Dhruvil Dobariya

dhruvildobariya21@gmail.com

Abstract

In this module I learn basic about DMS controls

DMS Controls

Documentation



Index

[1 Basic Controls 1](#_Toc136514937)

[1.1 Button 1](#_Toc136514938)

[1.2 Textbox 2](#_Toc136514939)

[1.3 Numberbox 2](#_Toc136514940)

[1.4 Radio Button 3](#_Toc136514941)

[1.5 Checkbox 4](#_Toc136514942)

[2 Date Picker 5](#_Toc136514943)

[2.1 Introduction 5](#_Toc136514944)

[3 Select Picker 7](#_Toc136514945)

[3.1 Introduction 7](#_Toc136514946)

[4 jTable 11](#_Toc136514947)

[4.1 Introduction 11](#_Toc136514948)

1. Basic Controls

# Button

**Syntax:**

<form *action*="" *id*="form">

    <button *id*="" *class*="">Click me</button>

*<!-- Or -->*

    <input *type*="button" *value*="">

</form>

**Attributes:**

|  |  |
| --- | --- |
| Attribute | Description |
| Id | set id of button |
| class | set class of button |
| name | set name of button |
| type | set button type, we have three type of button like, button, submit and reset |
| value | set button value, it will only work with input type button |
| autofocus | set focus on page load |
| disabled | set disable button |
| form | specify action, where we want to post data, it will work if form type id submit |
| formaction | specify action, where we want to post data, it will work if form type id submit |
| formenctype | set encoded on form data which we send to the server, we have diffrent option like, application/x-www-form-urlencoded, multipart/form-data, text/plain |
| formmethod | set method of form, we have two type of method, get and post |
| formnovalidation | specify that form should not validate on submitting form, it only work with submit type |
| formtarget | specify where to display response after submitting the button, it will only work with submit type, we have different option like,   * \_blank: loads the response in a new window/tab * \_self: loads the response in a new window/tab (this is default) * \_parent: loads the response in the parent frame * \_top: loads the response in the full body of the window * <framename>: loads the response in a named iframe |

# Textbox

**Syntax:**

<form *action*="">

    <input *type*="text" *name*="city" *id*="" *class*="" *value*="" *placeholder*="" />

</form>

**Attributes:**

|  |  |
| --- | --- |
| Attribute | Description |
| Id | set id of button |
| class | set class of button |
| name | set name of button |
| type | set button type, we have three type of button like, button, submit and reset |
| value | set button value, it will only work with input type button |
| autofocus | set focus on page load |
| disabled | set disable button |
| dirname | set when form is submit direction of input is also include in data, ex: city=Los Angeles&city.dir=ltr |
| list | set id of data list which help in suggestions |
| required | set field is required |
| minlength | set minimum character which user must input |
| maxlength | set maximum character which user must input |
| form | specify action, where we want to post data, it will work if form type id submit |
| formnovalidation | specify that form should not validate on submitting form, it only work with submit type |

# Numberbox

**Syntax:**

<form *action*="">

    <input *type*="number" *name*="city" *id*="" *class*="" *value*="" *placeholder*="" />

</form>

**Attributes:**

|  |  |
| --- | --- |
| Attribute | Description |
| Id | set id of button |
| class | set class of button |
| name | set name of button |
| type | set button type, we have three type of button like, button, submit and reset |
| value | set button value, it will only work with input type button |
| autofocus | set focus on page load |
| disabled | set disable button |
| dirname | set when form is submit direction of input is also include in data, ex: city=Los Angeles&city.dir=ltr |
| list | set id of data list which help in suggestions |
| required | set field is required |
| min | set minimum value of input |
| max | set maximum value of input |
| step | set step for input |
| form | specify action, where we want to post data, it will work if form type id submit |
| formnovalidation | specify that form should not validate on submitting form, it only work with submit type |

# Radio Button

**Syntax:**

<form *action*="">

    <input *type*="radio" *name*="gender" *id*="male" *class*="" *value*="male" />

    <label *for*="male">Male</label>

    <input *type*="radio" *name*="gender" *id*="female" *class*="" *value*="female" />

    <label *for*="female">Female</label>

</form>

**Attributes:**

|  |  |
| --- | --- |
| Attribute | Description |
| Id | set id of button |
| class | set class of button |
| name | set name of button |
| type | set button type, we have three type of button like, button, submit and reset |
| value | set button value, it will only work with input type button |
| autofocus | set focus on page load |
| disabled | set disable button |
| checked | checked radio button |
| form | specify action, where we want to post data, it will work if form type id submit |
| formnovalidation | specify that form should not validate on submitting form, it only work with submit type |

# Checkbox

**Syntax:**

<form *action*="">

    <input *type*="checkbox" *name*="hobbies" *id*="playingCricket" *class*="" *value*="Playing Cricket" />

    <label *for*="playingCricket">Playing Cricket</label>

    <input *type*="checkbox" *name*="hobbies" *id*="playingFootball" *class*="" *value*="Playing Football" />

    <label *for*="playingFootball">Playing Football</label>

    <input *type*="checkbox" *name*="hobbies" *id*="playingChess" *class*="" *value*="Playing Chess" />

    <label *for*="playingChess">Playing Chess</label>

    <input *type*="checkbox" *name*="hobbies" *id*="readingBook" *class*="" *value*="Reading Book" />

    <label *for*="readingBook">Reading Book</label>

</form>

**Attributes:**

|  |  |
| --- | --- |
| Attribute | Description |
| Id | set id of button |
| class | set class of button |
| name | set name of button |
| type | set button type, we have three type of button like, button, submit and reset |
| value | set button value, it will only work with input type button |
| autofocus | set focus on page load |
| disabled | set disable button |
| checked | checked checkbox |
| form | specify action, where we want to post data, it will work if form type id submit |
| formnovalidation | specify that form should not validate on submitting form, it only work with submit type |

1. Date Picker

# Introduction

* It is provide by jQuery.
* So when we use datepicker we must first include jQuery then we must include jQuery ui.

**Example:**

<!DOCTYPE *html*>

<html *lang*="en">

    <head>

        <meta *charset*="UTF-8" />

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

        <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

        <title>Document</title>

        <link *rel*="stylesheet" *href*="../../assets/lib/jquery-ui-1.13.2/jquery-ui.min.css" />

    </head>

    <body>

        <div>

            <input *type*="text" *id*="datepicker" />

        </div>

        <script *src*="../../assets/lib/jQuery/jquery-3.6.3.min.js"></script>

        <script *src*="../../assets/lib/jquery-ui-1.13.2/jquery-ui.min.js"></script>

<script>

            $(document).ready(*function* () {

                $("#datepicker").datepicker();

            });

        </script>

    </body>

</html>

**Options:**

|  |  |
| --- | --- |
| Option | Description |
| selectOtherMonths | enable to select date of before or after month in current mount menu |
| showButtonPanel | Display button bar |
| changeMonth | Display month menu |
| changeYear | Display year menu |
| numberOfMonths | Display multiple month |
| showOn | Show button, when we click it, datepicker will open |
| buttonImage | Set button image |
| buttonImageOnly | Set button is image only or not in boolean |
| buttonText | Set button text |
| minDate | Set minimum date |
| maxDate | Set maximum date |
| showWeek | Show number of week of year |
| firstDay | Set first day of month |
| dateFormat | Set date format |
| showAnim | Set animation, we have many animation options like, show, slideDown, fadeIn, blind, bounce, clip, drop, fold, slide |

1. Select Picker

# Introduction

* Select Picket provides by bootstrap.
* So, we must first include bootstrap after we should include selectpicker.

**Example:**

<!DOCTYPE *html*>

<html *lang*="en">

    <head>

        <meta *charset*="UTF-8" />

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

        <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

        <title>Document</title>

        <link *rel*="stylesheet" *href*="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css" *integrity*="sha384-xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N" *crossorigin*="anonymous" />

        <link *rel*="stylesheet" *href*="../../../assets/lib/bootstrap-select-1.13.14/dist/css/bootstrap-select.min.css" />

        <link *rel*="stylesheet" *href*="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.2/font/bootstrap-icons.css" *integrity*="sha384-b6lVK+yci+bfDmaY1u0zE8YYJt0TZxLEAFyYSLHId4xoVvsrQu3INevFKo+Xir8e" *crossorigin*="anonymous" />

    </head>

    <body>

        <div *class*="mt-3">

            <select *name*="" *id*="select" *title*="Select City" *multiple*>

                <option *value*="1">Channai</option>

                <option *value*="2">Ahmedabad</option>

                <option *value*="3">Bangalore</option>

                <option *value*="4">Los Angeles</option>

                <option *value*="5">New York</option>

                <option *value*="6">Las Vegas</option>

                <option *value*="7">City of Miami</option>

            </select>

        </div>

        <div *class*="mt-3 ml-3">

            <button *class*="btn btn-danger" *id*="hide">Hide</button>

            <button *class*="btn btn-success" *id*="show">Show</button>

        </div>

        <script *src*="../../../assets/lib/jQuery/jquery-3.6.3.min.js"></script>

        <script *src*="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js" *integrity*="sha384-Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D6OP9dO5Vg3Q9ct" *crossorigin*="anonymous"></script>

        <script *src*="../../../assets/lib/bootstrap-select-1.13.14/dist/js/bootstrap-select.min.js"></script>

        <script>

            $(document).ready(*function* () {

                $("#select").selectpicker();

            });

        </script>

    </body>

</html>

**Options:**

|  |  |
| --- | --- |
| Option | Description |
| actionsBox | set selectAll and deselectAll button |
| countSelectedText | set format of count |
| deselectAllText | deselect all selected item when dropdown rendered |
| dropdownAlignRight | set all the option align right side, we also have one more option which is "auto" |
| dropupAuto | first it check upper or lower side have enough space or not, if have then it open normally else it open in dropup |
| header | set header of dropdown |
| hideDisabled | remove all element which is disable, also remove disable groups |
| iconBase | set icon base |
| liveSearch | set live search |
| liveSearchNormalize | set accent-insensitive search |
| liveSearchPlaceholder | set placeholder in search box |
| liveSearchStyle | set search style, We have two options like, When set to 'contains', searching will reveal options that contain the searched text. For example, searching for pl with return both Apple, Plum, and Plantain. When set to 'startsWith', searching for pl will return only Plum and Plantain. |
| maxOptions | set max number of selection |
| mobile | set mobile view |
| multipleSeparator | set seperator between multiple |
| noneSelectedText | The text that is displayed when a multiple select has no selected options, it show it's output when we don't give title to dropdown |
| selectedTextFormat | set display format, we have many options like,   * value: give comma seperated value * count: give count of selected value * count > 2: give value until condition become true, after it give count * static: always so placeholder |
| selectAllText | show text when we select all the option of dropdown |
| selectOnTab | treat tab like enter or space character in dropdown |
| showContent | enable render content inside dropdown elements |
| showIcon | enable to render icon in dropdown options |
| showSubtext | enable to render subtext in dropdown options |
| showTick | enable to render tick on selected options |
| size | set how many element show in window when window is open, if dropdown have more window then it enable scrollbar in window |
| style | set style |
| styleBase | set base style |
| tickIcon | set which icon we want to use as tick |
| title | set title of dropdown |
| width | set width, we have different option like, auto, fit, 100%, 100px |
| windowPadding | set padding in widow, we also can pass array like [10, 20, 20, 40] which represent [top, right, bottom, left] |
| sanitize | set sanitize, which sanitize data-content of options |
| whiteList | set whishList, it take object which contains allow attributes and tags |
| sanitizeFn | set sanitize function, it is used for dedicated library which use to parform sanitization. |

**Methods:**

|  |  |
| --- | --- |
| Methods | Description |
| val | select one or more based on value |
| render | It will render the selectpicker |
| refresh | It will refresh the selectpicker |
| selectAll | it will select all options |
| deselectAll | it will deselect all options |
| mobile | enable mobile |
| setStyle | set button style |
| toggle | open/close dropdown |
| hide | hide dropdown |
| show | show dropdown |
| destroy | destroy dropdown |

**Events:**

|  |  |
| --- | --- |
| Event | Description |
| show.bs.select | immediately triggered when dropdown show event start |
| shown.bs.select | triggered when dropdown show event end |
| hide.bs.select | immediately triggered when dropdown hide event start |
| hiddem.bs.select | triggered when dropdown hide event end |
| refresh.bs.select | triggered after dropdown is refresh |
| rendered.bs.select | triggered after dropdown is rendered |
| loaded.bs.select | triggered after select has been initialized |
| changed.bs.select | triggered after select value is changed |

1. jTable

# Introduction

* jTable provides by jQuery.
* So when we use jTable we must first include jQuery then we include jTable.

**Example:**

<!DOCTYPE *html*>

<html *lang*="en">

    <head>

        <meta *charset*="UTF-8" />

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

        <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

        <title>Document</title>

        <link *rel*="stylesheet" *href*="../../assets/lib/jquery-ui-1.13.2/jquery-ui.min.css" />

        <link *rel*="stylesheet" *href*="../../assets/lib/jtable/jtable.2.4.0/themes/metro/darkgray/jtable.min.css" />

    </head>

    <body>

        <div *id*="table"></div>

        <script *src*="../../assets/lib/jQuery/jquery-3.6.3.min.js"></script>

        <script *src*="../../assets/lib/jquery-ui-1.13.2/jquery-ui.min.js"></script>

        <script *src*="../../assets/lib/jtable/jtable.2.4.0/jquery.jtable.min.js"></script>

        <script *src*="../../assets/lib/jtable/jtable.2.4.0/jquery.jtable.min.js"></script>

*<!-- <script src="./custom.js"></script> -->*

        <script>

            $(document).ready(*function* () {

                $("#table").jtable({

*// general options*

                    actions: {

                        listAction: *function* (*postData*, *jtParams*) {

*return* {

                                Result: "OK",

                                Records: [{ Username: "Dhruvil Dobariya", Email: "dhruvildobariya@gmail.com", Password: "Dhruvil@123" }],

                            };

                        },

                    },

                    fields: {

                        Username: {

                            title: "Username",

                        },

                        Email: {

                            title: "Email",

                        },

                        Password: {

                            title: "Password",

                        },

                    },

*// Event handlers*

                });

                $("#table").jtable("load");

            });

        </script>

    </body>

</html>

**Options:**

|  |  |
| --- | --- |
| Option | Description |
| title | set title of table |
| addRecordButton | configure add button |
| showCloseButton | default: true, when user click on button, closeRequested event triggered, this is used to close child table |
| ajaxSetting | configure global ajax request |
| columnResizable | default: true, allow to resize column |
| columnSelectable | default: true, user can show and hide column by right click on header |
| sorting | default: false, give sorting button in header |
| defaultSorting | "ColumnnName ASC" or "ColumnName DESC" we can also set multiple column |
| multiSorting | default: false, set multiple sorting |
| paging | default: false, enable paging |
| pageSize | default: 10, set page size |
| gotoPageArea | set direct page navigation is enable or not |
| pageSizeChangeArea | default: true, set pagination combobox is enable or not |
| pageList | set behaviour of navigation panel, we have two behaviour for navigation panel   * minimal: Show only first, previous, next and last links. * normal: Shows page numbers in addition to 'minimal'. |
| pageSizes | set combobox option |
| selecting | default: false, it enable to select row |
| multiselect | default: false, it enable multiple selection of rows |
| selectingCheckboxes | default: false, it enable checkbox for select row |
| selectOnRowClick | default: false, it enables selection row on clicking |
| deleteConfiguration | default: true, it set whether dialog box open or not on delete |
| jqueryuiTheme | default: false, set whether jqueryui theam use or not |
| animationsEnabled | default: true, set whether animation enable or not |
| loadingAnimationDelay | set loading animation delay |
| dialogShowEffect | set dialog show effect |
| dialogHideEffect | set dialog hide effect, we have multiple types of effect like, blind', 'bounce', 'clip', 'drop', 'explode', 'fold', 'highlight', 'puff', 'pulsate', 'scale', 'shake', 'size', 'slide' |
| openChildAsAccordion | default: false, set child table is open and close in accordion |
| defaultDateFormat | default: yy-mm-dd, set default format of date |
| saveUserPreferences | default: true it save the user customization and apply on other page |
| showCloseButton | default: false, it set the close button in ui, it work with child table |
| tableId | set table id for saving and restoring user preferance |
| toolbar | configure toolbar |
| hoverAnimation | enable/disable small animation of mouse hover on toolbar |
| hoverAnumationDuration | set hover animation duration in toolbar |
| hoverAnimationEasing | set hover animation easing in toolbar |
| items | array of custom toolbar items |
| unAuthorizedRequestRedirectUrl | it redirect specific url if ajax call is unauthorized |

**Fields:**

|  |  |
| --- | --- |
| Field | Description |
| title | title of this column in form |
| key | set key of table |
| edit | this field don't show when we edit record |
| create | this field don't show when we edit record |
| delete | it could be delete |
| columnResizable | it allows us to resize column |
| width | set width of column |
| visibility | set visibility behavior, we have three different option for visibility   * fixed: this is always visible can't hide by user * visible: this is always visible can't hide by user * hidden: this is always visible can't hide by user |
| sorting | enable sorting |
| optionSorting | set the behavior of sorting, we have four different behavior for sorting   * value: sort according to the value in ascending, ex: 1, 2, 3,..., 10, 11, 12,.. * value-desc: sort according to the value in descending ex: 21 , 20, 19, 18,..., 2, 1 * text: sort according to the display text in ascending ex: 1, 10, 11, 2, 21, 3, 4, 41,.. * text-desc: sort according to the display text in ascending ex: ex: 41,4, 3, 21, 2, 11, 10, 1,.. |
| options | Create dropdown for field in edit or add form |
| type | set type of field in form, we have different type of field type like, password, textarea, date(if we use date then field must be in date type in table), radiobutton, cheackbox, hidden(hide in form also hide in table) |
| defaultValue | set default value in field of form |
| inputTitle | set title of field in form |
| inputClass | set classes in field in form |
| list | hide the column on table, but show in form |
| listClass | set classes in particular field list |
| dependsOn | This property use to get the date of the other field inside the current column |
| input | create customized input field in form |
| display | create customized cell in list |

**Actions:**

|  |  |
| --- | --- |
| Action | Description |
| listAction | Configuration for get all data from api, callback function which take two parameters and return deferred   * postData: it contains data which we want to add, update or delete, but this is not use in list action * jtParams: it contains three properties, jtStartIndex, jtPageSize and jtSorting, base on thease properties we can do pagging and sorting in list.   We must return response of api in specific format,  {  Result: “OK”  Records: <list of object of record>,  TotalRecordCount: <Count of all record of table which exist in database not count of page record>  } |
| createAction | Configuration for add data from api, callback function which take one parameter and return deferred   * postData: it contains data which we want to add   We must return response of api in specific format,  {  Result: “OK”  Records: <object of newly created record>  } |
| updateAction | Configuration for update data from api, callback function which take one parameter and return deferred   * postData: it contains data which we want to update   We must return response of api in specific format,  {  Result: “OK”  Records: <object of newly created record> (Optional)  } |
| deleteAction | Configuration for delete data from api, callback function which take one parameter and return deferred   * postData: it contains id of data which we want to delete   We must return response of api in specific format,  {  Result: “OK”  } |

**Methods:**

|  |  |
| --- | --- |
| Method | Description |
| load | load a data into table |
| getRowByKey | It is used to retrieve record by key |
| addRecord | It is used to add new record in table programmatically  options:   * record: the object of record which we want to add * clientOnly: default: false, if it is true then record only add in client side not affect on server * animationsEnable: default: true, if true then show animation while deleting row * url: specify url for createAction * success: it is callback function which is execute when ajax call success * error: it is callback function which is execute when ajax call give something error |
| updateRocord | it is used to delete row by key  options:   * record: the object of record which we want to update * clientOnly: default: false, if it is true then record only add in client side not affect on server * animationsEnable: default: true, if true then show animation while deleting row * url: specify url for updateAction * success: it is callback function which is execute when ajax call success * error: it is callback function which is execute when ajax call give something error |
| deleteRecord | It is used to delete row by key  options:   * record: the object of record which we want to update * clientOnly: default: false, if it is true then record only add in client side not affect on server * animationsEnable: default: true, if true then show animation while deleting row * url: specify url for updateAction * success: it is callback function which is execute when ajax call success * error: it is callback function which is execute when ajax call give something error |
| changeColumnVisibility | It is used to change column visibility, we have three different behavior of visibility like, hidden, visible, fixed |
| selectedRows | it is used to get all selected rows from table |
| deleteRows | it will delete rows from server and client both side |
| reload | it will reload table and execute callback function |
| destroy | it will destroy table |
| selectRows | it is used to select rows |
| showCreateForm | it will show create form |
| openChildRow | it is used to open child rows |
| closeChildRow | it is used to close child rows |
| openChildTable | it is used to create and open child table  options:   * row: data of the table * tableOptions: options of table * opened: callback function which is execute after table is opened |
| closeChildTable | it is used to create and close child table  options:   * row: data of the table * closed: callback function which is execute after table is closed |

**Events:**

|  |  |
| --- | --- |
| Events | Description |
| closeRequested | it triggered when we click on close button of table |
| formClosed | it triggered when we form is close, it give two parameters, event and data,   * data.form: it give form as a jQuery selector * data.formType: it give type of form like create or edit * data.row: it give us edited row if form is in edit mode |
| formCreated | it triggered when we form is create, it give two parameters, event and data,   * data.form: it give form as a jQuery selector * data.formType: it give type of form like create or edit * data.record: it give us edited record if form is in edit mode * data.row: it give us edited row if form is in edit mode |
| formSubmitting | it triggered when we form is submit, it give two parameters, event and data,   * data.form: it give form as a jQuery selector * data.formType: it give type of form like create or edit * data.row: it give us edited row if form is in edit mode |
| loadingRecords | it triggered when loading recored in table |
| recordsLoaded | it triggered when recored load successfully, it give two parameters, event and data,   * data.record: it give us all the record which is loaded * data.serverResponse: it give use server response |
| recordAdded | it triggered when record is add, it give two parameters, event and data,   * data.record: it give us object of added record * data.serverResponse: it give use server response |
| recordUpdated | it triggered when record is update, it give two parameters, event and data,   * data.record: it give us object of added record * data.row: it give us edited row * data.serverResponse: it give use server response |
| recordDeleted | it triggered when record is delete, it give two parameters, event and data,   * data.record: it give us object of added record * data.row: it give us deleted row * data.serverResponse: it give use server response |
| rowInserted | it triggered when row insert in table, it give two parameters, event and data,   * data.record: it give us record * data.row: it give us row * data.isNewRow: it gives true if it is new row |
| rowUpdated | it triggered when row update in table, it give two parameters, event and data,   * data.record: it give us updated record * data.row: it give us updated row |
| rowRemoved | it triggered when row insert in table, it give two parameters, event and data,   * data.row: it give us deleted row * data.reason: it give us reason |
| selectionChanged | it triggered when select any row or unselect any row |