组件间通信

编辑星级评价

Stars.html

<!--41.星星的编辑状态-->  
<span \*ngFor="let star of stars; let i= index;" class="glyphicon glyphicon-star" [class.glyphicon-star-empty]="star" (click)="clickStar(i)"></span>

Stars.ts

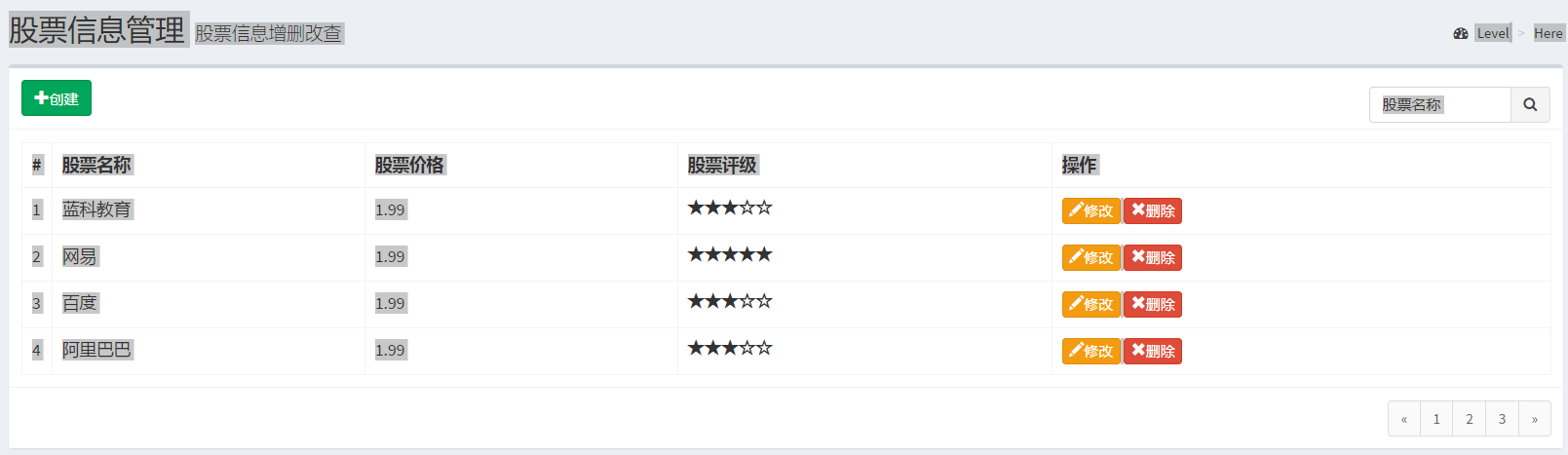
// 42.实现方法  
clickStar(index: **number**){  
 **this**.rating = index;  
 **this**.stars = [];  
 **for** (**let** i = 1; i <= 5; i++){  
 // this.stars = [false, false, true, true, true];  
 **this**.stars.push(i>**this**.rating);  
 }  
}



此时已经能够点击, 但是第五个星星无法点击

改成 **this**.rating = index +1;

此时引入了新的问题, 列表页的星星也是可以点击的



在stars.ts中

// 43.增加控制星星只读的属性  
@Input()  
readonly : **boolean** = **true**;

// 44.控制星星只读  
clickStar(index: **number**){  
 **if**(!**this**.readonly){  
 **this**.rating = index +1;  
 **this**.stars = [];  
 **for** (**let** i = 1; i <= 5; i++){  
 // this.stars = [false, false, true, true, true];  
 **this**.stars.push(i>**this**.rating);  
 }

Stockform.html’’

<!--&lt;!&ndash;45.表单里的星星是可以点击的&ndash;&gt;-->  
<!--<app-stars [rating]="stock?.rating" [readonly]="false"></app-stars>-->

实现星星数据的双向绑定

Stars.ts

// 46.将数据传出外部  
@Output()  
 ratingChange: EventEmitter<**number**> = **new** EventEmitter();

// 44.控制星星只读  
clickStar(index: **number**){  
 **if**(!**this**.readonly){  
 **this**.rating = index +1;  
 **this**.stars = [];  
 **for** (**let** i = 1; i <= 5; i++){  
 // this.stars = [false, false, true, true, true];  
 **this**.stars.push(i>**this**.rating);  
 }  
 // 47.点击时发射此数据  
 **this**.ratingChange.emit(**this**.rating);  
 }  
}

Stockform.ts中测试

save(){  
 **this**.router.navigateByUrl('/stock');  
 // 48.测试双向绑定成功  
 console.log(**this**.stock.rating);  
}

优化stars.ts代码

// 49.优化  
**export class** StarsComponent **implements** OnInit , OnChanges {  
 // 2外部接收数据  
 @Input()  
 rating: **number** = 0;  
  
 // 46.将数据传出外部  
 @Output()  
 ratingChange: EventEmitter<**number**> = **new** EventEmitter();  
  
 // 1声明变量  
 stars: **boolean**[];  
  
 // 43.增加控制星星只读的属性  
 @Input()  
 readonly : **boolean** = **true**;  
  
 **constructor**() { }  
  
 ngOnInit() {  
   
 }  
 // 输入属性发生变化时调用  
 ngOnChanges(changes: SimpleChanges): **void**{  
 **this**.stars = [];  
 **for** (**let** i = 1; i <= 5; i++){  
 // this.stars = [false, false, true, true, true];  
 **this**.stars.push(i>**this**.rating);  
 }  
 }  
 // // 42.实现方法  
 // clickStar(index: number){  
 // this.rating = index +1;  
 // this.stars = [];  
 // for (let i = 1; i <= 5; i++){  
 // // this.stars = [false, false, true, true, true];  
 // this.stars.push(i>this.rating);  
 // }  
 // }  
  
 // 44.控制星星只读  
 clickStar(index: **number**){  
 **if**(!**this**.readonly){  
 **this**.rating = index +1;  
  
 // 47.点击时发射此数据  
 **this**.ratingChange.emit(**this**.rating);  
 }  
 }  
}