer —  
valeur d'un input

RF

TIDF

for Control Name

ng/model

Principal



App Comp

RF  
f-Group  
f-Control Name  
(click)

TDF  
w/for—  
w/Model  
w/Submit

\$ \omega \quad g \quad p \quad N\_{-} \text{life}

Binding :

$-\cdot h h l$

$-\cdot h s$

event Bindg

$\{ h \} \} \leftarrow$

$[ ] \leftarrow$

$[ ( ) ] \leftarrow$

La différence entre `[]` et `{ }`

`x = "btn btn-danger";` `<button [class]="x"> -----`  
`<button class={x}> ---`  
`y = "All Products";` `<h1>{{y}}</h1>`  
`<h1>[y]</h1>`

Add

→ rks

→ hmd

user: any = { };

①

②

<input

[(ngModel)] =

"user.fName"

Al;

Plus user = Faire retour  
{ firstName: "Ali" } l'objet user  
(rempli)

Add



Editor

vs

— help

AppService.gtk --

(this.user = {  
  name: "Ali"  
  email: "a@a.a"  
})

→ [InfModel] =  
  username

| Ali Bu  
| —

this.user =

{  
  name: "Ali Bu",  
  email: "a@a.a"  
}

| Editor

Reactive For-

this user = { fnc: "Ali" }

<input fnc="ControlName:"

[(ngModel)]="fnc">

→ "user.fnc"

<input fnc="fnc" = "fnc"

[(ngModel)]="user.fnc">