# DevExtreme Documentation

## 1.1 Introduction to DevExtreme

DevExtreme is a paid JavaScript library that provides pre-made web components for creating responsive web applications.

It includes over 70 UI components, integrated development templates, and tools for Angular, React, Vue, and jQuery.

DevExtreme plugin which provides features: - Data grid - Data editors - Navigation - Data Layer - Client Side Data Validation - Themes & styles - Localization - Modularity - Customization using templates - 70+ responsive UI components

Can add dx by: - CDN files - Local files (using dx zip file)

## 1.2 Installation – NuGet Package

In Visual Studio: - Manage NuGet Package Manager - Install DevExtreme package from console

Install-Package DevExtreme.Web -Version 21.1.3

## 1.3 Widget Basics – jQuery

### Create and Configure a Widget

<div id="gridContainer"></div>  
<script>  
 $(function() {  
 $("#gridContainer").dxDataGrid({  
 dataSource: data,  
 columns: ["name", "age", "city"]  
 });  
 });  
</script>

### Get a Widget Instance

var dataGridInstance = $("#gridContainer").dxDataGrid("instance");

### Get and Set Options

var pageSize = dataGridInstance.option("paging.pageSize");  
dataGridInstance.option("paging.pageSize", 20);

### Call Methods

dataGridInstance.refresh();

### Handle Events

dataGridInstance.on("cellClick", function(e) {  
 alert("Cell clicked: " + e.column.dataField);  
});

### Destroy a Widget

dataGridInstance.dispose();

# Editors DevExtreme CheckBox

## Options

1. **value**: Boolean or null. The value of the CheckBox.
2. **text**: String. The text displayed by the CheckBox.
3. **validationErrors**: Array. An array of validation error messages.
4. **validationMessageMode**: String. Specifies when the validation message is shown. Possible values: “always”, “auto”, “never”.
5. **elementAttr**: Object. Specifies the global attributes for the CheckBox element.
6. **disabled**: Boolean. Specifies whether the CheckBox is disabled.
7. **height**: Number or String. Specifies the CheckBox height.
8. **hint**: String. The hint displayed when the CheckBox is hovered over.
9. **accessKey**: String. The key that activates the CheckBox.
10. **activeStateEnabled**: Boolean. Specifies whether the CheckBox changes its state when a user presses it.
11. **enableThreeStateBehavior**: Boolean. Specifies whether the CheckBox supports three states (true, false, and indeterminate).
12. **focusStateEnabled**: Boolean. Specifies whether the CheckBox can be focused.
13. **hoverStateEnabled**: Boolean. Specifies whether the CheckBox changes its state when a user hovers the mouse pointer over it.
14. **isValid**: Boolean. Specifies whether the CheckBox is valid.
15. **tabIndex**: Number. The tab index of the CheckBox.
16. **rtlEnabled**: Boolean. Specifies whether the CheckBox supports right-to-left languages.

## Events

1. **onValueChanged**: Function. A function that is executed after the value is changed.
2. **onOptionChanged**: Function. A function that is executed after an option is changed.
3. **onInitialized**: Function. A function that is executed after a CheckBox is initialized.
4. **onContentReady**: Function. A function that is executed after the widget’s content has been fully rendered.
5. **onDisposing**: Function. A function that is executed before the CheckBox is disposed of.

## Methods

1. **element()**: Gets the root HTML element of the widget.
2. **instance()**: Gets the instance of the widget.
3. **beginUpdate()**: Prevents the widget from refreshing until the endUpdate() method is called.
4. **endUpdate()**: Refreshes the widget after a call to the beginUpdate() method.
5. **option(optionName, optionValue)**: Gets or sets an option’s value.
6. **repaint()**: Repaints the widget.
7. **focus()**: Sets focus on the widget.
8. **registerKeyHandler()**: Registers a handler to be executed when a key is pressed.
9. **reset()**: Resets the value option to the default value.
10. **dispose()**: Disposes of the CheckBox widget.
11. **off(eventName)**: Detaches all event handlers from the specified event.
12. **resetOption(optionName)**: Resets an option to its default value.

# DevExtreme DateBox

## Options

1. **acceptCustomValue**: Boolean. Prevents user from entering a custom value.
2. **accessKey**: String. Specifies the shortcut key that sets focus on the DateBox.
3. **focusStateEnabled**: Boolean. Specifies whether the DateBox can be focused.
4. **activeStateEnabled**: Boolean. Specifies whether the DateBox changes its state when a user presses it.
5. **applyButtonText**: String. The text displayed on the Apply button.
6. **applyValueMode**: String. Specifies when the selected value is applied. Possible values: “useButtons”, “instantly”.
7. **max**: Date. The maximum date allowed.
8. **min**: Date. The minimum date allowed.
9. **dateOutOfRangeMessage**: String. The message displayed when the date is out of range.
10. **cancelButtonText**: String. The text displayed on the Cancel button.
11. **displayFormat**: String. Specifies the display format of the date.
12. **deferRendering**: Boolean. Specifies whether to render the drop-down field’s content when it is displayed.
13. **height**: Number or String. Specifies the DateBox height.
14. **width**: Number or String. Specifies the DateBox width.
15. **hint**: String. The hint displayed when the DateBox is hovered over.
16. **openOnFieldClick**: Boolean. Specifies whether to open the drop-down when the field is clicked.
17. **inputAttr**: Object. Specifies the global attributes for the DateBox element.
18. **placeholder**: String. The placeholder text displayed in the DateBox.
19. **rtlEnabled**: Boolean. Specifies whether the DateBox supports right-to-left languages.
20. **showClearButton**: Boolean. Specifies whether to show the Clear button.
21. **stylingMode**: String. Specifies the styling mode of the DateBox. Possible values: “outlined”, “underlined”, “filled”.
22. **useMaskBehavior**: Boolean. Specifies whether to use masked behavior.

## Events

1. **onValueChanged**: Function. A function that is executed after the value is changed.
2. **onContentReady**: Function. A function that is executed after the widget’s content has been fully rendered.
3. **onDisposing**: Function. A function that is executed before the DateBox is disposed of.
4. **onInitialized**: Function. A function that is executed after a DateBox is initialized.
5. **onOptionChanged**: Function. A function that is executed after an option is changed.

## Methods

1. **element()**: Gets the root HTML element of the widget.
2. **instance()**: Gets the instance of the widget.
3. **beginUpdate()**: Prevents the widget from refreshing until the endUpdate() method is called.
4. **endUpdate()**: Refreshes the widget after a call to the beginUpdate() method.
5. **option(optionName, optionValue)**: Gets or sets an option’s value.
6. **repaint()**: Repaints the widget.
7. **focus()**: Sets focus on the widget.
8. **close()**: Closes the drop-down editor.
9. **blur()**: Removes focus from the widget.
10. **reset()**: Resets the value option to the default value.
11. **dispose()**: Disposes of the DateBox widget.
12. **off(eventName)**: Detaches all event handlers from the specified event.
13. **resetOption(optionName)**: Resets an option to its default value.

# DevExtreme DropDown

## Options

1. **dataSource**: Object. The data source that provides items for the DropDown.
2. **contentTemplate**: Function. Specifies a custom template for the DropDown content.
3. **displayExpr**: String or Function. Specifies the data field whose values are displayed by the DropDown.
4. **valueExpr**: String or Function. Specifies the data field whose values are used as DropDown values.
5. **selectionMode**: String. Specifies the selection mode. Possible values: “single”, “multiple”, “all”.
6. **showSelectionControls**: Boolean. Specifies whether to display selection controls.
7. **placeholder**: String. The placeholder text displayed in the DropDown.
8. **showDropDownButton**: Boolean. Specifies whether to show the drop-down button.
9. **acceptCustomValue**: Boolean. Specifies whether a user can enter a custom value.

## Events

1. **onSelectionChanged**: Function. A function that is executed after the selection is changed.
2. **onItemClick**: Function. A function that is executed when an item is clicked.
3. **onValueChanged**: Function. A function that is executed after the value is changed.
4. **onOpened**: Function. A function that is executed after the DropDown is opened.
5. **onClosed**: Function. A function that is executed after the DropDown is closed.

## Methods

1. **open()**: Opens the DropDown.
2. **close()**: Closes the DropDown.
3. **toggle()**: Toggles the DropDown between open and closed states.
4. **focus()**: Sets focus on the DropDown.
5. **blur()**: Removes focus from the DropDown.
6. **reset()**: Resets the DropDown value to the default value.
7. **selectAll()**: Selects all items in the DropDown.
8. **unselectAll()**: Unselects all items in the DropDown.

# DevExtreme NumberBox

## Options

1. **format**: String. Specifies the format of the number to be displayed. Possible values: “0.00”, “#.##”, “,0”, “#0.##%”, “#0.##; -#0.##”.
2. **mode**: String. Specifies the mode of the NumberBox. Possible values: “text”, “number”, “tel”.
3. **min**: Number. The minimum value allowed.
4. **max**: Number. The maximum value allowed.
5. **showSpinButtons**: Boolean. Specifies whether to show spin buttons.
6. **useLargeSpinButtons**: Boolean. Specifies whether to use large spin buttons. Mostly used for desktop.
7. **step**: Number. Specifies the step by which the value is increased or decreased using spin buttons.
8. **onCopy**: Function. A function that is executed after a copy action.
9. **onCut**: Function. A function that is executed after a cut action.
10. **onPaste**: Function. A function that is executed after a paste action.
11. **onKeyDown**: Function. A function that is executed when a key is pressed down.
12. **onKeyUp**: Function. A function that is executed when a key is released.
13. **onInput**: Function. A function that is executed when the input is changed.
14. **onFocusIn**: Function. A function that is executed when the NumberBox gets focus.

## Events

1. **onCopy**: Function. A function that is executed after a copy action.
2. **onCut**: Function. A function that is executed after a cut action.
3. **onPaste**: Function. A function that is executed after a paste action.
4. **onKeyDown**: Function. A function that is executed when a key is pressed down.
5. **onKeyUp**: Function. A function that is executed when a key is released.
6. **onInput**: Function. A function that is executed when the input is changed.
7. **onFocusIn**: Function. A function that is executed when the NumberBox gets focus.

## Methods

1. **open()**: Opens the NumberBox.
2. **close()**: Closes the NumberBox.
3. **toggle()**: Toggles the NumberBox between open and closed states.
4. **focus()**: Sets focus on the NumberBox.
5. **blur()**: Removes focus from the NumberBox.
6. **reset()**: Resets the NumberBox value to the default value.
7. **selectAll()**: Selects all items in the NumberBox.
8. **unselectAll()**: Unselects all items in the NumberBox.

# DevExtreme SelectBox with Toggle Buttons

## Options

1. **dataSource**: Object. The data source that provides items for the SelectBox.
2. **valueExpr**: String or Function. Specifies the data field whose values are used as SelectBox values.
3. **width**: Number or String. Specifies the SelectBox width.
4. **displayExpr**: String or Function. Specifies the data field whose values are displayed by the SelectBox.
5. **searchEnabled**: Boolean. Specifies whether the search box is enabled.
6. **grouped**: Boolean. Specifies whether the items are grouped.
7. **noDataText**: String. The text displayed when there is no data.
8. **opened**: Boolean. Specifies whether the drop-down editor is opened.
9. **searchMode**: String. Specifies the search mode. Possible values: “startswith”, “contains”.
10. **showDataBeforeSearch**: Boolean. Specifies whether to show all items when the search box is empty.
11. **showDropDownButton**: Boolean. Specifies whether to show the drop-down button.
12. **spellcheck**: Boolean. Specifies whether the browser spell check is enabled.
13. **groupTemplate**: Function. Specifies a custom template for the group headers.
14. **itemTemplate**: Function. Specifies a custom template for the items.
15. **buttons**: Array. Specifies the buttons displayed in the SelectBox.

## Events

1. **onValueChanged**: Function. A function that is executed after the value is changed.

## Methods

1. **open()**: Opens the SelectBox.
2. **close()**: Closes the SelectBox.
3. **toggle()**: Toggles the SelectBox between open and closed states.
4. **focus()**: Sets focus on the SelectBox.
5. **blur()**: Removes focus from the SelectBox.
6. **reset()**: Resets the SelectBox value to the default value.
7. **selectAll()**: Selects all items in the SelectBox.
8. **unselectAll()**: Unselects all items in the SelectBox.

# DevExtreme TextArea

## Options

1. **stylingMode**: String. Specifies the styling mode of the TextArea. Possible values: “outlined”, “underlined”, “filled”.
2. **autoResizeEnabled**: Boolean. Specifies whether the TextArea automatically resizes based on content.
3. **maxHeight**: Number or String. Specifies the maximum height of the TextArea.
4. **minHeight**: Number or String. Specifies the minimum height of the TextArea.
5. **maxLength**: Number or String. Specifies the maximum number of characters that can be entered.
6. **placeholder**: String. The placeholder text displayed in the TextArea.
7. **spellcheck**: Boolean. Specifies whether the browser spell check is enabled.

## Events

1. **onValueChanged**: Function. A function that is executed after the value is changed.

## Methods

1. **open()**: Opens the TextArea.
2. **close()**: Closes the TextArea.
3. **toggle()**: Toggles the TextArea between open and closed states.
4. **focus()**: Sets focus on the TextArea.
5. **blur()**: Removes focus from the TextArea.
6. **reset()**: Resets the TextArea value to the default value.
7. **selectAll()**: Selects all text in the TextArea.
8. **unselectAll()**: Unselects all text in the TextArea.

# DevExtreme TextBox Mask Demo

## Options

1. **mask**: String. Specifies the mask to be applied on the TextBox input.
2. **placeholder**: String. Specifies the placeholder text displayed in the TextBox.
3. **showMaskMode**: String. Specifies when the mask is displayed. Possible values: “always”, “onFocus”.
4. **useMaskedValue**: Boolean. Specifies whether the value should include mask characters.
5. **maskRules**: Object. Specifies custom rules for mask symbols.
6. **maskChar**: String. Specifies the character used to represent the absence of input in the mask.

## Events

1. **onValueChanged**: Function. A function that is executed after the value is changed.

## Methods

1. **open()**: Opens the TextBox.
2. **close()**: Closes the TextBox.
3. **toggle()**: Toggles the TextBox between open and closed states.
4. **focus()**: Sets focus on the TextBox.
5. **blur()**: Removes focus from the TextBox.
6. **reset()**: Resets the TextBox value to the default value.
7. **selectAll()**: Selects all text in the TextBox.
8. **unselectAll()**: Unselects all text in the TextBox.

# DevExtreme Button Demo

## Options

1. **stylingMode**: String. Specifies the styling mode of the Button. Possible values: “contained”, “outlined”, “text”.
2. **text**: String. Specifies the text displayed on the Button.
3. **type**: String. Specifies the type of the Button. Possible values: “normal”, “success”, “default”, “danger”.
4. **width**: Number or String. Specifies the width of the Button.
5. **icon**: String. Specifies the icon to be displayed on the Button.
6. **onClick**: Function. A function that is executed when the Button is clicked.
7. **template**: Function. Specifies a custom template for the Button.
8. **useSubmitBehavior**: Boolean. Specifies whether the Button should have submit behavior.

## Events

1. **onClick**: Function. A function that is executed when the Button is clicked.

## Methods

1. **open()**: Opens the Button.
2. **close()**: Closes the Button.
3. **toggle()**: Toggles the Button between open and closed states.
4. **focus()**: Sets focus on the Button.
5. **blur()**: Removes focus from the Button.
6. **reset()**: Resets the Button to the default state.
7. **selectAll()**: Selects all text in the Button.
8. **unselectAll()**: Unselects all text in the Button.

# DevExtreme File Uploader Demo

## Options

1. **multiple**: Boolean. Specifies whether multiple files can be selected.
2. **accept**: String. Specifies the types of files that the file uploader accepts.
3. **width**: Number or String. Specifies the width of the file uploader.
4. **value**: Array. Specifies the selected files.
5. **allowCanceling**: Boolean. Specifies whether file upload can be canceled.
6. **uploadMode**: String. Specifies the upload mode. Possible values: “instantly”, “useButtons”, “useForm”.
7. **uploadUrl**: String. Specifies the URL where files are uploaded.
8. **chunkSize**: Number. Specifies the size of a chunk in bytes.
9. **name**: String. Specifies the name of the file input.
10. **dialogTrigger**: String. Specifies the selector for the element that triggers the file dialog.
11. **dropZone**: String. Specifies the selector for the drop zone element.
12. **invalidFileExtensionMessage**: String. Specifies the message displayed when the file extension is invalid.
13. **invalidMaxFileSizeMessage**: String. Specifies the message displayed when the file size exceeds the maximum size.
14. **invalidMinFileSizeMessage**: String. Specifies the message displayed when the file size is below the minimum size.
15. **uploadAbortedMessage**: String. Specifies the message displayed when the upload is aborted.
16. **readyToUploadMessage**: String. Specifies the message displayed when the file is ready to be uploaded.
17. **labelText**: String. Specifies the label text for the file uploader.
18. **uploadedMessage**: String. Specifies the message displayed when the file is uploaded successfully.
19. **uploadFailedMessage**: String. Specifies the message displayed when the upload fails.
20. **uploadButtonText**: String. Specifies the text for the upload button.
21. **selectButtonText**: String. Specifies the text for the select button.
22. **maxFileSize**: Number. Specifies the maximum file size in bytes.
23. **minFileSize**: Number. Specifies the minimum file size in bytes.
24. **showFileList**: Boolean. Specifies whether to show the file list.
25. **uploadHeaders**: Object. Specifies headers for the upload request.

## Events

1. **onBeforeSend**: Function. A function that is executed before sending the upload request.
2. **onUploadStarted**: Function. A function that is executed when the upload starts.
3. **onProgress**: Function. A function that is executed to report the progress of the file upload.
4. **onDropZoneEnter**: Function. A function that is executed when the file enters the drop zone.
5. **onDropZoneLeave**: Function. A function that is executed when the file leaves the drop zone.
6. **onValueChanged**: Function. A function that is executed after the value is changed.
7. **onUploadError**: Function. A function that is executed when the upload fails.
8. **onFilesUploaded**: Function. A function that is executed when all files are uploaded.
9. **onUploadAborted**: Function. A function that is executed when the upload is aborted.

## Methods

1. **open()**: Opens the file uploader.
2. **close()**: Closes the file uploader.
3. **toggle()**: Toggles the file uploader between open and closed states.
4. **focus()**: Sets focus on the file uploader.
5. **blur()**: Removes focus from the file uploader.
6. **reset()**: Resets the file uploader to the default state.
7. **selectAll()**: Selects all files in the file uploader.
8. **unselectAll()**: Unselects all files in the file uploader.

# DevExtreme RadioGroup Demo

## Options

1. **items**: Array. Specifies the data source for the RadioGroup.
2. **value**: Any. Specifies the currently selected value.
3. **itemTemplate**: Function. Specifies a custom template for the RadioGroup items.
4. **valueExpr**: String or Function. Specifies the data field whose values are used as RadioGroup values.
5. **displayExpr**: String or Function. Specifies the data field whose values are displayed by the RadioGroup.
6. **layout**: String. Specifies the layout of the RadioGroup. Possible values: “horizontal”, “vertical”.

## Events

1. **onValueChanged**: Function. A function that is executed after the value is changed.

## Methods

1. **open()**: Opens the RadioGroup.
2. **close()**: Closes the RadioGroup.
3. **toggle()**: Toggles the RadioGroup between open and closed states.
4. **focus()**: Sets focus on the RadioGroup.
5. **blur()**: Removes focus from the RadioGroup.
6. **reset()**: Resets the RadioGroup value to the default state.
7. **selectAll()**: Selects all items in the RadioGroup.
8. **unselectAll()**: Unselects all items in the RadioGroup.

# DevExtreme Validator Demo

## Options

1. **validationMessageMode**: String. Specifies when the validation message is shown. Possible values: “always”, “auto”.
2. **placeholder**: String. Specifies the placeholder text displayed in the input field.
3. **mask**: String. Specifies the mask to be applied on the input field.
4. **inputAttr**: Object. Specifies the HTML attributes for the input element.
5. **mode**: String. Specifies the mode of the input field. Possible values: “text”, “password”, “email”, etc.
6. **value**: Any. Specifies the current value of the input field.
7. **applyButtonText**: String. Specifies the text displayed on the Apply button.
8. **pickerType**: String. Specifies the type of the date picker. Possible values: “calendar”, “rollers”, “list”.
9. **useMaskBehavior**: Boolean. Specifies whether to use masked behavior.

## Validation Rules

1. **type**: String. Specifies the type of the validation rule. Possible values: “required”, “pattern”, “stringLength”, “compare”, “range”, “email”.
2. **message**: String. Specifies the validation error message.
3. **pattern**: String. Specifies the regular expression pattern for the pattern validation rule.
4. **min**: Number. Specifies the minimum length for the stringLength validation rule.
5. **max**: Number. Specifies the maximum length for the stringLength validation rule.
6. **comparisonTarget**: Function. Specifies the target value for the compare validation rule.

## Events

1. **onValueChanged**: Function. A function that is executed after the value is changed.
2. **onClick**: Function. A function that is executed when the button is clicked.

## Methods

1. **validate()**: Validates the input field.
2. **option()**: Gets or sets an option’s value.