

آموزش نمایش یک لیست از Heroe ها در Angular

در این قسمت ، برنامه Heroe را برای نمایش لیستی از Heroes ها گسترش می دهیم و به کاربر اجازه خواهیم داد تا Heroes را انتخاب کنند و جزئیات را ببینند.

آموزش ایجاد mock heroes در Angular

شما برای نمایش heroها به برخی تگ ها نیاز دارید :

- در نهایت شما آنها را از یک سرور داده از راه دور دریافت خواهید کرد. در حال حاضر، شما می توانید برخی از قهرمانان فکری را ایجاد کنید و وانمود کنید که از سرور دریافت شده اند.
- یک فایل با نام mock-heroes.ts در src / app / folder ایجاد کنید. یک HEROES ثابت را به صورت مجموعه ای از ده heroes تعریف کنید و آن را export کنید. فایل باید مانند این باشد.

```
import { Hero } from './hero';
export const HEROES: Hero[] = [
  id: 11, name: 'Mr. Nice' },
  id: 12, name: 'Narco' },
  id: 13, name: 'Bombasto' },
  id: 14, name: 'Celeritas' },
  id: 15, name: 'Magnetia' },
  id: 16, name: 'RubberMan' },
  id: 17, name: 'Dynamia' },
  id: 18, name: 'Dr IQ' },
  id: 19, name: 'Magma' },
  id: 20, name: 'Tornado' }
];
```

آموزش به نمایش در آوردن heroes در Angular

شما برای نمایش لیستی از heroes ها به فایل HeroesComponent احتیاج دارید. فایل کلاس HeroesComponent را باز کنید و HEROES را وارد کنید.

```
import { HEROES } from '../mock-heroes';
```

یک ویژگی heroes را به کلاس اضافه کنید که این heroes برای اتصال به آن قرار می دهد.

```
heroes = HEROES;
```

آموزش فهرست heroes با * ngFor در Angular

فایل قالب HeroesComponent را باز کنید و تغییرات زیر را انجام دهید:

- تگ `<h2>` در بالای صفحه اضافه کنید.
- در زیر آن یک لیست HTML unordered اضافه می کند (``)
- تگ `` را در داخل تگ `` که ویژگی های hero را نمایش می دهد اضافه کنید.
- برخی از کلاس های CSS را برای یک ظاهر خوب وارد کنید (در کوتاه مدت سبک های CSS را اضافه کنید).
- و در آخر کدهای شما باید مانند زیر باید شده باشد.

- `<h2>My Heroes</h2>`
- `<ul class="heroes">`
- ``
- `{{hero.id}} {{hero.name}}`
- ``
- ``

حالا `` را به صورت زیر تغییر دهید:

```
<li *ngFor="let hero of heroes">
```

*ngFor یک دستور تکرار کننده Angular می باشد. این دستور تگ میزبان را برای هر تگ در یک لیست تکرار می کند.

در این مثال :

`` تگ میزبان است.

Heroes لیستی از کلاس HeroesComponent می باشد.

Hero آجکت فعلی hero را برای هر تکرار از طریق لیست نگه میدارد.

ستاره (*) را در مقابل دستور ngFor فراموش نکنید. این بخش مهمی از دستور است.

پس از رفرش مرورگر، لیست hero ظاهر می شود.

آموزش استایل heroes در Angular

لیست heroes باید جذاب باشد و زمانی که کاربر ماوس خود را بر روی آن حرکت می دهد یک hero از لیست پررنگ شود.

در اولین آموزش، سبک های پایه برای کل برنامه در styles.css را تنظیم می کنید. این شیوه شامل سبک هایی برای این لیست heroes ها نبود.

شما می توانید سبک های بیشتری را به styles.css اضافه کنید و وقتی که components ها را اضافه می کنید، این استایل را رشد کند.

شما ممکن است ترجیح دهید به تعریف سبک های خصوصی برای یک component خاص و هر یک از نیاز های component - کد، HTML و CSS را در یک مکان نگهداری کنید.

این روش باعث می شود که مجدداً از component در جایی دیگر استفاده شود و حتی اگر سبک (Style) های جهانی متفاوت باشند، ظاهر مورد نظر این component را ارائه می دهد.

شما می توانید سبک های (styles) شخصی خود را به صورت خطی در آرایه @Component.styles تعریف کنید و یا به عنوان فایل یا فایل های stylesheet در آرایه @Component.styleUrls تعریف کنید.

زمانی که CLI یک HeroesComponent را تولید کرد ، یک فایل heroes.component.css خالی را برای HeroesComponent ایجاد می کند و @Component.styleUrls به این مسئله اشاره می کند.

```
@Component({
  selector: 'app-heroes',
  templateUrl: './heroes.component.html',
  styleUrls: ['./heroes.component.css']
})
```

فایل heroes.component.css را باز کنید و سبک های CSS خصوصی برای HeroesComponent وارد کنید. در نهایت در انتهای همین مقاله شما می توانید کدهای وارد شده را ببینید.

استایل ها و شیوه های تعیین شده در متادیتای @Component به آن component خاص تخصیص داده می شود.

استایل و سبک heroes.component.css فقط بر روی HeroesComponent اعمال می شود و بر روی HTML بیرونی یا HTML در هر component دیگر تاثیر نمی گذارد.

آموزش Angular در Master/Detail

هنگامی که کاربر یک hero را در لیست اصلی باز کند، component باید جزئیات hero انتخاب شده را در پایین صفحه را نمایش دهد.

در این بخش شما event های آیتم hero را چک و به روز می کنید.

افزودن event کلیک برای اتصال

یک event کلیک را به مانند زیر اضافه کنید:

```
<li *ngFor="let hero of heroes" (click)="onSelect(hero)">
```

این یک مثال از دستور زنجیره اتصال event در Angular است.

پراپرتی در اطراف click به Angular می گوید که برای چک کردن تگ کلیک کنید. هنگامی که کاربر در کلیک میکند، انگولار onSelect expression (hero) را اجرا میکند.

onSelect () یک متد HeroesComponent می باشد که شما در حال نوشتن آن هستید. Angular آن را با آجکت hero نمایش داده شده در کلیک ، همان hero که قبلا در عبارت *ngFor تعریف شده است فرخوانی می کند.

آموزش اضافه کردن دستیار event کلیک در Angular

ویژگی های کامپوننت hero را به selectedHero تغییر نام دهید اما تایید نکنید ، هیچ hero ای در زمانی که برنامه اجرا می شود انتخاب نشده است.

متد onSelect () زیر اضافه کنید ، که کلیک hero از قالب را به کامپوننت selectedHero اختصاص میدهد.

```
selectedHero: Hero;
onSelect(hero: Hero): void {
  this.selectedHero = hero;
}
```

جزئیات قالب را به روز کنید

قالب هنوز به ویژگی های کامپوننت hero اشاره دارد که دیگه همچین چیزی وجود ندارد. Hero را به selectedHero تغییر نام دهید.

```
<h2>{{ selectedHero.name | uppercase }} Details</h2>
<div><span>id: </span>{{selectedHero.id}}</div>
<div>
  <label>name:
  <input [(ngModel)]="selectedHero.name" placeholder="name">
</label>
</div>
```

آموزش مخفی کردن empty details با *ngIf در Angular

پس از رفرش مرورگر، برنامه با خطا مواجه می شود.

developer tools مرورگر را باز کنید و در کنسول یک پیام خطا مانند زیر است که مشاهده می کنید :

```
HeroesComponent.html:3 ERROR TypeError: Cannot read property 'name' of undefined
```

در حال حاضر بر روی یکی از موارد لیست کلیک کنید. به نظر می رسد برنامه دوباره کار میکند. heroes هادر یک لیست ظاهر می شوند و جزئیات مربوط به hero کلیک شده در پایین صفحه ظاهر می شود.

چه اتفاقی افتاد؟

هنگامی که برنامه شروع به کار می کند، "selectedHero" با طراحی تعریف نشده است. عبارات مرتبط در قالب که به properties های selectedHero (برای مثال مانند {{selectedHero.name}}) اشاره دارند، باید با خط مواجه شده باشند زیرا هیچ hero ای انتخاب نشده است.

رفع خطا

در کامپوننت فقط باید جزئیات hero انتخاب شده را نمایش بدهد در صورتی که باید selectedHero وجود داشته باشد.

جزئیات hero در یک تگ <div> وارد کنید.

دستورالعمل * ngIf Angular را به <div> اضافه کنید و آن را به selectedHero تنظیم کنید.

سناره (*) را در مقابل ngIf فراموش نکنید. این بخش مهمی از دستور می باشد.

```
<div *ngIf="selectedHero">
  <h2>{{ selectedHero.name | uppercase }} Details</h2>
  <div><span>id: </span>{{selectedHero.id}}</div>
</div>
<label>name:
<input [(ngModel)]="selectedHero.name" placeholder="name">
</label>
</div>
</div>
```

پس از رفرش مرورگر، لیست نامها دوباره ظاهر می شود. قسمت جزئیات خالی است یک hero را انتخاب کنید تا جزئیات آن ظاهر شود.

چگونه مشکل بر طرف شد؟

هنگامی که selectedHero نامشخص است، ngIf جزئیات hero را از DOM حذف می کند.

هنگامی که کاربر یک hero را انتخاب می کند، selectedHero یک value دارد و ngIf جزئیات hero را در DOM قرار می دهد.

استایل hero انتخاب شده

وقتی تگ های `` به صورت یکسان ظاهر می شوند، شناسایی hero انتخاب شده در لیست دشوار است.

در واقع برای مثال کاربر با انتخاب و کلیک بر روی Magneta رنگی متفاوت و ظریفی باید مشاهده کند تا به سادگی انتخاب خودش را بتواند تشخیص بدهد.



برای اینکه شما هم مانند تصویر بالا همچین ظاهری داشته باشید شما فقط باید کلاس انتخاب شده را به `` اختصاص دهید تا زمانی که کاربر روی آن کلیک کند ظاهری مانند تصویر بالا داشته باشد.

حذف و اضافه و اتصال کلاس ها در انگولار بسیار ساده می باشد. شما فقط باید دستور `[class.some-css]` به تگ مورد نظرتون که می خواهید استایل خاصی داشته باشد اضافه کنید.

مانند اتصال کلاس اضافه شده ، کد `[class.selected]` زیر را به `` در قالب `HeroesComponent` اضافه کنید:

```
[class.selected]="hero === selectedHero"
```

هنگامی که hero ردیف فعلی همان `selectedHero` است، Angular کلاس CSS انتخاب شده را اضافه می کند. هنگامی که دو hero متفاوت هستند، Angular کلاس را حذف می کند.

آخرین `` مانند زیر باید باشد :

```
<li *ngFor="let hero of heroes"
  [class.selected]="hero === selectedHero"
  (click)="onSelect(hero)">
  <span class="badge">{{hero.id}}</span> {{hero.name}}
</li>
```

بررسی کد نهایی

در این قسمت تمامی کد فایل هایی که در این بخش از آموزش بررسی کردیم به صورت یکجا وجود دارد.

خلاصه

برنامه Tour of Heroes لیستی از hero ها را در یک نمایه Master / Detail نمایش می دهد.

کاربر می تواند یک hero را انتخاب کند و جزئیات این hero را ببیند.

شما برای نمایش یک لیست از * ngFor استفاده می کنید.

شما با استفاده از * ngIf به صورت شرطی می توانید یک بلوک از HTML را کم و یا اضافه کنید .

شما می توانید کلاس استایل CSS را با استفاده از class binding تغییر دهید