Getting Started

REQUIRED: JQuery and Bootstrap

Add reference to HTML head:

<head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<link rel="stylesheet" href="https://drive.google.com/uc?export=download&id=1VO47Uxij5M1sLr6fNFh\_0TiDhmsDLj\_P">

<script

src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script

src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

<script src="https://drive.google.com/uc?export=download&id=1tR3lgLek\_r3NdfRG1WmnjLXMpuTzBZJG"></script>

<!-- Use this file (myJS.js) for examples -->   
<script src="myJS.js"></script>  
</head>

TABLE

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Description | Type |
| Data | Jason data | required |
| TableID | Table id if you want to assign specific id to table | optional |
| AppendTo | HTML Element ID to append table to, by default table will be append to <body> | optional |
| DefaultHeader | By default show the default header |  |
| CustomerHeader | Set column(s) to display and custom header | optional |
| AddToColumn | Append or Prepend value to column | optional |
| Sort | By default it is On |  |

1. Add div tag to html body

<div id="divTable"></div>

1. Create JavaScript file myJS.js and add reference to head. Use this file (myJS.js) for examples

User the following JSON for example. Add the following example to myJS.js

var myObject = [{

"CustomerName": "Alfreds Futterkiste",

"City": "Berlin",

"Country": "Germany"

},

{

"CustomerName": "Ana Trujillo Emparedados y helados",

"City": "México D.F.",

"Country": "Mexico"

},

{

"CustomerName": "Antonio Moreno Taquería",

"City": "México D.F.",

"Country": "Mexico"

},

{

"CustomerName": "Around the Horn",

"City": "London",

"Country": "UK"

},

{

"CustomerName": "B's Beverages",

"City": "London",

"Country": "UK"

},

{

"CustomerName": "Berglunds snabbköp",

"City": "Luleå",

"Country": "Sweden"

},

{

"CustomerName": "Blauer See Delikatessen",

"City": "Mannheim",

"Country": "Germany"

},

{

"CustomerName": "Blondel père et fils",

"City": "Strasbourg",

"Country": "France"

},

{

"CustomerName": "Bólido Comidas preparadas",

"City": "Madrid",

"Country": "Spain"

},

{

"CustomerName": "Bon app'",

"City": "Marseille",

"Country": "France"

},

{

"CustomerName": "Bottom-Dollar Marketse",

"City": "Tsawassen",

"Country": "Canada"

},

{

"CustomerName": "Cactus Comidas para llevar",

"City": "Buenos Aires",

"Country": "Argentina"

},

{

"CustomerName": "Centro comercial Moctezuma",

"City": "México D.F.",

"Country": "Mexico"

},

{

"CustomerName": "Chop-suey Chinese",

"City": "Bern",

"Country": "Switzerland"

},

{

"CustomerName": "Comércio Mineiro",

"City": "São Paulo",

"Country": "Brazil"

}

];

1. Called table function inside JQuery document ready function

$(function() {

//Create instance of JSON To Html

var x = new jsonToHtml();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\* EXAMPLE-1 (BASIC) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

x.Table({

Data: myObject,

AppendTo: 'divTable' //div created in step 1

});

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\* EXAMPLE-2 (CUSTOMHEADER) \*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Adding customer header to display or change single or multiple

//column name

/\*

1. WILL DISPLAY ONY ONE COLUMN ‘CustomerName’ with ‘Name’ header ---

UN-Comment the following code to for to display one column with custom header name

\*/

var customHeader = [

{ 'orginalColumnName': 'CustomerName', 'newColumnName': 'Name' }

];

/\*

2. FOLLOWING WILL SHOW ALL COLUMN

UN-Comment the following code to display all column with custom header name

\*/

var customHeader = [

{ 'orginalColumnName': 'CustomerName', 'newColumnName': 'Name' },

{ 'orginalColumnName': 'City', 'newColumnName': 'Customer City Name' },

{ 'orginalColumnName': 'Country', 'newColumnName': 'Country' }

];

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**NOTE**: USE OF THE OBJECT(customerHeader) FORM THE ABOVE and COMMENT THE OTHER.

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

x.Table({

Data: myObject,

AppendTo: 'divTable', //div created in step 1

DefaultHeader: false, //MUST BE FALSE

CustomHeader: customHeader

});

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* EXAMPLE-3 (APPEND OR PREPAND VALYE) \*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

Append or Prepend value to Column(s)

\*/

var addToColumn = [

{ 'ColumanName': 'CustomerName', 'Value': 'Name: ', 'Type': 'prepend' },

{ 'ColumanName': 'City', 'Value': '(city name)', 'Type': 'append' }

];

x.Table({

Data: myObject,

AppendTo: 'divTable', //div created in step 1

AddToColumn: addToColumn

});

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\* EXAMPLE-4 (HEADING STYLE)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

BY USING HEADINGSTYLE FUNCTION WILL CHANGE THE TABLE HEADER FORECOLOR AND

BACKGROUND COLOR

\*/

x.Table({

Data: myObject,

TableID: 'tbTest',

AppendTo: 'divTable', //div created in step 1

}).HeadingStyle({

TableID: 'tbTest',

BackgroundColor: 'black',

Forecolor: 'white'

});

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\* EXAMPLE-5 (TABLE STYLE)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

BY USING TABLESTYLE FUNCTION WILL CHANGE THE TABLE FORECOLOR AND BACKGROUND COLOR

\*/

x.Table({

Data: myObject,

TableID: 'tbTest',

AppendTo: 'divTable', //div created in step 1

}).HeadingStyle({

TableID: 'tbTest',

BackgroundColor: 'lightyellow',

Forecolor: 'black'

}).TableStyle({

TableID: 'tbTest',

BackgroundColor: 'black',

Forecolor: 'lightblue'

});

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\* EXAMPLE-6 (SEARCH) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

BY USING SEARCH FUNCTION SEARCH WITH IN TABLE OR WITHIN SPECIFIC COLUMN(s)

ADD <input id="txtSearch" type="text" class="form-control" placeholder="search...">

to your page

1. SEARCH WITH IN WHOLE TABLE

\*/

x.Table({

Data: myObject,

TableID: 'tbTest',

AppendTo: 'divTable', //div created in step 1

}).Search({

TableID: 'tbTest',

SearchTextBoxID: 'txtSearch'

});

/\*

2. SEARCH WITH IN SPECIFIC COLUMN(s)

IN THIS EXAMPLE WE ARE SEARECH WITH city and Country column

\*/

x.Table({

Data: myObject,

TableID: 'tbTest',

AppendTo: 'divTable', //div created in step 1

}).Search({

TableID: 'tbTest',

SearchTextBoxID: 'txtSearch',

SearchInColumns: ['City', 'Country']

});

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\* EXAMPLE-7 (PAGING)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

BY USING PAGING FUNCTION ADD PAGE TO TABLE

\*/

x.Table({

Data: myObject,

TableID: 'tbTest',

AppendTo: 'divTable', //div created in step 1

}).Paging({

TableID: 'tbTest',

RowsPerPage: 5

});

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* EXAMPLE-8 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

BY USING PRINT FUNCTION YOU CAn EXPORT HTML TABLE TO CSV OR PDF

1. ADD button to HTML

<button class="btn btn-sm btn-primary" onclick="printCSV();">CSV</button>

CREATE function printCSV ASSUMPTION: table is avaiable in HTML

\*/

function printCSV() {

var p = new jsonToHtml();

p.Print({

//TableID: 'tbTest',

Print: 'CSV'

});

}

LIST

/\*

BY USING LIST FUNCTION CREATE THE LIST

Where TEXT is the column name in the JSON object

in the following example: list of 'CITY' form the

JSON object

Add the unorder list to HTML

<ul id="unorderList"></ul>

\*/

x.List({

Data: myObject,

AppendTo: 'unorderList',

Text: 'City'

});

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* FILTER LIST \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

BY USING FILTER FUNCTION, FILTER THE LIST

1. Create input for filter in HTML

<input type="text" class="form-control" **oninput="filterResult('#unorderList',**

**'li', this.value)"** placeholder="filter list ...">

CREATE function filterResult ASSUMPTION: list is avaiable in HTML

<ul id="unorderList"></ul>

\*/

function filterResult(controlName, li, value) {

var x = new jsonToHtml();

x.Filter(controlName, li, value);

}

DROPDOWN

/\*

REQUIRED

Data: JSON OBJECT

AppendTo: CONTROL NAME

Text: NAME OF THE COLUM

OPTIONAL

Value: NAME OF THE COLUMN

1. Create SELECT TAG in HTML

<select id="ddlTest" class="form-control">

<option value="-1">Select...</option>

</select>

In the following Example Text:'CITY' will get all the Cities from the

JSON object and bind to dropdown

\*/

x.Dropdown({

Data: myObject,

AppendTo: 'ddlTest',

Text: 'City'

});

RADIO BUTTON

/\*

REQUIRED

Data: JSON OBJECT

AppendTo: CONTROL NAME

Text: NAME OF THE COLUM

OPTIONAL

Value: NAME OF THE COLUMN

GroupName: Name of the radio b utton list

Direction: 'Vertical' By Default is horizontal

1. Create DIV TAG in HTML

<div id="divRadioButton"></div>

In the following Example Text:'CITY' will get all the Cities from the

JSON object and bind to Radio button

\*/

x.Radio({

Data: myObject,

AppendTo: 'divRadioButton',

Text: 'City'

});

CHECKBOX

/\*

REQUIRED

Data: JSON OBJECT

AppendTo: CONTROL NAME

Text: NAME OF THE COLUM

OPTIONAL

Value: NAME OF THE COLUMN

GroupName: Name of the radio b utton list

Direction: 'Vertical'

By Default is horizontal

1. Create DIV TAG in HTML

<div id="divCheckbox"></div>

In the following Example Text:'CITY' will get all the Cities from the

JSON object and bind to Radio button

\*/

x.Checkbox({

Data: myObject,

AppendTo: 'divCheckbox',

Text: 'City',

Direction: 'Vertical'

});