Component Name: Autocomplete with pagination

Author: pramod lad

Date: 5/3/2018

**Component Screen-Shot:**



**Prerequisites:**

1. Node JS 6.0 or later

**Steps to start the angular application:**

1. Open Command Prompt.
2. Go to the folder directory ‘Reusable Component - Autocomplete with pagination - Pramod Lad’.
3. Run ‘**npm install**’, after the all the components have being installed.
4. Run ‘**npm run ng serve**’ to compile the angular application.
5. Once successfully compiled it will start the app in <http://localhost:4200>.
6. And you can see the component 😊

**Component Features:**

* This component basically provides combined features of two functionalities, one is autocomplete and another one is pagination
* User can customize the array or assigned dynamic array
* The Autocomplete Pagination component will automatically calculate the pages depending on the given options.
* User can customize the no of items to be displayed as per choice using "pageSize" variable
* User can select only one item and selected item is displayed below the search text box.

**Component Description**:

* It gives the benefit of both the features Autocomplete and pagination
* Based on current search text, search results and pagination will get updated.
* Autocomplete results will be retrieved even with single letter search.
* When you select any item in search result box, the selected item gets populated in search text box.
* The Autocomplete Pagination component will automatically calculate the pages depending on the given options.
* User can customize the array or assigned dynamic array
* Usage: To use as per user requirement, there is flexibility to modify below mentioned parameters:1. Array - Required for Autocomplete functionality, 2. curPage - specifies the active page of pagination, 3. pageSize - specifies no of items need to show on each page

**Component HTML to be included:**

|  |
| --- |
| <app-autocomplete-pagination></app-autocomplete-pagination> |

**Angular Code for the Component:**

|  |
| --- |
| **autocomplete-pagination.component.html** |
| <div class="paginationWrapper">    <div class="container">  <!--autocomplete box-->  <input type="text" class="searchBox" [(ngModel)]="searchText" (keyup)="funcShowResultBox()" placeholder="search movie">  <!--autocomplete box-->  <!--resultbox starts-->  <div class="resultBoxWrapper" \*ngIf="showResultBox">  <!--autocomplete result list starts-->  <div \*ngFor="let movie of updatedMovies;let index =index; let isOdd=odd;let isEven=even" [class.odd]="isOdd"  [class.even]="isEven" class="resultItem" (click)="itemClick(movie.title)">  {{movie.title}}  </div>  <!--autocomplete result list ends-->  <div class="pagination-wrapper">  <div class="pagination">  <button [disabled]="curPage == 1" (click)="prev()" class="btnPrev">&laquo;</button>  <a \*ngFor="let page of getTotalNoOfPages(totalPages); let i=index;" class="active" (click)="pageClick(i)">  {{page}}  </a>    <button [disabled]="curPage >= totalPages" (click) ="next()" class="btnNext">&raquo;</button>  </div>  </div>  <!--pagination ends-->    </div>  <!--resultbox ends-->  <!--showing selected results-->  <div \*ngIf="!showResultBox">  <div class="selectedValue" \*ngIf="searchText">Selected movie name is <b>{{searchText}}</b></div>  <!--showing selected results-->  <div class="selectedValue" \*ngIf="!searchText">No movie selected</div>  </div>    </div>  </div> |

|  |
| --- |
| **autocomplete-pagination.component.css** |
| .odd{  background:#eee;  }  .even{  background:#fff;  }  .paginationWrapper .container{  position:relative;  width:auto;  margin:0px 10px;  }  .paginationWrapper .searchBox{  width: 100%;  height: 40px;  padding: 10px 15px;  box-sizing: border-box;  color:#000;  opacity:0.9;  font-size:14px;  line-height:14px;  outline:0px;  }  .paginationWrapper .resultBoxWrapper{  border: 1px solid #eee;  height: auto;  position:absolute;  width:100%;    }  .paginationWrapper .resultItem{  padding:10px 15px;  border-bottom: 1px solid #eee;  font-size: 14px;  line-height: 14px;  color: #000;  opacity: 0.9;  height:40px;  box-sizing:border-box;  }  .paginationWrapper .resultItem:first-letter{  text-transform:uppercase;  }  .resultBoxWrapper .pagination{  text-align:center;  position:relative;  }  .selectedValue{  margin:10px 0px 0px;  }  /\*---pagination starts here---\*/  .pagination-wrapper{  text-align:center;  margin:25px 0px;  }  .center {  text-align: center;  }    .pagination {  display: inline-block;  padding:0px 52px;  }    .pagination a {  color: black;  float: left;  padding: 8px 16px;  text-decoration: none;  transition: background-color .3s;  border: 1px solid #ddd;  margin: 0 4px;  }    .pagination a.active {  background-color: #4CAF50;  color: white;  border: 1px solid #4CAF50;  }    .pagination a:hover:not(.active) {background-color: #ddd;}  .pagination .btnPrev{  position:absolute;  left:0;  margin: 0px 4px 0px;  padding: 8px 16px;  }  .pagination .btnNext{  position:absolute;  right:0;  margin: 0px 4px 0px;  padding: 8px 16px;  }  /\*---pagination ends here---\*/ |

|  |
| --- |
| **autocomplete-pagination.component.ts** |
| import { Component, OnInit } from '@angular/core';  import { FilterPipe } from '../filter.pipe';  @Component({  selector: 'app-autocomplete-pagination',  templateUrl: './autocomplete-pagination.component.html',  styleUrls: ['./autocomplete-pagination.component.css'],  providers: [FilterPipe]  })  export class AutocompletePaginationComponent implements OnInit {  constructor(private filter: FilterPipe) { }  searchText;  showResultBox = false;//hide resultbox  //movies object array  movies = [  {  name: "jab we met",  title: "jab we met"  },  {  name: "mumbai pune mumbai",  title: "mumbai pune mumbai"  },  {  name: "dhkmn",  title: "dhkmn"  },  {  name: "arya",  title: "arya"  },  {  name: "barfi",  title: "barfi"  },  {  name: "banvabanvi",  title: "banvabanvi"  },  {  name: "Mohabattein",  title: "Mohabattein"  }  ];  updatedMovies = [];  curPage = 1; //current page in pagination  pageSize = 2; //total no of pages in pagination  totalPages;    numberOfPages() {  this.curPage = 1;  this.totalPages = Math.ceil(this.updatedMovies.length / this.pageSize); //returns total pages  };  getTotalNoOfPages(totalPages){  let totalPageArray = [];  for(let i=1; i<=totalPages;i++){  totalPageArray.push(i);  }  return totalPageArray;  }  totalNoOfPages = [1,2,3,4];    funcShowResultBox = function () {  this.showResultBox = true; //shows resultbox    this.updatedMovies = this.filter.transform(this.movies, this.searchText);  this.numberOfPages();  this.updatedMovies = this.updatedMovies.slice(((this.curPage \* this.pageSize) - this.pageSize), (this.curPage \* this.pageSize))    }    itemClick = function (itemValue) {  this.searchText = itemValue;  this.showResultBox = false;  }  prev() {  this.curPage = this.curPage - 1;  this.updatedMovies = this.filter.transform(this.movies, this.searchText).slice(((this.curPage \* this.pageSize) - this.pageSize), (this.curPage \* this.pageSize));  }  next() {  this.curPage = this.curPage + 1;  this.updatedMovies = this.filter.transform(this.movies, this.searchText).slice(((this.curPage \* this.pageSize) - this.pageSize), (this.curPage \* this.pageSize));  }  pageClick(index){  this.curPage=index+1;//as index starts with 0, need to increment it by 1  this.updatedMovies = this.filter.transform(this.movies, this.searchText).slice(((this.curPage \* this.pageSize) - this.pageSize), (this.curPage \* this.pageSize));  console.log(this.updatedMovies);  }  ngOnInit() {  }  } |

|  |
| --- |
| **app.module.ts** |
| import { NgModule } from '@angular/core';  import { FormsModule } from '@angular/forms';  import { BrowserModule } from '@angular/platform-browser';  import { AppComponent } from './app.component';  import { FilterPipe }from './filter.pipe';  import { AutocompletePaginationComponent } from './autocomplete-pagination/autocomplete-pagination.component';  @NgModule({  declarations: [  AppComponent, FilterPipe, AutocompletePaginationComponent  ],  imports: [  BrowserModule,  FormsModule  ],  providers: [],  bootstrap: [AppComponent]  })  export class AppModule { } |