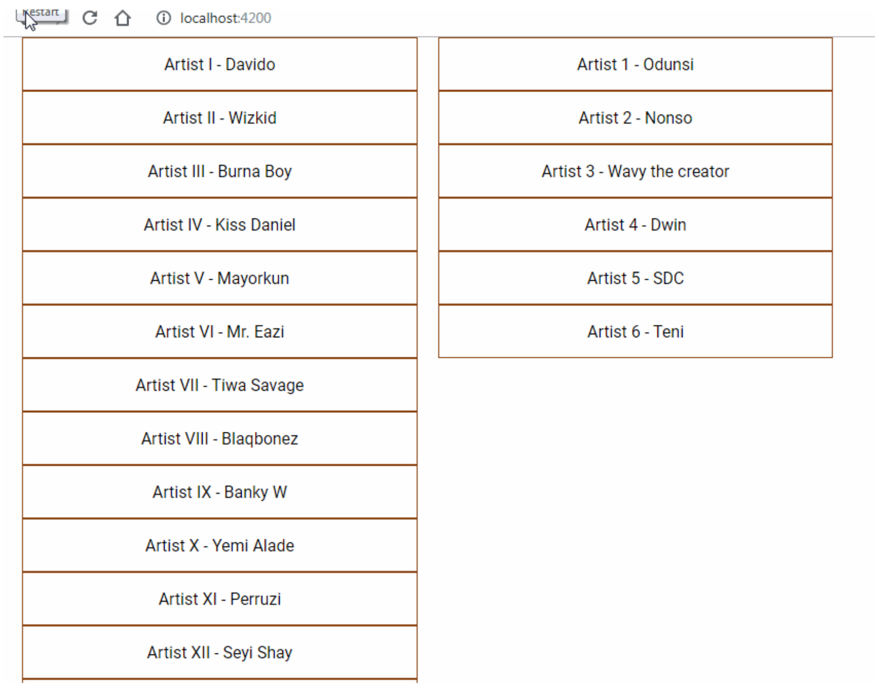
Building Interactive Lists with the new Angular 7 Drag and Drop tool

Nwose Lotanna Follow
Nov 22, 2018 · 6 min read

In this article, we will learn about and take a good look at how to use the Drag and Drop tool in the Angular Material Development Kit.



Artist I - Davido	Artist 1 - Odunsi
Artist II - Wizkid	Artist 2 - Nonso
Artist III - Burna Boy	Artist 3 - Wavy the creator
Artist IV - Kiss Daniel	Artist 4 - Dwin
Artist V - Mayorkun	Artist 5 - SDC
Artist VI - Mr. Eazi	Artist 6 - Teni
Artist VII - Tiwa Savage	
Artist VIII - Blaqbonez	
Artist IX - Banky W	
Artist X - Yemi Alade	
Artist XI - Perruzi	
Artist XII - Seyi Shay	

Angular is JavaScript (TypeScript) framework for building web applications, mobile or desktop with over 42,000 ⭐ on GitHub. It is maintained by the Angular Team at Google and a host of community members and organizations. Angular Version 7 was released few weeks ago. One of the features it shipped with is the **Drag and Drop** tool in the **Component Development Kit**.

Component Development Kit (CDK)

The **Component Development Kit** is a set of tools that implement common behaviors and components with very unique interaction styles without being opinionated about the template choices. It is a kind of abstraction of the Angular Material Library, with no styling specific to material design. It provides more unique ways to get creative while building your angular components.

Drag and Drop Tool

The Drag and Drop tool is one of the component development kit common behaviors. It contains directives that enable really top notch drag and drop capabilities on component parts.

The `@angular/cdk/drag-drop` module provides you with a way to easily and declaratively create drag-and-drop interfaces, with support for free dragging, sorting within a list, transferring items between lists, animations, touch devices, custom drag handles, previews, and placeholders, in addition to horizontal lists and locking along an axis.

Before we start

Here is a list of 4 things we need in order to be able to use the Drag and Drop tool in our Angular 7 workspace.

- The latest version of Angular (version 7)

```
// run the command in a terminal

ng version
```

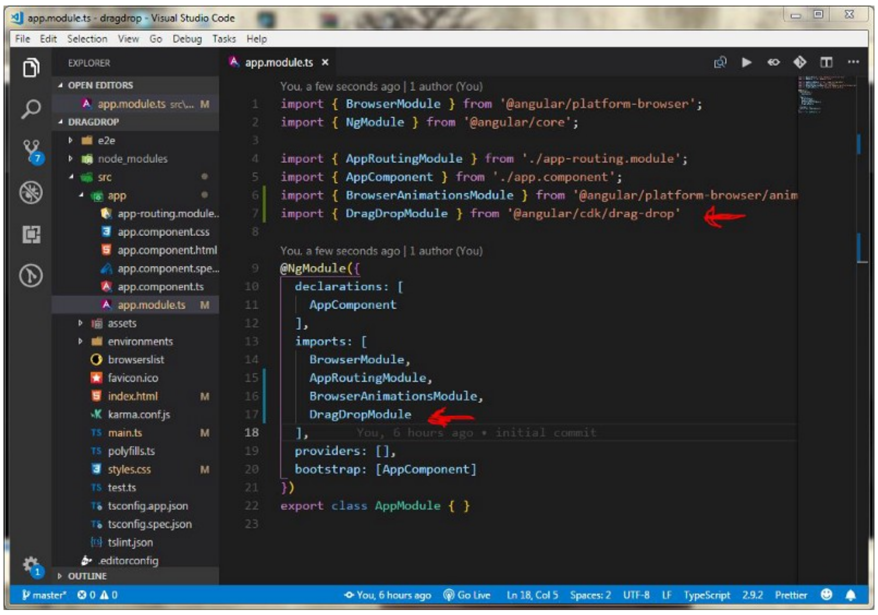
- An Angular 7 application

```
//cd to a preferred location

// run the command below

ng new dragdrop
```

- Angular Material installed on the above app

```
ng add @angular/material
```

- Import the Drag Drop module in the app module so it is available for use in the Angular application like so:


app.module.ts - dragdrop - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
EXPLORER
OPEN EDITORS
DRAGDROP
app.module.ts ×
master* ⊗ 0 ⚠ 0
You, 6 hours ago Go Live Ln 18, Col 5 Spaces: 2 UTF-8 LF TypeScript 2.9.2 Prettier

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/anim
import { DragDropModule } from '@angular/cdk/drag-drop'

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    DragDropModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Rendering a List

Drag and Drop works mostly on list items, let's create a small list of pop artists and then add the drag and drop functionality to it.

- Open the app component file and add an array of artists like so:

```
// hardcode the array of artists below
import { Component } from '@angular/core';

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

export class AppComponent {
  artists = [
    'Artist I - Davido',
    'Artist II - Wizkid',
    'Artist III - Burna Boy',
    'Artist IV - Kiss Daniel',
    'Artist V - Mayorkun',
    'Artist VI - Mr. Eazi',
    'Artist VII - Tiwa Savage',
    'Artist VIII - Blaqbonez',
```

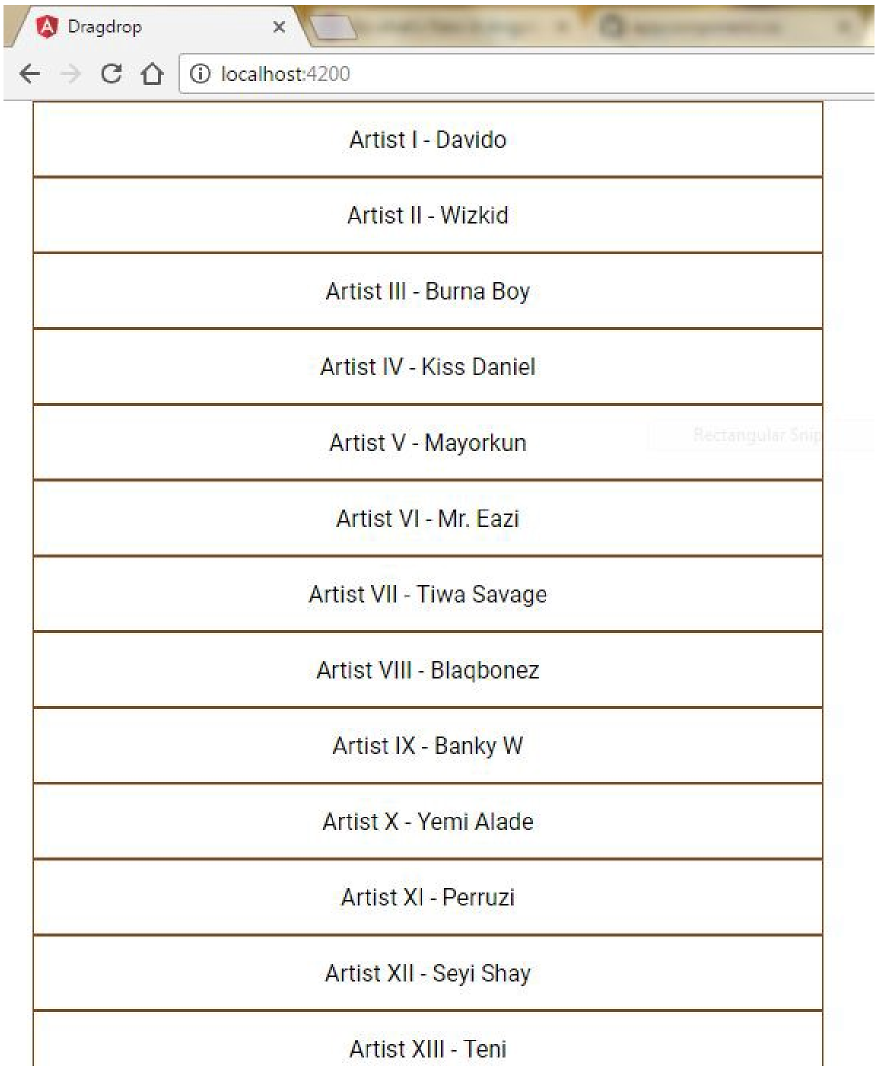
- In the app component HTML file, replace the boilerplate HTML with the ngFor command to render the list.

```
// replace the boilerplate code with this
<div  *ngFor="let artist of artists" >{{artist}}</div>
```

- Then add a little bit of styling in the app component CSS file like this:

```
// add a pop class to the html file first
.pop{
    height: 50px;
    width: 40%;
    margin-left: 20px;
    border: 1px solid saddlebrown;
    display: flex;
    justify-content: center;
```

We now have our list of pop artists styled and rendered the way it can be easily seen when we drag list items around.


Dragdrop

localhost:4200

Artist I - Davido

Artist II - Wizkid

Artist III - Burna Boy

Artist IV - Kiss Daniel

Artist V - Mayorkun

Artist VI - Mr. Eazi

Artist VII - Tiwa Savage

Artist VIII - Blaqbonez

Artist IX - Banky W

Artist X - Yemi Alade

Artist XI - Perruzi

Artist XII - Seyi Shay

Artist XIII - Teni

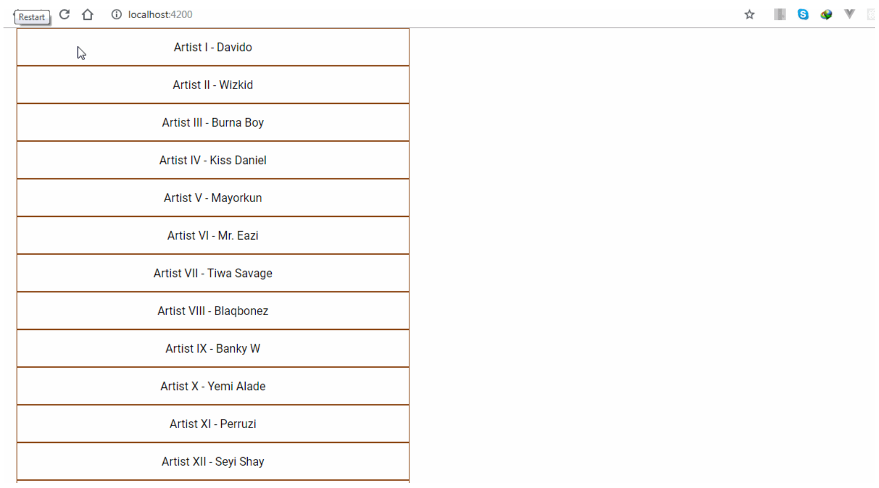
Now we have a list set up, we'll go through all the major aspects of the new drag and drop tool in the component development kit below:

- Basic Drag and Drop
- Sorting
- Moving Items between Lists
- Animations

Basic Drag and Drop

Just by adding the cdkDrag directive to a list item renderer, the list items becomes draggable.

```
<div *ngFor="let artist of artists" class="pop" cdkDrag>
{{artist}}</div>
```


Artist I - Davido

Artist II - Wizkid

Artist III - Burna Boy

Artist IV - Kiss Daniel

Artist V - Mayorkun

Artist VI - Mr. Eazi

Artist VII - Tiwa Savage

Artist VIII - Blaqbonez

Artist IX - Banky W

Artist X - Yemi Alade

Artist XI - Perruzi

Artist XII - Seyi Shay

We can see that at this stage it is draggable to anywhere in the DOM, the scope can be limited to the list by wrapping it in a cdkDropList div.

```
<div cdkDropList>

<div *ngFor=”let artist of artists” class=”pop” cdkDrag>
{{artist}}</div>

</div>
```

Sorting

Now that we can drag list items around the DOM or within a list, we can go further by re-ordering items or basically trying to move list items up and down a list. This involves listening for a cdkDropListDropped event and specifying moveItemsInArray method to bring the re-ordering to life.

- Update the **app.component.html** file with this:

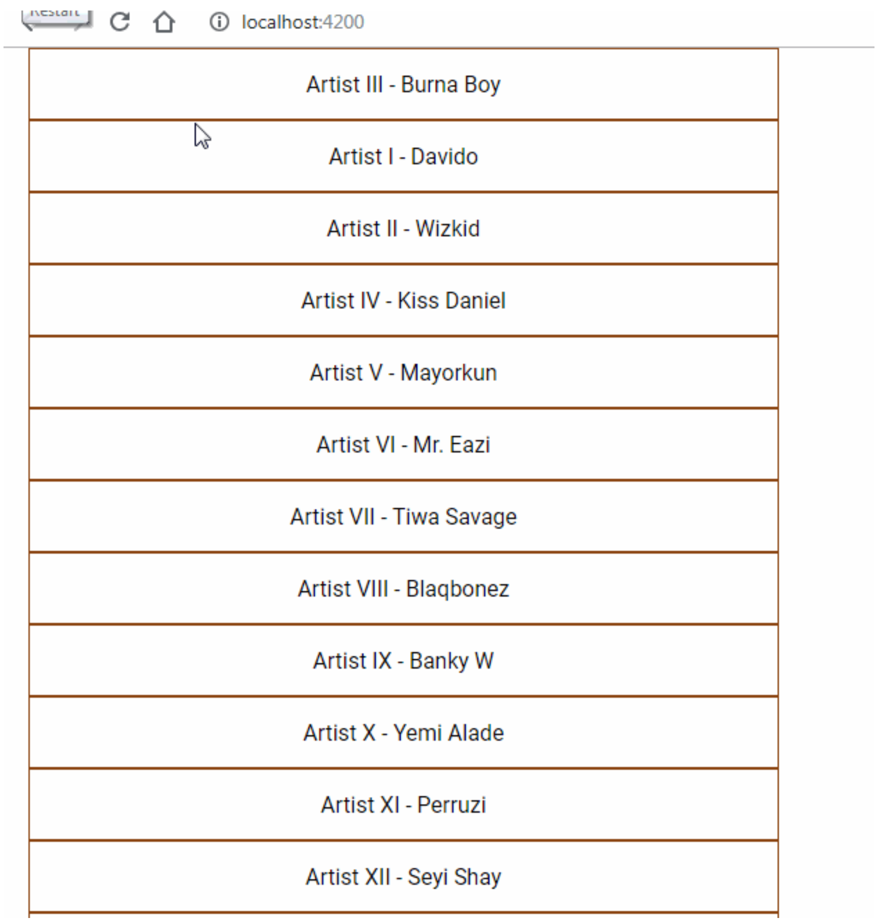
```
<div cdkDropList (cdkDropListDropped)=”drop($event)”>

<div *ngFor=”let artist of artists” class=”pop” cdkDrag>
{{artist}}</div>

</div>
```

- update the **app.component.ts** file with this

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

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

}
```

The MoveItemsInArray method takes 3 parameters:

- the list array where it would do the re-ordering work
- Event.previousIndex which is the previous location of the dragged item to drop.
- Event.CurrentIndex which is the current location where the dragged item is being dropped so it can do the swap.



Artist III - Burna Boy
Artist I - Davido
Artist II - Wizkid
Artist IV - Kiss Daniel
Artist V - Mayorkun
Artist VI - Mr. Eazi
Artist VII - Tiwa Savage
Artist VIII - Blaqbonez
Artist IX - Banky W
Artist X - Yemi Alade
Artist XI - Perruzi
Artist XII - Seyi Shay

> *Remember to import CdkDragDrop, moveItemInArray from angular/cdk/drag-drop*

-

Moving Items Between Lists

Isn’t this exciting? We can now re-order items on a list using drag and drop, taking it even a bit further let us try re-ordering again but this time we would make it work between 2 separate lists. With the cdkDropList directive we have been using, we can connect one or more cdkDropList elements (divs) together by wrapping them in an element with a cdkDropListGroup attribute or setting the cdkDropListConnectedTo property.

- In **app.component.html**

```
<div style=”display: flex”>

<div cdkDropList [cdkDropListData]=”artists”

[cdkDropListConnectedTo]=”secondList”
#firstList=”cdkDropList”

(cdkDropListDropped)=”drop($event)” style=”width: 30%”>

<div *ngFor=”let artist of artists” class=”pop” cdkDrag>
{{artist}}</div>

</div>

<div cdkDropList [cdkDropListData]=”alteArtists”

[cdkDropListConnectedTo]=”firstList”
#secondList=”cdkDropList”

(cdkDropListDropped)=”drop($event)” style=”width: 30%”>

<div *ngFor=”let artist of alteArtists” class=”pop” cdkDrag>
{{artist}}</div>

</div>

</div>
```

Breakdown of what we did:

1. We added a new droplist from line 7–11 where we tried to render a new list

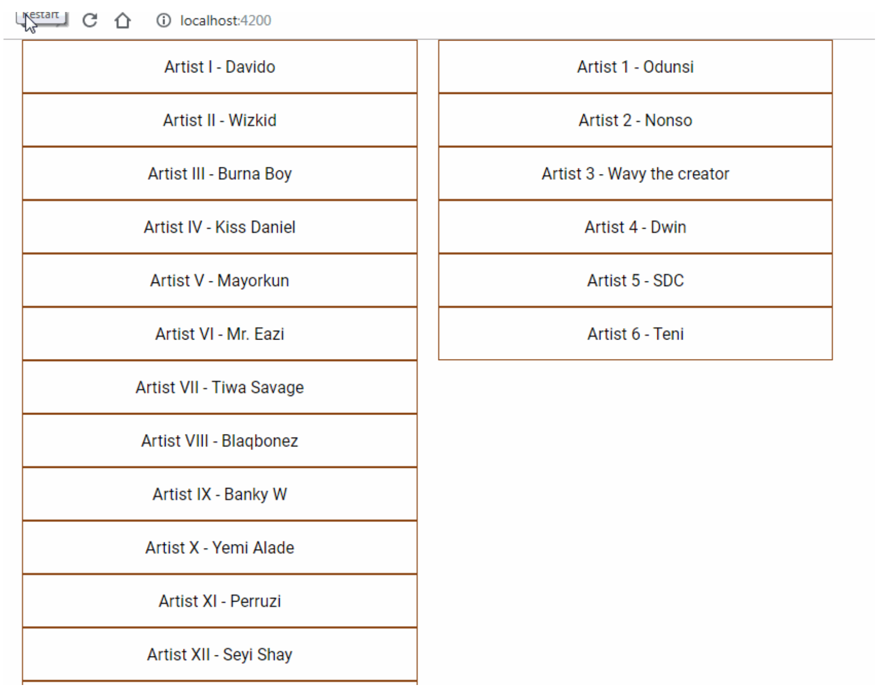
2. We wrapped the two droplist divs in a div and styled the display to flex so they span out standing separately
3. We specified the droplist data for each list on lines 2 and 7
4. We specified the list that each list is connected to in lines 3 and 8
5. Finally, we created IDs on the drop lists to tell Angular these are separate drop lists.

- In **app.component.ts**

```
alteArtists = [

‘Artist 1 — Odunsi’,

‘Artist 2 — Nonso’,

‘Artist 3 — Wavy the creator’,

‘Artist 4 — Dwin’,

‘Artist 5 — SDC’,

‘Artist 6 — Teni’

];

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

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

transferArrayItem(event.previousContainer.data,event.contain
er.data,

event.previousIndex, event.currentIndex)

} else {

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

}

}
```

Breakdown of what we did:

1. We created a new `alteArtists` array and filled it with data.
2. We added an if else statement to modify the moveinarray method.
3. The if part contains logic for when a list item is dragged into a separate drop list
4. The else part contains the old logic for re-ordering list items in its own list.



Artist I - Davido	Artist 1 - Odunsi
Artist II - Wizkid	Artist 2 - Nonso
Artist III - Burna Boy	Artist 3 - Wavy the creator
Artist IV - Kiss Daniel	Artist 4 - Dwin
Artist V - Mayorkun	Artist 5 - SDC
Artist VI - Mr. Eazi	Artist 6 - Teni
Artist VII - Tiwa Savage	
Artist VIII - Blaqbonez	
Artist IX - Banky W	
Artist X - Yemi Alade	
Artist XI - Perruzi	
Artist XII - Seyi Shay	

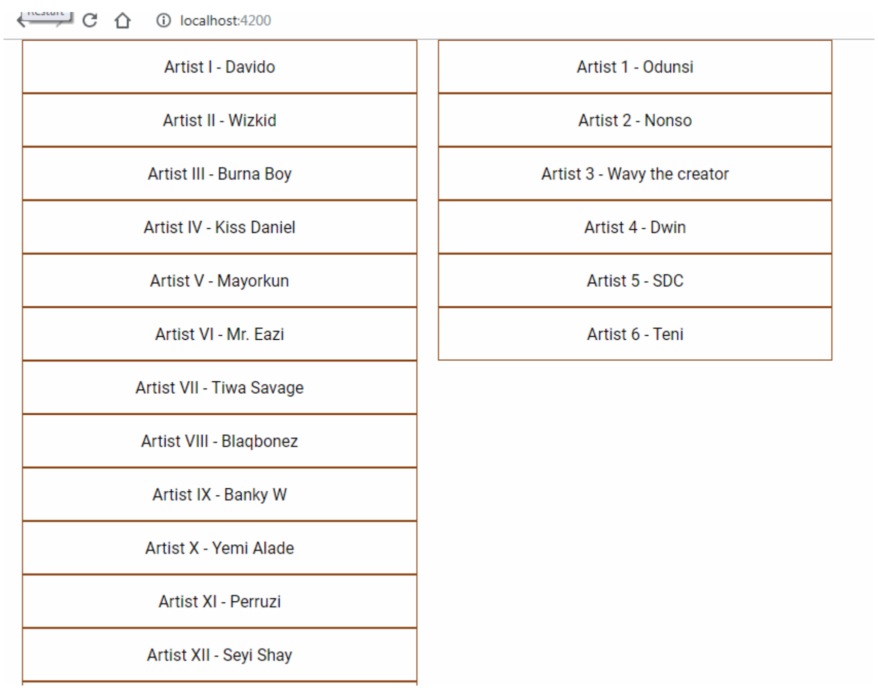
Animations

At this stage, we can choose to apply transition styling to the dragging and dropping process. The module supports animations applied to list items while sorting an element inside a list, as well as when dropping an element in a list. Let’s see a simple example:

- In **app.component.css**:

```
/* Animate items as they’re being sorted. */

.cdk-drop-list-dragging .cdk-drag {

transition: transform 250ms cubic-bezier(0, 0, 0.2, 1);

}
```

```
/* Animate an item that has been dropped. */

.cdk-drag-animating {

transition: transform 300ms cubic-bezier(0, 0, 0.2, 1);

}
```



localhost:4200

Artist I - Davido	Artist 1 - Odunsi
Artist II - Wizkid	Artist 2 - Nonso
Artist III - Burna Boy	Artist 3 - Wavy the creator
Artist IV - Kiss Daniel	Artist 4 - Dwin
Artist V - Mayorkun	Artist 5 - SDC
Artist VI - Mr. Eazi	Artist 6 - Teni
Artist VII - Tiwa Savage	
Artist VIII - Blaqbonez	
Artist IX - Banky W	
Artist X - Yemi Alade	
Artist XI - Perruzi	
Artist XII - Seyi Shay	

Maybe it's not quite smooth enough, but you can notice the transition effects compared to how it felt without transitions. If you followed the tutorial to this stage, your final app component folder should look exactly like these below:

- app.component.html

```
<div style="display: flex">
<div cdkDropList  [cdkDropListData]="artists"
[cdkDropListConnectedTo]="secondList" #firstList="cdkD
(cdkDropListDropped)="drop($event)" style="width: 30%"
<div  *ngFor="let artist of artists" class="pop" cdkDr
</div>
<div cdkDropList [cdkDropListData]="alteArtists"
[cdkDropListConnectedTo]="firstList" #secondList="cdkD
 (cdkDropListDropped)="drop($event)" style="width: 30%
```

- app.component.ts

```
import { Component } from '@angular/core';
import { CdkDragDrop, moveItemInArray, transferArrayIt

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

export class AppComponent {
  artists = [
    'Artist I - Davido',
    'Artist II - Wizkid',
    'Artist III - Burna Boy',
    'Artist IV - Kiss Daniel',
    'Artist V - Mayorkun',
    'Artist VI - Mr. Eazi',
    'Artist VII - Tiwa Savage',
    'Artist VIII - Blaqbonez',
    'Artist IX - Banky W',
    'Artist X - Yemi Alade',
    'Artist XI - Perruzi',
    'Artist XII - Seyi Shay',
    'Artist XIII - Teni'
  ];

  alteArtists = [
    'Artist 1 - Odunsi',
    'Artist 2 - Nonso'
```

- app.component.css

```
.pop{
    height: 50px;
    margin-left: 20px;
    border: 1px solid saddlebrown;
    display: flex;
    justify-content: center;
    align-items: center;

}
.cdk-drop-list-dragging .cdk-drag {
    transition: transform 250ms cubic-bezier(0, 0, 0.2
```

Conclusion

In this article we were introduced to the new Angular 7 CDK tool called drag and drop, we saw practical examples of how it can be used and the possibilities it affords for a wonderful user experience. You can take a look at the CDK docs here. This tutorial project would be open source on GitHub here so you can fork it and play around with it. I would like to hear your use case ideas for drag and drop in the comment section, Happy Coding!

Follow me here and do not forget to clap if you enjoyed reading this.