https://blog.csdn.net/weixin\_43938259/article/details/104714806

对于ng8：

cnpm i –save @angulat/cdk@8.1.2 @angular/material@8.1.2 @angular/animations@8.2.8 hammerjs

cnpm I –save @angular/flex-layout@8.0.0-beta.26

1、安装

* 1. **安装@angular/material、@angular/cdk**

cnpm install --save @angular/material @angular/cdk

* 1. **app.modules.ts导入**

1. import { DragDropModule } from '@angular/cdk/drag-drop';
2. imports: [
3. ...
4. DragDropModule
5. ]

**2.功能实现**

## 2.1 拖拽

html编辑如下代码即可：

<div cdkDrag class="drag-box"

drag me

</div>

## 2.2 排序

html:

<h3>列表排序</h3>

<div class="box-list" cdkDropList (cdkDropListDropped)="drop($event)">

<div class="drag-box" \*ngFor="let customer of customers" cdkDrag>

{{customer.name}}

</div>

</div>

ts:

import { ...CdkDragDrop, moveItemInArray } from '@angular/cdk/drag-drop';

customers = [

{ name: 'Adam', age: 23 },

{ name: 'Jack', age: 27 },

{ name: 'Katherin', age: 26 },

{ name: 'John', age: 30 },

{ name: 'Watson', age: 42 },

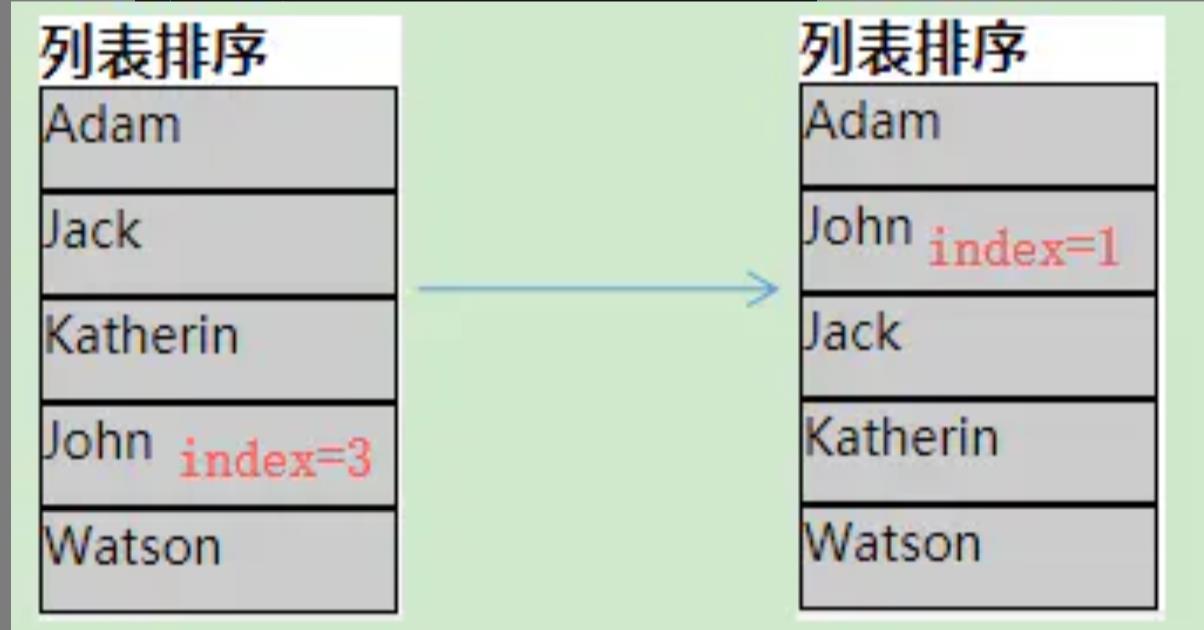
];

drop(event: CdkDragDrop<string[]>) {

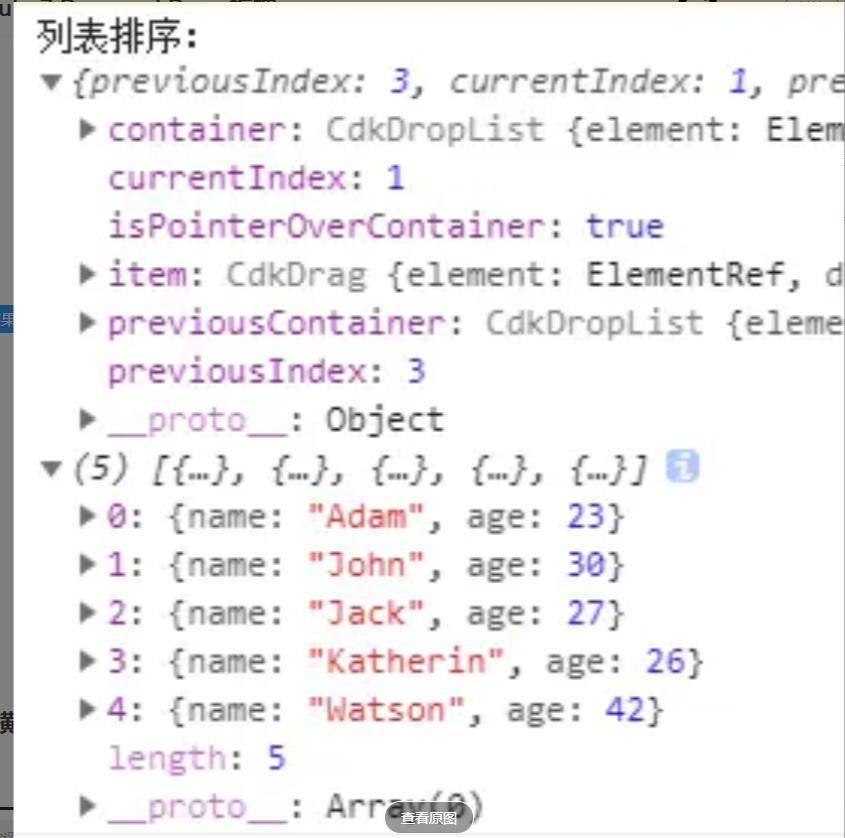
console.log('列表排序：', event, this.customers);

moveItemInArray(this.customers, event.previousIndex, event.currentIndex);

执行如下操作：



打印结果：



## 2.3 横向排序

html:

<h3>横向排序</h3>

<div class="box-list-horizontal" cdkDropList cdkDropListOrientation="horizontal"

(cdkDropListDropped)="drop($event)">

<div class="drag-box" \*ngFor="let customer of customers" cdkDrag>

{{customer.name}}

</div>

</div>

## 2.4 添加动画效果

为了设置动画，我们定义了一个以transform属性为目标的转换。拖放CDK支持动画：

--对列表中的元素进行拖动排序时的动画=> .cdk-drag

.cdk-drag {

transition: transform 100ms ease;

}

--从动画的位置到最终把它放在列表的位置上时的动画=> .cdk-drag-animating

.cdk-drag-animating {

transition: transform 300ms ease;

}

## 2.5 占位符元素 placeholder

### ****默认占位符****

使用.cdk-drag-placeholder来显示占位符元素，而不是实际的元素，因为它是在cdkDropList中拖动的。默认情况下，这看起来与正在排序的元素完全相同。

.cdk-drag-placeholder {

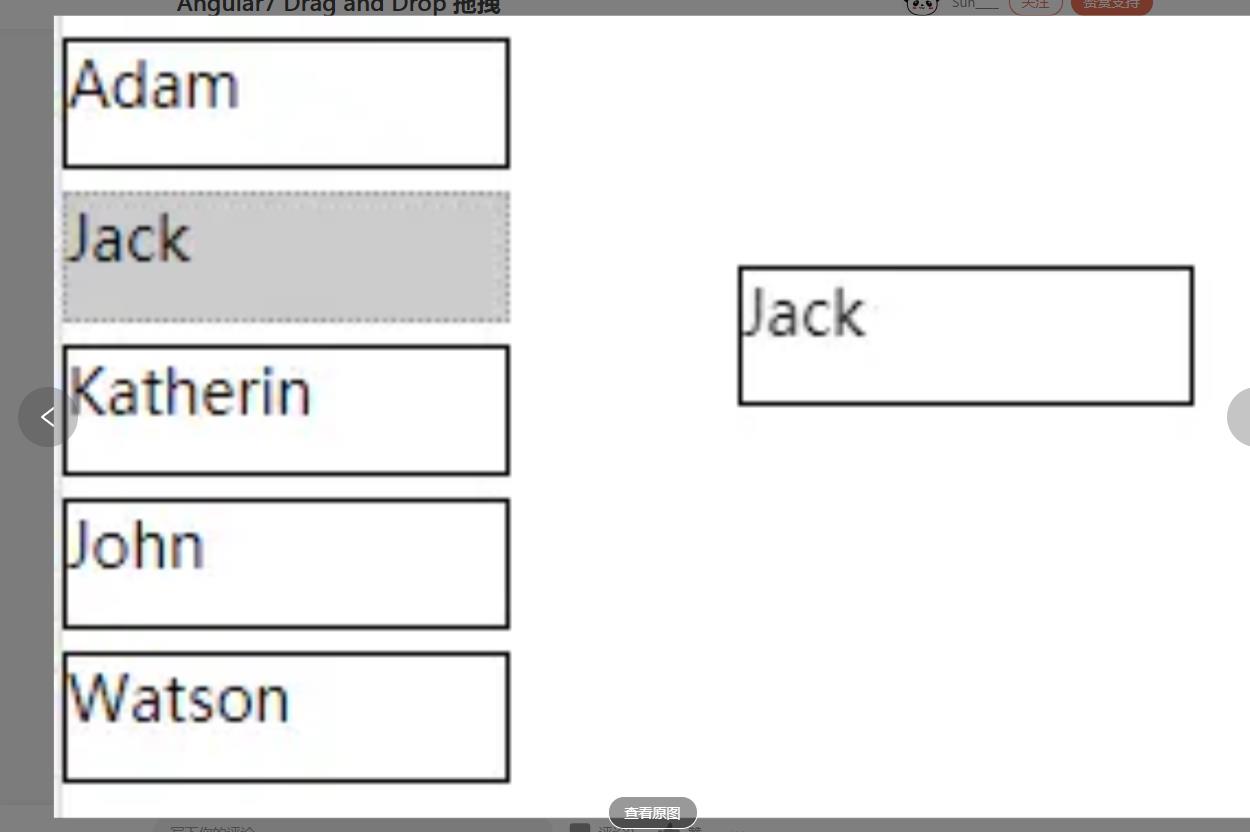
background: #ccc;

border: dotted 1px #999;

transition: transform 500ms ease;

}

拖拽时效果如下图：



### ****自定义占位符****

使用\*cdkDragPlaceholder指令，我们可以用一个自定义的占位符来代替默认的占位符

html:

<div class="box-list customph" cdkDropList (cdkDropListDropped)="drop($event)">

<div class="drag-box dragList" \*ngFor="let customer of customers" cdkDrag >

<div class="box-custom-placeholder" \*cdkDragPlaceholder></div>

{{customer.name}}

</div>

</div>

css:

// 自定义占位符

.customph{

.box-custom-placeholder {

height: 28px;

width: 100px;

background: #fff;

border: dotted 1px #0084ff;

transition: transform 200ms ease;

}

&.box-list.cdk-drop-list-dragging .drag-box:not(.cdk-drag-placeholder) {

transition: transform 500ms ease;

}

}

## 2.6 预览

当提取并拖动cdkDrag元素时，可以看到预览元素。默认情况下，预览元素看起来与被拖动的元素完全相同。

使用.cdk-drag-preview来定义预览CSS:

// 预览

.cdk-drag-preview {

height: 28px;

width: 100px;

box-shadow: 0 3px 3px -3px #0084ff;

}

我们还需要使用\*cdkDragPreview提供一个自定义模板:

<div cdkDropList class="box-list" (cdkDropListDropped)="drop($event)">

<div class="drag-box dragList" \*ngFor="let customer of customers" cdkDrag>

{{customer.name}}

<p \*cdkDragPreview>Age: {{customer.age}}</p>

</div>

</div>

## 2.7 多个列表之间拖拽

我们可以使用cdkDropListConnectedTo属性将一个或多个cdkDropList连接到一起。然后设置cdkDropListData和cdkDragData，将数据与cdkDropList和cdkDrag关联起来。

html:

<div class="box-list" cdkDropList #inactiveList="cdkDropList" id="Inactive Customers"

[cdkDropListData]="inactiveCustomers" [cdkDropListConnectedTo]="[activeList]"

(cdkDropListDropped)="drop1($event)">

<div class="drag-box dragList"

[cdkDragData]="customer"

\*ngFor="let customer of inactiveCustomers"

cdkDrag>

{{customer.name}}

</div>

</div>

<div class="box-list" cdkDropList #activeList="cdkDropList" id="Active Customers"

[cdkDropListData]="activeCustomers" [cdkDropListConnectedTo]="[inactiveList]"

(cdkDropListDropped)="drop1($event)">

<div class="drag-box dragList"

[cdkDragData]="customer"

\*ngFor="let customer of activeCustomers"

cdkDrag>

{{customer.name}}

</div>

</div>

ts:

import { ...transferArrayItem } from '@angular/cdk/drag-drop';

inactiveCustomers = [

{name: 'Jack', age: 18},

{name: 'Katherin', age: 16},

{name: 'Adam', age: 36}

];

activeCustomers = [

{name: 'John', age: 10},

{name: 'Watson', age: 24}

];

drop1(event: CdkDragDrop<string[]>) {

console.log('拖拽事件：event', event);

if (event.previousContainer === event.container) {

console.log('拖拽事件',

`> 拖 '${event.item.data}' 到 '${event.container.id}'`);

moveItemInArray(event.container.data, event.previousIndex, event.currentIndex);

} else {

console.log('拖拽事件',

`> 拖 '${event.item.data}' 到 '${event.container.id}'`);

transferArrayItem(

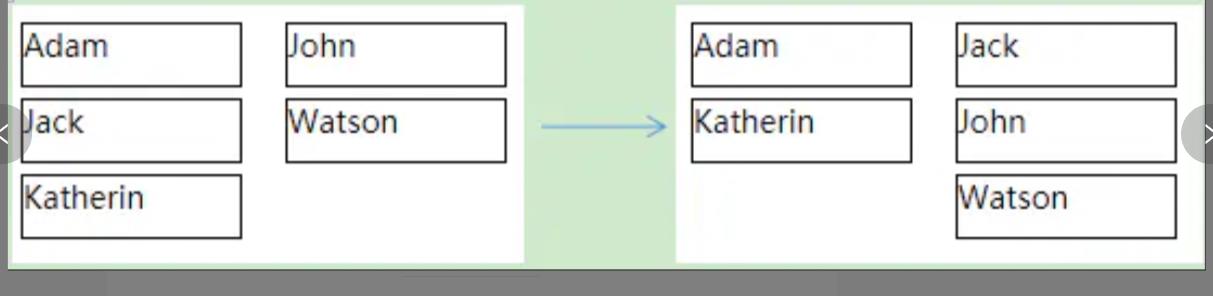
event.previousContainer.data, event.container.data,

event.previousIndex, event.currentIndex);

}

}

执行如下操作：







**CdkDropListGroup**

CdkDropListGroup指令的范围比CdkDropList要大一级。CdkDropListGroup可以包含多个CdkDropList。而且当CdkDropListGroup包含多个CdkDropList的时候，这些CdkDropList直接是相互connect的(CdkDropList就不用去写cdkDropListConnectedTo属性了)。

<div cdkDropListGroup>

<div class="box-list" cdkDropList id="Inactive Customers"

[cdkDropListData]="inactiveCustomers"

(cdkDropListDropped)="drop1($event)">

<div class="drag-box dragList" [cdkDragData]="customer"

\*ngFor="let customer of inactiveCustomers" cdkDrag>

{{customer.name}}

</div>

</div>

<div class="box-list" cdkDropList id="Active Customers"

[cdkDropListData]="activeCustomers"

(cdkDropListDropped)="drop1($event)">

<div class="drag-box dragList"

[cdkDragData]="customer"

\*ngFor="let customer of activeCustomers"

cdkDrag>

{{customer.name}}

</div>

</div>

</div>

## 2.8 事件监听

**Drag Event Handlers**

– **cdkDragStarted:** 当用户开始拖动时发出。

– **cdkDragEnded:** 当用户停止拖动时发出。

– **cdkDragEntered:** 将item移动到新容器中时发出。

– **cdkDragExited:**将item移出当前容器时发出。

**Drop Event Handlers**

– **cdkDropListDropped**: emits when an item was dropped inside the container.

– **cdkDropListEntered**: emits when a new item is dragged into this container.

– **cdkDropListExited**: emits when an item is dragged from this container into another container.

### cdkDragStarted、cdkDragEnded、cdkDragMoved

html:

<h3>事件监听</h3>

<div cdkDrag class="drag-box"

(cdkDragStarted)="dragStarted($event)"

(cdkDragEnded)="dragEnded($event)"

(cdkDragMoved)="dragMoved($event)"

>

drag me

</div>

<p>{{state}} {{position}}</p>

ts:

import { CdkDragEnd, CdkDragStart, CdkDragMove... } from '@angular/cdk/drag-drop';

dragStarted(event: CdkDragStart) {

this.state = '拖拽开始';

console.log('拖拽开始');

}

dragEnded(event: CdkDragEnd) {

this.state = '拖拽结束';

console.log('拖拽结束');

}

dragMoved(event: CdkDragMove) {

console.log('dragMoved');

this.position = `> Position X: ${event.pointerPosition.x} - Y: ${event.pointerPosition.y}`;

}

### cdkDropListEntered、cdkDropListExited、cdkDragEntered、cdkDragExited

html:

<div class="box-list" cdkDropList #inactiveList="cdkDropList" id="Inactive Customers"

[cdkDropListData]="inactiveCustomers" [cdkDropListConnectedTo]="[activeList]"

(cdkDropListDropped)="drop1($event)"

(cdkDropListEntered)="dropListEntered($event)" (cdkDropListExited)="dropListExited($event)">

<div class="drag-box dragList" [cdkDragData]="customer"

(cdkDragEntered)="dragEntered($event)" (cdkDragExited)="dragExited($event)"

\*ngFor="let customer of inactiveCustomers"

cdkDrag>

{{customer}}

</div>

</div>

<div class="box-list" cdkDropList #activeList="cdkDropList" id="Active Customers"

[cdkDropListData]="activeCustomers" [cdkDropListConnectedTo]="[inactiveList]"

(cdkDropListDropped)="drop1($event)">

<div class="drag-box dragList"

[cdkDragData]="customer"

\*ngFor="let customer of activeCustomers"

cdkDrag>

{{customer}}

</div>

</div>

ts:

import { ...CdkDragEnter, CdkDragExit} from '@angular/cdk/drag-drop';

dragEntered(event: CdkDragEnter) {

console.log(`drag:把 '${event.item.data}' 拖进 '${event.container.id}' `);

}

dragExited(event: CdkDragExit) {

console.log(`drag:把 '${event.item.data}' 从 '${event.container.id}' 中拖出 `);

}

dropListEntered(event: CdkDragEnter) {

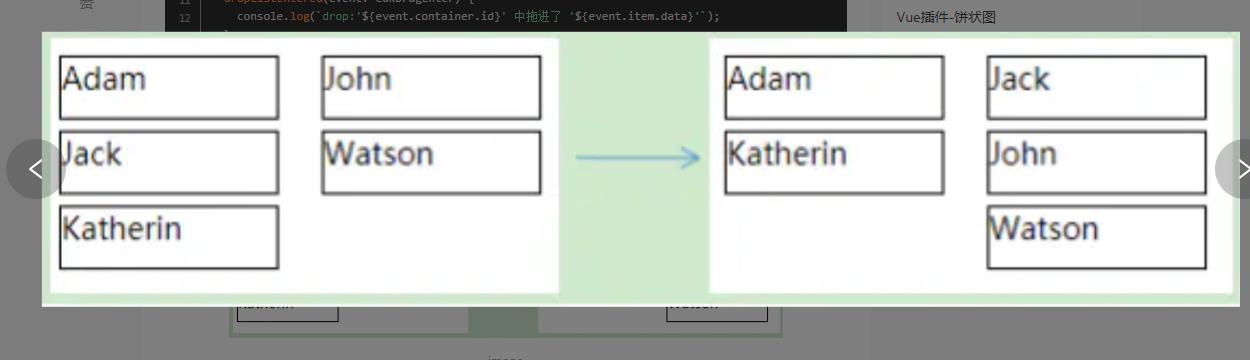
console.log(`drop:'${event.container.id}' 中拖进了 '${event.item.data}'`);

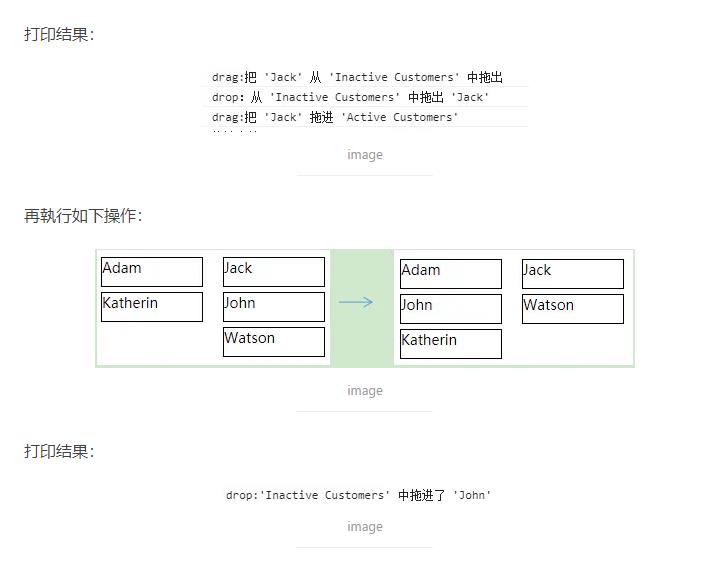
}

dropListExited(event: CdkDragExit) {

console.log(`drop：从 '${event.container.id}' 中拖出 '${event.item.data}'`);

}

执行如下操作



### ****cdkDropListSorted****

拖拽过程中交换item的时候回调

<div class="box-list" cdkDropList #activeList="cdkDropList" id="Active Customers"

[cdkDropListData]="activeCustomers" [cdkDropListConnectedTo]="[inactiveList]"

(cdkDropListDropped)="drop1($event)"

(cdkDropListSorted)="dropListSorted($event)">

<div class="drag-box dragList"

[cdkDragData]="customer"

\*ngFor="let customer of activeCustomers"

cdkDrag>

{{customer.name}}

</div>

</div>

ts:

dropListSorted(event: CdkDropList) {

console.log('Sorted:', event);

}

## 2.9 属性

### ****cdkDropListEnterPredicate****

enterPredicate: (drag: CdkDrag, drop: CdkDropList) => boolean

指定哪些item是可以拖拽到当前容器

html:

<div class="box-list" cdkDropList #activeList="cdkDropList" id="Active Customers"

[cdkDropListData]="activeCustomers" [cdkDropListConnectedTo]="[inactiveList]"

(cdkDropListDropped)="drop1($event)" [cdkDropListEnterPredicate]="predicate">

<div class="drag-box dragList"

[cdkDragData]="customer"

\*ngFor="let customer of activeCustomers"

cdkDrag>

{{customer.name}}

</div>

</div>

ts:

import { ...CdkDrag, CdkDropList } from '@angular/cdk/drag-drop';

predicate(drag: CdkDrag, drop: CdkDropList) {

console.log('predicate:', drag, drop);

return drag.data.age >= 18;

}

**CdkDrag属性：**



**CdkDropList属性**



代码：

html:

<div class="drag-box" cdkDrag>

drag me

</div>

<h3>锁定方向</h3>

<div class="drag-box" cdkDragLockAxis="y" cdkDrag>

only up/down

</div>

<div class="drag-box" cdkDragLockAxis="x" cdkDrag>

only left/right

</div>

<h3>事件监听</h3>

<div cdkDrag class="drag-box"

(cdkDragStarted)="dragStarted($event)"

(cdkDragEnded)="dragEnded($event)"

(cdkDragMoved)="dragMoved($event)"

>

drag me

</div>

<p>{{state}} {{position}}</p>

<h3>拖拽handler</h3>

<div class="drag-box-with-handler drag-box" cdkDrag>

<div class="box-handler" cdkDragHandle>

点我拖拽

</div>

</div>

<h3>列表排序</h3>

<div class="box-list" cdkDropList (cdkDropListDropped)="drop($event)">

<div class="drag-box dragList" \*ngFor="let customer of customers" cdkDrag>

{{customer.name}}

</div>

</div>

<h3>横向排序</h3>

<div class="box-list-horizontal" cdkDropList cdkDropListOrientation="horizontal"

(cdkDropListDropped)="drop($event)">

<div class="drag-box dragListH" \*ngFor="let customer of customers" cdkDrag>

{{customer.name}}

</div>

</div>

<h3>placeholder 默认占位符元素</h3>

<div>使用.cdk-drag-placeholder来显示占位符元素，而不是实际的元素，

因为它是在cdkDropList中拖动的。默认情况下，占位符元素看起来与正在排序的元素完全相同。</div>

<div class="box-list defplaceh" cdkDropList (cdkDropListDropped)="drop($event)">

<div class="drag-box dragList" \*ngFor="let customer of customers" cdkDrag>

{{customer.name}}

</div>

</div>

<h3>placeholder 自定义占位符元素</h3>

<div>使用\*cdkDragPlaceholder指令，我们可以用一个自定义的占位符来代替默认的占位符</div>

<div class="box-list customph" cdkDropList (cdkDropListDropped)="drop($event)">

<div class="drag-box dragList" \*ngFor="let customer of customers" cdkDrag >

<div class="box-custom-placeholder" \*cdkDragPlaceholder></div>

{{customer.name}}

</div>

</div>

<h3>预览 Preview</h3>

<div>当提取并拖动cdkDrag元素时，可以看到预览元素。默认情况下，

预览元素看起来与被拖动的元素完全相同。</div>

<div cdkDropList class="box-list" (cdkDropListDropped)="drop($event)">

<div class="drag-box dragList" \*ngFor="let customer of customers" cdkDrag>

{{customer.name}}

<p \*cdkDragPreview>Age: {{customer.age}}</p>

</div>

</div>

<h3>两个列表间拖放</h3>

<div class="twoList">

<div class="box-list" cdkDropList #inactiveList="cdkDropList" id = "Inactive Customers"

[cdkDropListData]="inactiveCustomers" [cdkDropListConnectedTo]="[activeList]"

(cdkDropListDropped)="drop1($event)"

(cdkDropListEntered)="dropListEntered($event)" (cdkDropListExited)="dropListExited($event)"

[cdkDropListEnterPredicate]="predicate"

>

<div class="drag-box dragList" [cdkDragData]="customer"

(cdkDragEntered)="dragEntered($event)" (cdkDragExited)="dragExited($event)"

\*ngFor="let customer of inactiveCustomers;isLast as last"

cdkDrag>

{{customer.name}}

</div>

</div>

<div class="box-list" cdkDropList #activeList="cdkDropList" id = "Active Customers"

[cdkDropListData]="activeCustomers" [cdkDropListConnectedTo]="[inactiveList]"

(cdkDropListDropped)="drop1($event)" [cdkDropListEnterPredicate]="predicate"

(cdkDropListSorted)="dropListSorted($event)"

>

<div

class="drag-box dragList"

[cdkDragData]="customer"

\*ngFor="let customer of activeCustomers"

cdkDrag>

{{customer.name}}

</div>

</div>

</div>

<!-- <h3>两个列表间拖放 cdkDropListGroup</h3>

<div class="twoList" cdkDropListGroup>

<div class="box-list" cdkDropList id = "Inactive Customers"

[cdkDropListData]="inactiveCustomers"

(cdkDropListDropped)="drop1($event)">

<div class="drag-box dragList" [cdkDragData]="customer"

\*ngFor="let customer of inactiveCustomers" cdkDrag>

{{customer.name}}

</div>

</div>

<div class="box-list" cdkDropList id = "Active Customers"

[cdkDropListData]="activeCustomers"

(cdkDropListDropped)="drop1($event)">

<div

class="drag-box dragList"

[cdkDragData]="customer"

\*ngFor="let customer of activeCustomers"

cdkDrag>

{{customer.name}}

</div>

</div>

</div> -->

ts:

import { Component, OnInit } from '@angular/core';

import { CdkDragEnd, CdkDragStart, CdkDragMove,

CdkDragDrop, CdkDragEnter, CdkDragExit, moveItemInArray,

transferArrayItem, CdkDrag, CdkDropList } from '@angular/cdk/drag-drop';

@Component({

selector: 'app-drag-drop',

templateUrl: './drag-drop.component.html',

styleUrls: ['./drag-drop.component.scss']

})

export class DragDropComponent implements OnInit {

state = '';

position = '';

customers = [

{ name: 'Adam', age: 23 },

{ name: 'Jack', age: 27 },

{ name: 'Katherin', age: 26 },

{ name: 'John', age: 30 },

{ name: 'Watson', age: 42 },

];

inactiveCustomers = [

{name: 'Jack', age: 18},

{name: 'Katherin', age: 16},

{name: 'Adam', age: 36}

];

activeCustomers = [

{name: 'John', age: 10},

{name: 'Watson', age: 24}

];

constructor() { }

ngOnInit() {

}

dragStarted(event: CdkDragStart) {

this.state = '拖拽开始';

console.log('拖拽开始');

}

dragEnded(event: CdkDragEnd) {

this.state = '拖拽结束';

console.log('拖拽结束');

}

dragMoved(event: CdkDragMove) {

console.log('dragMoved');

this.position = `> Position X: ${event.pointerPosition.x} - Y: ${event.pointerPosition.y}`;

}

dragEntered(event: CdkDragEnter) {

console.log(`drag:把 '${event.item.data}' 拖进 '${event.container.id}' `);

}

dragExited(event: CdkDragExit) {

console.log(`drag:把 '${event.item.data}' 从 '${event.container.id}' 中拖出 `);

}

dropListEntered(event: CdkDragEnter) {

console.log(`drop:'${event.container.id}' 中拖进了 '${event.item.data}'`);

}

dropListExited(event: CdkDragExit) {

console.log(`drop：从 '${event.container.id}' 中拖出 '${event.item.data}'`);

}

drop(event: CdkDragDrop<string[]>) {

console.log('列表排序：', event, this.customers);

moveItemInArray(this.customers, event.previousIndex, event.currentIndex);

}

drop1(event: CdkDragDrop<string[]>) {

console.log('拖拽事件：event', event);

if (event.previousContainer === event.container) {

console.log('拖拽事件',

`> 拖 '${event.item.data}' 到 '${event.container.id}'`);

moveItemInArray(event.container.data, event.previousIndex, event.currentIndex);

} else {

console.log('拖拽事件',

`> 拖 '${event.item.data}' 到 '${event.container.id}'`);

transferArrayItem(

event.previousContainer.data, event.container.data,

event.previousIndex, event.currentIndex);

}

}

predicate(drag: CdkDrag, drop: CdkDropList) {

console.log('predicate:', drag, drop);

return drag.data.age >= 18;

}

dropListSorted(event: CdkDropList) {

console.log('Sorted:', event);

}

}

app.module.ts

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { ScrollingModule } from '@angular/cdk/scrolling';

import { MatCardModule } from '@angular/material';

import { DragDropModule } from '@angular/cdk/drag-drop';

// import { VIEWPORT\_RULER\_PROVIDER } from '@angular/cdk/scrolling';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { VirtualScrollComponent } from './virtual-scroll/virtual-scroll.component';

import { DragDropComponent } from './drag-drop/drag-drop.component';

@NgModule({

declarations: [

AppComponent,

VirtualScrollComponent,

DragDropComponent

],

imports: [

BrowserModule,

AppRoutingModule,

ScrollingModule,

MatCardModule,

DragDropModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

css:

.drag-box{

height: 100px;

width: 100px;

border: 1px solid #000;

}

.box-handler{

width: 100%;

height: 20px;

background: #ccc;

}

.dragList{

height: 28px;

margin: 5px 0;

}

.dragListH{

display: inline-block;

margin: 0 5px;

height: 28px;

width: 55px;

}

// 动画效果

// .cdk-drag {

// transition: transform 100ms ease;

// }

.cdk-drag-animating {

transition: transform 300ms ease;

}

// 占位符 默认 placeholder

.defplaceh .cdk-drag-placeholder {

background: #ccc;

border: dotted 1px #999;

transition: transform 500ms ease;

}

// 自定义占位符

.customph{

.box-custom-placeholder {

height: 28px;

width: 100px;

background: #fff;

border: dotted 1px #0084ff;

transition: transform 200ms ease;

}

&.box-list.cdk-drop-list-dragging .drag-box:not(.cdk-drag-placeholder) {

transition: transform 500ms ease;

}

}

// 预览

.cdk-drag-preview {

height: 28px;

width: 100px;

box-shadow: 0 3px 3px -3px #0084ff;

}

.twoList{

overflow: hidden;

>div{

float: left;

margin: 10px;

}

}