A javaScript library which supports an HTML 5 Data Table with paging, using Handlebars.js and Bootstrap.

# Features

1. Provides Paging movement controls:
   * First, Previous, Next, Last
2. Allows control of Page Size via a control
   * Dropdown list with multiple page size options, including **All**
3. Displays Paging information
   * Showing 1 to X of Y entries
4. Provides a client-side Searching, but only for data in the table (i.e. doesn’t pass search to backend data server)
5. Allows Header columns to be sortable
   * Sorting done server-side
   * Support for Date, Integer, and String data
   * Passes sort id and direction to server-side
6. Allows Header columns to be filterable
   * Filtering done server-side
   * Support for String data (future support for integer and datetime)
   * Passes filter column and value to server-side
7. Filtering
   * Use optionally by clicking on filter icon to open the filter controls
   * Ability to clear the filter
   * Submit filter via button
   * Hiding filter controls resets the filter column and value (i.e. ignore them)
   * Provide a flag (showFilter) for enabling / disabling the filtering functionality
8. Integrates seamlessly with Handlebars
   * Can populate data asynchronously via Handlebars template
   * Perhaps register the Handlebars template and update functions
   * Passes Page, Page Size, Sort Column, Sort Direction, Filter Column, and Filter Value to backend data server
   * Receives data plus total number of entries
9. Provides ‘spinner’ for displaying data loading asynchronously
10. Displays default message when there are no records to display
11. Uses Bootstrap 4 for styling
12. Needs to be responsive:
    * Columns are intelligently wrapped
    * Supports hidden columns

# Implementation

1. Written in javaScript
2. Uses a minimum of jQuery 3.3.1
3. Use javaScript prototypes (why?)
4. Try to hide the private helper functions
5. Throw exception where appropriate
6. Return itself, so that dot-operator will work
7. Dot-operator must also return itself, so object chaining will work
8. Eventually, control its scope so more than one can be on a page (future)
9. Locates the HTML table in the DOM and creates the necessary elements dynamically
   * Paging movement controls
   * Page Size control
   * Paging information
   * Filtering column
   * Filtering value
   * Sortable column headers
   * Filterable column headers
   * Note: Column header Ids must be defined in the HTML table
10. Defaults:
    * Page: 1
    * PageSize: 10
    * SortColumn: Id
    * SortDirection: ‘asc’
    * FilterColumn = ‘’
    * FilterValue = ‘’
    * showFilter = false

# Interface

1. Requires a HTML table definition
   * Heading row must be classed

<table id=”CustomerTable” ctTotalItems=”100” class=”table table-striped tabled-bordered table-condensed table-hover”>

<thead>

<tr class=”colHeaders”>

<th id=”Id” class=”sortable”> <a href="#">ID</a></th>

<th id=”FirstName” class=”sortable filterable”><a href="#">First Name</a></th>

<th id=”LastName” class=”sorable filterable”> <a href="#">Last Name</a></th>

<th id=”Email” class=”sortable filterable”><a href="#">E-Mail</a></th>

</tr>

</thead>

<tbody id=”CustomerList”>

</tbody>

</table>

1. Instantiation

var pager = new TablePagerLib( {

tableId: ‘CustomerTable’,

totalItemsAttribute: ‘ctTotalItems’,

page: 1,

pageSize: 10,

pageSizeOptions: [[10, 25, 50, -1],[10, 25, 50, ‘All’]],

sortColumn: ‘Id’,

sortDirection: ‘asc’,

showFilter: true,

columnHeaderClass: ‘colHeaders’,

spinnerRef: ‘/images/spinner-128.gif’

});

1. Data Binding

pager.setDataBinding( function() {

//Do jQuery ajax and Handlebars.js stuff here

pager.endDataBinding(results.totalItems);

});

1. Test

# Layout

<div class=”row”>

<div class=”col-4”>

<Filter Icon>

<Label>Showing</Label>

</div>

<form class=”col-8”>

<Input placeholder=”Search” />

</form>

</div>

<div class=”row hidden”>

<Filter Field Dropdown>

<Filter Value Input>

<clear button>

<filter button>

</div>

<table>

</table>

<div class=”row”>

# To Dos

1. Add code blocks to README for sample configuration

# Issues

1. Control onclick/change events can’t find the library instance.
   * Possible solution is to register each instance by name in a hash table stored in the windows scope.
   * At least partial solution by using **addEventListener** and using self (var self = this;)
2. Spinner needs to compute it’s top and left locations