

2. Editors

2.1 Common

- Options :

Name	Description
accessKey	Specifies the shortcut key that sets focus on the UI component.
activeStateEnabled	Specifies whether the UI component changes its state as a result of user interaction.
disabled	Specifies whether the UI component responds to user interaction.
elementAttr	Specifies the global attributes to be attached to the UI component's container element.
focusStateEnabled	Specifies whether the UI component can be focused using keyboard navigation.
height	Specifies the UI component's height.
hint	Specifies text for a hint that appears when a user pauses on the UI component.
hoverStateEnabled	Specifies whether the UI component changes its state when a user pauses on it.
isValid	Specifies or indicates whether the editor's value is valid.
name	The value to be assigned to the name attribute of the underlying HTML element.
onContentReady	A function that is executed when the UI component's content is ready and each time the content is changed.
onDisposing	A function that is executed before the UI component is disposed of.
onInitialized	A function used in JavaScript frameworks to save the UI component instance.
onOptionChanged	A function that is executed after a UI component property is changed.
onValueChanged	A function that is executed after the UI component's value is changed.
readOnly	Specifies whether the editor is read-only.
rtlEnabled	Switches the UI component to a right-to-left representation.
tabIndex	Specifies the number of the element when the Tab key is used for navigating.
text	Specifies the text displayed by the check box.
validationError	Information on the broken validation rule. Contains the first item from the validationErrors array.
validationErrors	An array of the validation rules that failed.
validationMessageMode	Specifies how the message about the validation rules that are not satisfied by this editor's value is displayed.
validationStatus	Indicates or specifies the current validation status.
value	Specifies the UI component state.
visible	Specifies whether the UI component is visible.
width	Specifies the UI component's width.

- Methods :

Name	Description
beginUpdate()	Prevents the UI component from refreshing until the endUpdate() method is called.
defaultOptions(rule)	Specifies the device-dependent default configuration properties for this component.
dispose()	Disposes of all the resources allocated to the CheckBox instance.
element()	Gets the root UI component element.
endUpdate()	Refreshes the UI component after a call of the beginUpdate() method.
focus()	Sets focus on the UI component.
getInstance(element)	Gets the instance of a UI component found using its DOM node.
instance()	Gets the UI component's instance. Use it to access other methods of the UI component.
off(eventName)	Detaches all event handlers from a single event.
off(eventName, eventHandler)	Detaches a particular event handler from a single event.
on(eventName, eventHandler)	Subscribes to an event.
on(events)	Subscribes to events.
option()	Gets all UI component properties.
option(optionName)	Gets the value of a single property.
option(optionName, optionValue)	Updates the value of a single property.
option(options)	Updates the values of several properties.
registerKeyHandler(key, handler)	Registers a handler to be executed when a user presses a specific key.
repaint()	Repaints the UI component without reloading data. Call it to update the UI component's markup.
reset()	Resets the value property to the default value.
resetOption(optionName)	Resets a property to its default value.

- Events :

Name	Description
contentReady	Raised when the UI component's content is ready.
disposing	Raised before the UI component is disposed of.
initialized	Raised only once, after the UI component is initialized.
optionChanged	Raised after a UI component property is changed.
valueChanged	Raised after the UI component's value is changed.

2.2 CheckBox

- The CheckBox is a small box, which when selected by the end user, shows that a particular feature has been enabled or a specific property has been chosen.

2.3 DateBox

- The DateBox is a UI component that displays date and time in a specified format, and enables a user to pick or type in the required date/time value.
- Options :

Name	Description
acceptCustomValue	Specifies whether or not the UI component allows an end-user to enter a custom value.
adaptivityEnabled	Specifies whether or not adaptive UI component rendering is enabled on a small screen.
applyButtonText	The text displayed on the Apply button.
applyValueMode	Specifies the way an end-user applies the selected value.
buttons	Allows you to add custom buttons to the input text field.
calendarOptions	Configures the calendar's value picker. Applies only if the pickerType is "calendar".
cancelButtonText	The text displayed on the Cancel button.
dateOutOfRangeMessage	Specifies the message displayed if the specified date is later than the max value or earlier than the min value.
dateSerializationFormat	Specifies the date-time value serialization format. Use it only if you do not specify the value at design time.
deferRendering	Specifies whether to render the drop-down field's content when it is displayed. If false, the content is rendered immediately.
disabledDates	Specifies dates that users cannot select. Applies only if pickerType is "calendar".
displayFormat	Specifies the date display format. Ignored if the pickerType property is "native"
dropDownButtonComponent	An alias for the dropDownButtonTemplate property specified in React. Accepts a custom component. Refer to Using a Custom Component for more information.
dropDownButtonRender	An alias for the dropDownButtonTemplate property specified in React. Accepts a rendering function. Refer to Using a Rendering Function for more information.
dropDownButtonTemplate	Specifies a custom template for the drop-down button.
dropDownOptions	Configures the drop-down field which holds the content.
interval	Specifies the interval between neighboring values in the popup list in minutes.
invalidDateMessage	Specifies the message displayed if the typed value is not a valid date or time.
max	The last date that can be selected within the UI component.
maxLength	Specifies the maximum number of characters you can enter into the textbox.

min	The minimum date that can be selected within the UI component.
onChange	A function that is executed when the UI component loses focus after the text field's content was changed using the keyboard.
onClosed	A function that is executed once the drop-down editor is closed.
onCopy	A function that is executed when the UI component's input has been copied.
onCut	A function that is executed when the UI component's input has been cut.
onEnterKey	A function that is executed when the Enter key has been pressed while the UI component is focused.
onFocusIn	A function that is executed when the UI component gets focus.
onFocusOut	A function that is executed when the UI component loses focus.
onInput	A function that is executed each time the UI component's input is changed while the UI component is focused.
onKeyDown	A function that is executed when a user is pressing a key on the keyboard.
onKeyUp	A function that is executed when a user releases a key on the keyboard.
onOpened	A function that is executed once the drop-down editor is opened.
onPaste	A function that is executed when the UI component's input has been pasted.
opened	Specifies whether or not the drop-down editor is displayed.
openOnFieldClick	Specifies whether a user can open the drop-down list by clicking a text field.
pickerType	Specifies the type of the date/time picker.
placeholder	Specifies a placeholder for the input field.
showAnalogClock	Specifies whether to show the analog clock in the value picker. Applies only if type is "datetime" and pickerType is "calendar".
showClearButton	Specifies whether to display the Clear button in the UI component.
showDropDownButton	Specifies whether the drop-down button is visible.
spellcheck	Specifies whether or not the UI component checks the inner text for spelling mistakes.
stylingMode	Specifies how the UI component's text field is styled.
type	A format used to display date/time information.
useMaskBehavior	Specifies whether to control user input using a mask created based on the displayFormat.
valueChangeEvent	Specifies the DOM events after which the UI component's value should be updated.

- Methods :

Name	Description
blur()	Removes focus from the input element.

close()	Closes the drop-down editor.
content()	Gets the popup window's content.
field()	Gets the UI component's <input> element.
getButton(name)	Gets an instance of a custom action button.
open()	Opens the drop-down editor.

- Events :

Name	Description
change	Raised when the UI component loses focus after the text field's content was changed using the keyboard.
closed	Raised once the drop-down editor is closed.
copy	Raised when the UI component's input has been copied.
cut	Raised when the UI component's input has been cut.
enterKey	Raised when the Enter key has been pressed while the UI component is focused.
focusIn	Raised when the UI component gets focus.
focusOut	Raised when the UI component loses focus.
input	Raised each time the UI component's input is changed while the UI component is focused.
keyDown	Raised when a user is pressing a key on the keyboard.
keyUp	Raised when a user releases a key on the keyboard.
opened	Raised once the drop-down editor is opened.
paste	Raised when the UI component's input has been pasted.

2.4 DropDownBox

- The DropDownBox UI component consists of a text field, which displays the current value, and a drop-down field, which can contain any UI element.
- Options :

Name	Description
acceptCustomValue	Specifies whether the UI component allows a user to enter a custom value.
activeStateEnabled	Specifies whether the UI component changes its state as a result of user interaction.
buttons	Allows you to add custom buttons to the input text field.
contentComponent	An alias for the contentTemplate property specified in React. Accepts a custom component. Refer to Using a Custom Component for more information.
contentRender	An alias for the contentTemplate property specified in React. Accepts a rendering function. Refer to Using a Rendering Function for more information.
contentTemplate	Specifies a custom template for the drop-down content.
dataSource	Binds the UI component to data.
deferRendering	Specifies whether to render the drop-down field's content when it is displayed. If false, the content is rendered immediately.

displayExpr	Specifies the data field whose values should be displayed.
displayValueFormatter	Customizes text before it is displayed in the input field.
dropDownButtonComponent	An alias for the dropDownButtonTemplate property specified in React. Accepts a custom component. Refer to Using a Custom Component for more information.
dropDownButtonRender	An alias for the dropDownButtonTemplate property specified in React. Accepts a rendering function. Refer to Using a Rendering Function for more information.
dropDownButtonTemplate	Specifies a custom template for the drop-down button.
dropDownOptions	Configures the drop-down field which holds the content.
fieldComponent	An alias for the fieldTemplate property specified in React. Accepts a custom component. Refer to Using a Custom Component for more information.
fieldRender	An alias for the fieldTemplate property specified in React. Accepts a rendering function. Refer to Using a Rendering Function for more information.
fieldTemplate	Specifies a custom template for the text field. Must contain the TextBox UI component.
items	An array of items used to synchronize the DropDownBox with an embedded UI component.
maxLength	Specifies the maximum number of characters you can enter into the textbox.
onChange	A function that is executed when the UI component loses focus after the text field's content was changed using the keyboard.
onClosed	A function that is executed once the drop-down editor is closed.
onCopy	A function that is executed when the UI component's input has been copied.
onCut	A function that is executed when the UI component's input has been cut.
onEnterKey	A function that is executed when the Enter key has been pressed while the UI component is focused.
onFocusIn	A function that is executed when the UI component gets focus.
onFocusOut	A function that is executed when the UI component loses focus.
onInput	A function that is executed each time the UI component's input is changed while the UI component is focused.
onKeyDown	A function that is executed when a user is pressing a key on the keyboard.
onKeyUp	A function that is executed when a user releases a key on the keyboard.
onOpened	A function that is executed once the drop-down editor is opened.

onPaste	A function that is executed when the UI component's input has been pasted.
opened	Specifies whether or not the drop-down editor is displayed.
openOnFieldClick	Specifies whether a user can open the drop-down list by clicking a text field.
placeholder	The text displayed by the UI component when the UI component value is empty.
showClearButton	Specifies whether to display the Clear button in the UI component.
showDropDownButton	Specifies whether the drop-down button is visible.
stylingMode	Specifies how the UI component's text field is styled.
validationErrors	An array of the validation rules that failed.
valueChangeEvent	Specifies the DOM events after which the UI component's value should be updated.
valueExpr	Specifies which data field provides unique values to the UI component's value.

- **Methods :**

Name	Description
blur()	Removes focus from the input element.
close()	Closes the drop-down editor.
content()	Gets the popup window's content.
field()	Gets the UI component's <input> element.
getButton(name)	Gets an instance of a custom action button.
getDataSource()	Gets the DataSource instance.
open()	Opens the drop-down editor.

- **Events :**

Name	Description
change	Raised when the UI component loses focus after the text field's content was changed using the keyboard.
closed	Raised once the drop-down editor is closed.
copy	Raised when the UI component's input has been copied.
cut	Raised when the UI component's input has been cut.
enterKey	Raised when the Enter key has been pressed while the UI component is focused.
focusIn	Raised when the UI component gets focus.
focusOut	Raised when the UI component loses focus.
input	Raised each time the UI component's input is changed while the UI component is focused.
keyDown	Raised when a user is pressing a key on the keyboard.
keyUp	Raised when a user releases a key on the keyboard.
opened	Raised once the drop-down editor is opened.
paste	Raised when the UI component's input has been pasted.

2.5 NumberBox

- The NumberBox is a UI component that displays a numeric value and allows a user to modify it by typing in a value, and incrementing or decrementing it using the keyboard or mouse.
- Options :

Name	Description
buttons	Allows you to add custom buttons to the input text field.
format	Specifies the value's display format and controls user input accordingly.
invalidValueMessage	Specifies the text of the message displayed if the specified value is not a number.
max	The maximum value accepted by the number box.
min	The minimum value accepted by the number box.
mode	Specifies the value to be passed to the type attribute of the underlying <input> element.
onChange	A function that is executed when the UI component loses focus after the text field's content was changed using the keyboard.
onCopy	A function that is executed when the UI component's input has been copied.
onCut	A function that is executed when the UI component's input has been cut.
onEnterKey	A function that is executed when the Enter key has been pressed while the UI component is focused.
onFocusIn	A function that is executed when the UI component gets focus.
onFocusOut	A function that is executed when the UI component loses focus.
onInput	A function that is executed each time the UI component's input is changed while the UI component is focused.
onKeyDown	A function that is executed when a user is pressing a key on the keyboard.
onKeyUp	A function that is executed when a user releases a key on the keyboard.
onPaste	A function that is executed when the UI component's input has been pasted.
placeholder	The text displayed by the UI component when the UI component value is empty.
showClearButton	Specifies whether to display the Clear button in the UI component.
showSpinButtons	Specifies whether to show the buttons that change the value by a step.
step	Specifies how much the UI component's value changes when using the spin buttons, Up/Down arrow keys, or mouse wheel.
stylingMode	Specifies how the UI component's text field is styled.
useLargeSpinButtons	Specifies whether to use touch friendly spin buttons. Applies only if showSpinButtons is true.
valueChangeEvent	Specifies the DOM events after which the UI component's value should be updated.

- Methods :

Name	Description
blur()	Removes focus from the input element.
getButton(name)	Gets an instance of a custom action button.

- Events :

Name	Description
change	Raised when the UI component loses focus after the text field's content was changed using the keyboard.
copy	Raised when the UI component's input has been copied.
cut	Raised when the UI component's input has been cut.
enterKey	Raised when the Enter key has been pressed while the UI component is focused.
focusIn	Raised when the UI component gets focus.
focusOut	Raised when the UI component loses focus.
input	Raised each time the UI component's input is changed while the UI component is focused.
keyDown	Raised when a user is pressing a key on the keyboard.
keyUp	Raised when a user releases a key on the keyboard.
paste	Raised when the UI component's input has been pasted.

2.6 SelectBox

- The SelectBox UI component is an editor that allows an end user to select an item from a drop-down list.
- Options :

Name	Description
acceptCustomValue	Specifies whether the UI component allows a user to enter a custom value. Requires the onCustomItemCreating handler implementation.
activeStateEnabled	Specifies whether the UI component changes its state as a result of user interaction.
buttons	Allows you to add custom buttons to the input text field.
dataSource	Binds the UI component to data.
deferRendering	Specifies whether to render the drop-down field's content when it is displayed. If false, the content is rendered immediately.
displayExpr	Specifies the data field whose values should be displayed.
displayValue	Returns the value currently displayed by the UI component.

dropDownButtonComponent	An alias for the dropDownButtonTemplate property specified in React. Accepts a custom component. Refer to Using a Custom Component for more information.
dropDownButtonRender	An alias for the dropDownButtonTemplate property specified in React. Accepts a rendering function. Refer to Using a Rendering Function for more information.
dropDownButtonTemplate	Specifies a custom template for the drop-down button.
dropDownOptions	Configures the drop-down field which holds the content.
fieldComponent	An alias for the fieldTemplate property specified in React. Accepts a custom component. Refer to Using a Custom Component for more information.
fieldRender	An alias for the fieldTemplate property specified in React. Accepts a rendering function. Refer to Using a Rendering Function for more information.
fieldTemplate	Specifies a custom template for the text field. Must contain the TextBox UI component.
groupComponent	An alias for the groupTemplate property specified in React. Accepts a custom component. Refer to Using a Custom Component for more information.
grouped	Specifies whether data items should be grouped.
groupRender	An alias for the groupTemplate property specified in React. Accepts a rendering function. Refer to Using a Rendering Function for more information.
groupTemplate	Specifies a custom template for group captions.
itemComponent	An alias for the itemTemplate property specified in React. Accepts a custom component. Refer to Using a Custom Component for more information.
itemRender	An alias for the itemTemplate property specified in React. Accepts a rendering function. Refer to Using a Rendering Function for more information.
items	An array of items displayed by the UI component.
itemTemplate	Specifies a custom template for items.
maxLength	Specifies the maximum number of characters you can enter into the textbox.
minSearchLength	The minimum number of characters that must be entered into the text box to begin a search. Applies only if searchEnabled is true.
noDataText	Specifies the text or HTML markup displayed by the UI component if the item collection is empty.
onChange	A function that is executed when the UI component loses focus after the text field's content was changed using the keyboard.
onClosed	A function that is executed once the drop-down editor is closed.
onCopy	A function that is executed when the UI component's input has been copied.
onCustomItemCreating	A function that is executed when a user adds a custom item. Requires acceptCustomValue to be set to true.

onCut	A function that is executed when the UI component's input has been cut.
onEnterKey	A function that is executed when the Enter key has been pressed while the UI component is focused.
onFocusIn	A function that is executed when the UI component gets focus.
onFocusOut	A function that is executed when the UI component loses focus.
onInput	A function that is executed each time the UI component's input is changed while the UI component is focused.
onItemClick	A function that is executed when a list item is clicked or tapped.
onKeyDown	A function that is executed when a user is pressing a key on the keyboard.
onKeyUp	A function that is executed when a user releases a key on the keyboard.
onOpened	A function that is executed once the drop-down editor is opened.
onPaste	A function that is executed when the UI component's input has been pasted.
onSelectionChanged	A function that is executed when a list item is selected or selection is canceled.
opened	Specifies whether or not the drop-down editor is displayed.
openOnFieldClick	Specifies whether a user can open the drop-down list by clicking a text field.
placeholder	The text that is provided as a hint in the select box editor.
searchEnabled	Specifies whether to allow searching.
searchExpr	Specifies the name of a data source item field or an expression whose value is compared to the search criterion.
searchMode	Specifies a comparison operation used to search UI component items.
searchTimeout	Specifies the time delay, in milliseconds, after the last character has been typed in, before a search is executed.
selectedItem	Gets the currently selected item.
showClearButton	Specifies whether to display the Clear button in the UI component.
showDataBeforeSearch	Specifies whether or not the UI component displays unfiltered values until a user types a number of characters exceeding the minSearchLength property value.
showDropDownButton	Specifies whether the drop-down button is visible.
showSelectionControls	Specifies whether or not to display selection controls.
spellcheck	Specifies whether or not the UI component checks the inner text for spelling mistakes.
stylingMode	Specifies how the UI component's text field is styled.
useItemTextAsTitle	Specifies whether the SelectBox uses item's text a title attribute.

valueChangeEvent	Specifies the DOM events after which the UI component's value should be updated. Applies only if acceptCustomValue is set to true.
valueExpr	Specifies which data field provides unique values to the UI component's value.
wrapItemText	Specifies whether text that exceeds the drop-down list width should be wrapped.

- Methods :

Name	Description
blur()	Removes focus from the input element.
close()	Closes the drop-down editor.
content()	Gets the popup window's content.
field()	Gets the UI component's <input> element.
getButton(name)	Gets an instance of a custom action button.
getDataSource()	Gets the DataSource instance.
open()	Opens the drop-down editor.

- Events :

Name	Description
change	Raised when the UI component loses focus after the text field's content was changed using the keyboard.
closed	Raised once the drop-down editor is closed.
copy	Raised when the UI component's input has been copied.
customItemCreating	Raised when a user adds a custom item.
cut	Raised when the UI component's input has been cut.
enterKey	Raised when the Enter key has been pressed while the UI component is focused.
focusIn	Raised when the UI component gets focus.
focusOut	Raised when the UI component loses focus.
input	Raised each time the UI component's input is changed while the UI component is focused.
itemClick	Raised when a list item is clicked or tapped.
keyDown	Raised when a user is pressing a key on the keyboard.
keyUp	Raised when a user releases a key on the keyboard.
opened	Raised once the drop-down editor is opened.
paste	Raised when the UI component's input has been pasted.
selectionChanged	Raised when a list item is selected or selection is canceled.

2.7 DropDownBox VS SelectBox

- **DropDownBox** popup can contain different **views** like listView, treeView, gridView etc whereas **SelectBox** popup can only contain listView.
- **DropDownBox** can have **multiple values** at same time where as **SelectBox** can contain **single value at a time**.
- **DropDownBox** used in **complex scenarios**, on the other hand **SelectBox** is used in **simple scenarios**.

2.8 TextArea

- The TextArea is UI component that enables user to enter and edit a multi-line text.
- Options :

Name	Description
activeStateEnabled	Specifies whether the UI component changes its state as a result of user interaction.
autoResizeEnabled	A Boolean value specifying whether or not the auto resizing mode is enabled.
maxHeight	Specifies the maximum height of the UI component.
maxLength	Specifies the maximum number of characters you can enter into the textbox.
minHeight	Specifies the minimum height of the UI component.
onChange	A function that is executed when the UI component loses focus after the text field's content was changed using the keyboard.
onCopy	A function that is executed when the UI component's input has been copied.
onCut	A function that is executed when the UI component's input has been cut.
onEnterKey	A function that is executed when the Enter key has been pressed while the UI component is focused.
onFocusIn	A function that is executed when the UI component gets focus.
onFocusOut	A function that is executed when the UI component loses focus.
onInput	A function that is executed each time the UI component's input is changed while the UI component is focused.
onKeyDown	A function that is executed when a user is pressing a key on the keyboard.
onKeyUp	A function that is executed when a user releases a key on the keyboard.
onPaste	A function that is executed when the UI component's input has been pasted.
placeholder	The text displayed by the UI component when the UI component value is empty.
spellcheck	Specifies whether or not the UI component checks the inner text for spelling mistakes.
stylingMode	Specifies how the UI component's text field is styled.
valueChangeEvent	Specifies the DOM events after which the UI component's value should be updated.

- Methods :

Name	Description
blur()	Removes focus from the input element.

- Events :

Name	Description
change	Raised when the UI component loses focus after the text field's content was changed using the keyboard.
copy	Raised when the UI component's input has been copied.
cut	Raised when the UI component's input has been cut.
enterKey	Raised when the Enter key has been pressed while the UI component is focused.

focusIn	Raised when the UI component gets focus.
focusOut	Raised when the UI component loses focus.
input	Raised each time the UI component's input is changed while the UI component is focused.
keyDown	Raised when a user is pressing a key on the keyboard.
keyUp	Raised when a user releases a key on the keyboard.
paste	Raised when the UI component's input has been pasted.

2.9 TextBox

- The TextBox is a UI component that enables a user to enter and edit a single line of text.
- Options :

Name	Description
buttons	Allows you to add custom buttons to the input text field.
mask	The editor mask that specifies the custom format of the entered string.
maskChar	Specifies a mask placeholder. A single character is recommended.
maskInvalidMessage	A message displayed when the entered text does not match the specified pattern.
maskRules	Specifies custom mask rules.
maxLength	Specifies the maximum number of characters you can enter into the textbox.
mode	The "mode" attribute value of the actual HTML input element representing the text box.
onChange	A function that is executed when the UI component loses focus after the text field's content was changed using the keyboard.
onCopy	A function that is executed when the UI component's input has been copied.
onCut	A function that is executed when the UI component's input has been cut.
onEnterKey	A function that is executed when the Enter key has been pressed while the UI component is focused.
onFocusIn	A function that is executed when the UI component gets focus.
onFocusOut	A function that is executed when the UI component loses focus.
onInput	A function that is executed each time the UI component's input is changed while the UI component is focused.
onKeyDown	A function that is executed when a user is pressing a key on the keyboard.
onKeyUp	A function that is executed when a user releases a key on the keyboard.
onPaste	A function that is executed when the UI component's input has been pasted.
placeholder	The text displayed by the UI component when the UI component value is empty.
showClearButton	Specifies whether to display the Clear button in the UI component.

showMaskMode	Specifies when the UI component shows the mask. Applies only if useMaskedValue is true.
spellcheck	Specifies whether or not the UI component checks the inner text for spelling mistakes.
stylingMode	Specifies how the UI component's text field is styled.
useMaskedValue	Specifies whether the value should contain mask characters or not.
valueChangeEvent	Specifies the DOM events after which the UI component's value should be updated.

- Methods :

Name	Description
blur()	Removes focus from the input element.
getButton(name)	Gets an instance of a custom action button.

- Events :

Name	Description
change	Raised when the UI component loses focus after the text field's content was changed using the keyboard.
copy	Raised when the UI component's input has been copied.
cut	Raised when the UI component's input has been cut.
enterKey	Raised when the Enter key has been pressed while the UI component is focused.
focusIn	Raised when the UI component gets focus.
focusOut	Raised when the UI component loses focus.
input	Raised each time the UI component's input is changed while the UI component is focused.
keyDown	Raised when a user is pressing a key on the keyboard.
keyUp	Raised when a user releases a key on the keyboard.
paste	Raised when the UI component's input has been pasted.

2.10 TextArea VS TextBox

- TextArea is used to take multi-lines input where as TextBox can have only singleLine value.

2.11 Button

- The JavaScript Button is a simple button that performs specified commands when a user clicks it.
- Options :

Name	Description
icon	Specifies the icon to be displayed on the button.
onClick	A function that is executed when the Button is clicked or tapped.
rtlEnabled	Switches the UI component to a right-to-left representation.
stylingMode	Specifies how the button is styled.
template	Specifies a custom template for the Button UI component.
useSubmitBehavior	Specifies whether the button submits an HTML form.

validationGroup	Specifies the name of the validation group to be accessed in the click event handler.
-----------------	---

- Events :

Name	Description
click	Raised when the Button is clicked or tapped.

2.12 FileUploader

- The FileUploader UI component enables an end user to upload files to the server. An end user can select files in the file explorer or drag and drop files to the FileUploader area on the page.

- Options :

Name	Description
abortUpload	A function that cancels the file upload.
accept	Specifies a file type or several types accepted by the UI component.
allowCanceling	Specifies if an end user can remove a file from the selection and interrupt uploading.
allowedFileExtensions	Restricts file extensions that can be uploaded to the server.
chunkSize	Specifies the chunk size in bytes. Applies only if uploadMode is "instantly" or "useButtons". Requires a server that can process file chunks.
dialogTrigger	Specifies the HTML element which invokes the file upload dialog.
dropZone	Specifies the HTML element in which users can drag and drop files for upload.
invalidFileExtensionMessage	The text displayed when the extension of the file being uploaded is not an allowed file extension.
invalidMaxFileSizeMessage	The text displayed when the size of the file being uploaded is greater than the maxFileSize.
invalidMinFileSizeMessage	The text displayed when the size of the file being uploaded is less than the minFileSize.
labelText	Specifies the text displayed on the area to which an end-user can drop a file.
maxFileSize	Specifies the maximum file size (in bytes) allowed for uploading. Applies only if uploadMode is "instantly" or "useButtons".
minFileSize	Specifies the minimum file size (in bytes) allowed for uploading. Applies only if uploadMode is "instantly" or "useButtons".
multiple	Specifies whether the UI component enables an end-user to select a single file or multiple files.
onBeforeSend	A function that allows you to customize the request before it is sent to the server.
onDropZoneEnter	A function that is executed when the mouse enters a drop zone while dragging a file.
onDropZoneLeave	A function that is executed when the mouse leaves a drop zone as it drags a file.

onFilesUploaded	A function that is executed when the file upload process is complete.
onProgress	A function that is executed when a file segment is uploaded.
onUploadAborted	A function that is executed when the file upload is aborted.
onUploaded	A function that is executed when a file is successfully uploaded.
onUploadError	A function that is executed when an error occurs during the file upload.
onUploadStarted	A function that is executed when the file upload is started.
progress	Gets the current progress in percentages.
readyToUploadMessage	The message displayed by the UI component when it is ready to upload the specified files.
selectButtonText	The text displayed on the button that opens the file browser.
showFileList	Specifies whether or not the UI component displays the list of selected files.
uploadAbortedMessage	The message displayed by the UI component when the file upload is cancelled.
uploadButtonText	The text displayed on the button that starts uploading.
uploadChunk	A function that uploads a file in chunks.
uploadCustomData	Specifies custom data for the upload request.
uploadedMessage	The message displayed by the UI component when uploading is finished.
uploadFailedMessage	The message displayed by the UI component on uploading failure.
uploadFile	A function that uploads a file.
uploadHeaders	Specifies headers for the upload request.
uploadMethod	Specifies the method for the upload request.
uploadMode	Specifies how the UI component uploads files.
uploadUrl	Specifies a target Url for the upload request.

- Methods :

Name	Description
abortUpload()	Cancels the file upload.
abortUpload(file)	Cancels the file upload.
abortUpload(fileIndex)	Cancels the file upload.
removeFile(file)	Removes a file.
removeFile(fileIndex)	Removes a file with the specified index.
upload()	Uploads all the selected files.
upload(file)	Uploads the specified file.
upload(fileIndex)	Uploads a file with the specified index.

- Events :

Name	Description
beforeSend	Raised before the request is sent to the server and allows you to customize this request.
dropZoneEnter	Raised when the mouse enters a drop zone as it drags a file.
dropZoneLeave	Raised when the mouse leaves a drop zone as it drags a file.
filesUploaded	Raised when the file upload process is complete.

progress	Raised when a file segment is uploaded.
uploadAborted	Raised when the file upload is aborted.
uploaded	Raised when a file is successfully uploaded.
uploadError	Raised when an error occurs during the file upload.
uploadStarted	Raised when the file upload is started.

2.13 Validation

- A UI component that is used to validate the associated DevExtreme editors against the defined validation rules.
- Options :

Name	Description
adapter	An object that specifies what and when to validate, and how to apply the validation result.
onValidated	A function that is executed after a value is validated.
validationGroup	Specifies the validation group the editor will be related to.
validationRules	An array of validation rules to be checked for the editor with which the dxValidator object is associated.

- Method :

Name	Description
validate()	Validates the value of the editor that is controlled by the current Validator object against the list of the specified validation rules.

- Events :

Name	Description
validated	Raised after a value is validated.

- Validation Rules :

Name	Description
AsyncRule	A custom validation rule that is checked asynchronously. Use async rules for server-side validation.
CompareRule	A validation rule that demands that a validated editor has a value that is equal to a specified expression.
CustomRule	A rule with custom validation logic.
EmailRule	A validation rule that demands that the validated field match the Email pattern.
NumericRule	A validation rule that demands that the validated field has a numeric value.
PatternRule	A validation rule that demands that the validated field match a specified pattern.
RangeRule	A validation rule that demands the target value be within the specified value range (including the range's end points).
RequiredRule	A validation rule that demands that a validated field has a value.
StringLengthRule	A validation rule that demands the target value length be within the specified value range (including the range's end points).

- Validation Result :

Name	Description
brokenRule	A rule that failed to pass the check. Contains the first item from the brokenRules array
brokenRules	An array of the validationRules that failed to pass the check
complete	A promise that is fulfilled when all async rules are validated
isValid	Indicates whether all the checked rules are satisfied
pendingRule	An array of async rules whose promises are not fulfilled or rejected. Contains items only when the status is "pending"
status	Indicates the validation status
validationRules	Validation rules specified for the Validator
value	The value being validated

2.14 RadioGroup

- The RadioGroup is a UI component that contains a set of radio buttons and allows an end user to make a single selection from the set.
- Options :

Name	Description
dataSource	Binds the UI component to data.
displayExpr	Specifies the data field whose values should be displayed.
items	An array of items displayed by the UI component.
itemTemplate	Specifies a custom template for items.
layout	Specifies the radio group layout.
valueExpr	Specifies which data field provides unique values to the UI component's value.

- Methods :

Name	Description
getDataSource()	Gets the DataSource instance.