### Editors – Overview

## Check Box (dxCheckBox)

### Properties

* value → Boolean (true, false, null for indeterminate).
* text → String (Label text for the checkbox).
* disabled → Boolean (Enables/disables the checkbox).
* readOnly → Boolean (Prevents user interaction).
* elementAttr → Object (Custom attributes for the root element).

### Methods

* .option("value", true/false) → Sets the checkbox value.
* .toggle() → Toggles between checked and unchecked states.
* .focus() → Sets focus on the checkbox.
* .dispose() → Destroys the widget.

### Events

* onValueChanged → Fires when the checkbox state changes.
* onFocusIn → Fires when the checkbox gains focus.
* onFocusOut → Fires when the checkbox loses focus.

#### **Example**

$("#checkBox").dxCheckBox({  
 text: "Accept Terms",  
 value: false,  
 onValueChanged: function (e) {  
 console.log("Checkbox state: " + e.value);  
 },  
});

## Date Box (dxDateBox)

### Properties

* value → Date/String (Selected date).
* type → String ("date", "datetime", "time", "month", "year").
* min / max → Date/String (Defines valid date range).
* displayFormat → String (Formats date display, e.g., "MM/dd/yyyy").
* calendarOptions → Object (Customizes the popup calendar).

### Methods

* .option("value", new Date()) → Sets a new date value.
* .open() / .close() → Opens/closes the date picker manually.
* .reset() → Resets the value to default.

### Events

* onValueChanged → Fires when the selected date changes.
* onOpened / onClosed → Fires when the date picker opens/closes.
* onKeyDown → Captures key events inside the widget.

#### **Example**

$("#dateBox").dxDateBox({  
 type: "date",  
 displayFormat: "MM/dd/yyyy",  
 min: "2023-01-01",  
 max: "2025-12-31",  
 onValueChanged: function (e) {  
 console.log("Selected Date: " + e.value);  
 },  
});

## Drop Down Box (dxDropDownBox)

### Properties

* value → Any (Selected item value).
* items → Array (Dropdown item list).
* placeholder → String (Placeholder text).
* displayExpr / valueExpr → String (Defines how values are displayed and stored).
* searchEnabled → Boolean (Enables search inside the dropdown).
* opened → Boolean (Defines whether the dropdown is open).

### Methods

* .open() / .close() → Opens or closes the dropdown manually.
* .option("value", newValue) → Changes the selected item.
* .reset() → Resets the dropdown selection.

### Events

* onValueChanged → Fires when the selected item changes.
* onOpened / onClosed → Fires when the dropdown opens/closes.
* onSelectionChanged → Fires when a different item is selected.

#### **Example**

$("#dropDownBox").dxDropDownBox({  
 items: ["Apple", "Banana", "Cherry"],  
 placeholder: "Select a fruit",  
 searchEnabled: true,  
 onValueChanged: function (e) {  
 console.log("Selected Item: " + e.value);  
 },  
});

## Number Box (dxNumberBox)

### Properties

* value → Number (Current numeric value).
* min / max → Number (Defines valid numeric range).
* step → Number (Increment/decrement step).
* showSpinButtons → Boolean (Displays increase/decrease buttons).
* format → String ("fixedPoint", "currency", "percent", etc.).

### Methods

* .option("value", 10) → Sets the numeric value.
* .reset() → Resets the value to default.

### Events

* onValueChanged → Fires when the value changes.

#### **Example**

$("#numberBox").dxNumberBox({  
 value: 5,  
 min: 1,  
 max: 100,  
 showSpinButtons: true,  
 step: 1,  
 onValueChanged: function (e) {  
 console.log("New value: " + e.value);  
 },  
});

## Select Box (dxSelectBox)

### Properties

* items → Array (Dropdown items).
* value → Any (Selected item value).
* placeholder → String (Text before selection).
* searchEnabled → Boolean (Allows searching in dropdown).
* displayExpr / valueExpr → String (Defines how values are displayed and stored).

### Methods

* .open() / .close() → Opens/closes the dropdown manually.
* .option("value", newValue) → Changes the selected item.
* .reset() → Resets the dropdown selection.

### Events

* onValueChanged → Fires when the selection changes.
* onSelectionChanged → Fires when an item is selected.

#### **Example**

$("#selectBox").dxSelectBox({  
 items: ["Red", "Green", "Blue"],  
 placeholder: "Select a color",  
 searchEnabled: true,  
 onValueChanged: function (e) {  
 console.log("Selected: " + e.value);  
 },  
});

## Search and Editing

### Properties

* searchEnabled → Boolean (Enables search in dropdowns and lists).
* searchExpr → String/Array (Defines searchable fields).

#### **Example (Search in SelectBox)**

$("#searchBox").dxSelectBox({  
 items: ["Item 1", "Item 2", "Item 3"],  
 searchEnabled: true,  
 searchExpr: "name",  
});

## Grouped Items

### Properties

* grouped → Boolean (Enables grouping of list items).
* groupTemplate → Function/String (Customizes group headers).

#### **Example**

$("#groupedList").dxList({  
 dataSource: [  
 { key: "Fruits", items: ["Apple", "Banana"] },  
 { key: "Vegetables", items: ["Carrot", "Tomato"] },  
 ],  
 grouped: true,  
});

## Text Area (dxTextArea)

### Properties

* value → String (Text content).
* maxLength → Number (Character limit).
* placeholder → String (Hint text).

### Methods

* .option("value", "New text") → Sets text content.
* .reset() → Clears the text.

### Events

* onValueChanged → Fires when text changes.

#### **Example**

$("#textArea").dxTextArea({  
 placeholder: "Enter your message",  
 maxLength: 500,  
});

## Text Box (dxTextBox)

### Properties

* value → String (Entered text).
* mode → String ("text", "password", "email", "tel").
* showClearButton → Boolean (Adds a clear button).

### Methods

* .option("value", "New text") → Sets input value.
* .reset() → Clears the text.

### Events

* onValueChanged → Fires when text changes.

#### **Example**

$("#textBox").dxTextBox({  
 mode: "text",  
 placeholder: "Enter your name",  
 showClearButton: true,  
});

## Button (dxButton)

### Properties

* text → String (Button label).
* type → String ("default", "success", "danger", "back").
* disabled → Boolean (Enables/disables the button).

### Methods

* .option("text", "New Label") → Updates the button label.

### Events

* onClick → Fires when the button is clicked.

#### **Example**

$("#button").dxButton({  
 text: "Submit",  
 type: "success",  
 onClick: function () {  
 alert("Button clicked!");  
 },  
});

## File Uploader (dxFileUploader)

### Properties

* multiple → Boolean (Allows multiple file selection).
* accept → String (Defines allowed file types, e.g., "image/\*").
* uploadUrl → String (Server endpoint for uploads).

### Methods

* .upload() → Manually starts file upload.
* .reset() → Clears selected files.

### Events

* onUploaded → Fires after a file is uploaded.
* onProgress → Fires during upload progress.

#### **Example**

$("#fileUploader").dxFileUploader({  
 multiple: true,  
 accept: "image/\*",  
 uploadUrl: "/upload",  
 onUploaded: function (e) {  
 console.log("File uploaded: " + e.file.name);  
 },  
});

## Validation (dxValidator)

### Properties

* validationRules → Array (Defines validation rules, e.g., required, email, numeric).

### Methods

* .validate() → Manually triggers validation.

### Events

* onValidated → Fires when validation is complete.

#### **Example**

$("#textBox")  
 .dxTextBox({  
 placeholder: "Enter email",  
 })  
 .dxValidator({  
 validationRules: [{ type: "email", message: "Invalid email" }],  
 });

## Radio Group (dxRadioGroup)

### Properties

* items → Array (Available options).
* value → Any (Selected option).
* layout → String ("horizontal", "vertical").

### Methods

* .option("value", newValue) → Changes selection.

### Events

* onValueChanged → Fires when selection changes.

#### **Example**

$("#radioGroup").dxRadioGroup({  
 items: ["Male", "Female", "Other"],  
 value: "Male",  
 layout: "horizontal",  
 onValueChanged: function (e) {  
 console.log("Selected: " + e.value);  
 },  
});