Sharemycode xpg

Installer

Nodejs et npm nodejs.org

npm install -g @angular/cli

----Créer une app

ng new nom-app

* Servir l’app dans le nav en mode développement

Ng serve -port si on veut choisir un autre port que celui par défaul -4200)

Ng serve -o 🡺 compile et lance notre appli

---Buil pour la prod

Ng build

Compiler l’application en fichiers :

* Index.html
* Main.js, polyfills.js

--- Les concepts Essentiels

Le module principal

Src/app/app.module.ts

C’est simplement un fichier déclaratif

* Des class des components
* Des class des pipes
* D’autres modules
* …

src/index.html => point d’entrée

src/style.css => style global

le data-binging

un component c’est :

* La logique et les data : app.component.ts
* => la vue : app.component.html

La syntaxe d’interpolation :

* Permet d’afficher des données dans la vue
* Met en place la détection de changement

.ts .html

Name=’Fred’ <h1>Hello {{name}}</h1>

La syntaxe de property binding

.ts .html

imgUrl= ‘chien.jpg ‘ <img [src]=’imgUrl’>

La syntaxe d’évenement

.html

<button (click)="launchMethod()">Clickez ici</button>

.ts

launchMethod(){

//faire un truc

}

Les directives :

* Component <app-root></app-root>
* Les directives d’attribut
  + ngClasse <h1 [class.selected]=’false’></h1>
  + ngStyle <h1 [class.color]=’redColor’></h1>
* Les directive de structure
  + \*ngIf => permet d’inclure ou supprimer un template HTML

<img \*ngIf=’false’>

* + \*ngFor => permet d’itérer pour afficher des listes de balises Html

<UL>

<li \*ngFor=’Let item of arr’>{{item}}</li>

</ul/

Créer des components :

Syntaxe : ng generate component [folderName][name]

Le répertoire est optionnel

Exemple : ng generate component monrepertoire/card

(on peut écrire en abrégé : ng g c card)

Ca va générer :

* card.component.html
* card.component.css
* card.component.ts
* @Component({

Selector : « app-card »,

})

Class CardComponent{

}

Le pipes pour formatter l’affichage dans un fichier .html

{{VaraibleAAfficher | nomDePipe}}

Nom de pipes natifs  :

Uppercase, lowercase, titlecase

date :[format] exemple : {today | date : dd-MM-yyyy}

currency :[format] exemple : {price | currency}

decimal

json {{movie | json}}

On peut créer ces propre pipes

Synthaxe : ng generate pipe [nom]

Exemple : ng generate pipe pipes/format-time (pipes = repertoire pas obligatoire)

Cela génère : (hour, minutes et …. Sont les attributs à modifier

Transform(value) {

//doit retourner une string

}

@Pipe({

  name: 'formatTime'

})

export class FormatTimePipe implements PipeTransform {

  transform(value: number): string {

    let hour = Math.floor(value/60);

    let minutes = value%60;

    let formattedMinutes = minutes <10 ? '0' +minutes : minutes;

    return hour+'h'+formattedMinutes+'mn';

  }

}

* Le pipe est utilisable dans les vues .html

Pour Partager

@Input() movie

@Output event

Précision sur l’authentification Stateless (sans etat)

Rappel de la communication :

