Hajek Valentin ANGULAR TP5 info2

1a) Réalisation et création du composant

Dans src/app

ng generate component hero-detail

On a généré : un fichier css pour les styles du composant crée, un fichier html pour les template du composant, un fichier typeScript avec un composant de class nommer : HeroDetailComponent, un fichier class test file pour HeroDetailComponent

b) template de HeroDetail

On remplace le code actuel par : on crée l’html du HeroDetail

<div \*ngIf="hero">

<h2>{{ hero.name | uppercase }} Details</h2>

<div><span>id: </span>{{hero.id}}</div>

<div>

<label>name:

<input [(ngModel)]="hero.name" placeholder="name"/>

</label>

</div>

</div>

c)

On import le symbol Hero

import { Hero } from '../hero';

dans la classe heroDetailComponent :

Les propriétés de hero sont ajouter, décorée avec le décorateur @Input() qui permet la liaison

@Input() hero: Hero;

d) Rôle de parent/enfant entre les deux composants, le parent HeroesComponent contrôle l’enfant HeroDetailComponent en lui envoyant les nouveaux héros à afficher

e) Mise à jour du template HeroesComponent :

On ajoute au bas du templace un élément <app-hero-detail>, puis on va lier les deux composant. Le template à donc changer et toutes la partie en bas nous servant à afficher le nom du héros selectionner n’est pu gérer par le composant heroes.compenent mais par heroes-detail.component

<app-hero-detail [hero]="selectedHero"></app-hero-detail>

2 .

Comme précédemment, chaque fois qu'un utilisateur clique sur un nom de héros, le détail du héros apparaît sous la liste des héros. Maintenant, HeroDetailComponent présente ces détails au lieu de HeroesComponent.

Refactoriser le composant HeroesComponent original en deux composants apporte des avantages, à la fois maintenant et dans le futur:

- On a pu HeroesComponent en réduisant ses responsabilités.

- On a pu faire évoluer HeroDetailComponent dans un éditeur de héros riche sans toucher le parent HeroesComponent.

- On a put faire évoluer le composant HeroesComponent sans toucher à la vue détaillée des héros.

- On a put réutiliser le HeroDetailComponent dans le modèle d'un futur composant.

5. résumé :

- On a créé un composant HeroDetailComponent distinct et réutilisable.

- On a utilisé une liaison de propriété pour donner le contrôle HeroesComponent parent sur l'enfant HeroDetailComponent.

- On a utilisé le décorateur @Input pour rendre la propriété hero disponible pour la liaison par le composant HeroesComponent externe.

b) Sauvegarde du projet

Mettez à jour votre dépot git.

Ce dernier commit aura pour message : "TP5 Composants maître détail Final"