# DEVEXTREME

## 1 Basics of DevExtreme

### 1.1 What is DevExpress and DevExtreme?

* DevExpress is component vendor for variety of controls for different technologies.

### 1.2 Nuget Package installation

* Install-Package DevExtreme -Version 21.1.3

### 1.3 jQuery widgets

* Any devextreme UI component must be placed inside a container.
* dxButton is a jQuery plugin provided by devextreme to each UI component.
* To configure the UI component u have to pass **an object** to the plugin. The props of this object act as mirror to the props of the UI component.
* $(“#container”).dxButton() is different from $(“#conatiner”).dxButton(“instance”).
* At first place dx create a dx-button-content class div inside the conatiner and returns the conatiner jQuery element.
* At second, dx returns the button widget instance using which u can call dx’s method or access its properties.

### 1.4 Call methods of dx widgets

* There are several ways to invoke a dx UI component methods.

#### 1.4.1 Using the widget instance:

*myWidget.methodName();*

#### 1.4.2 Using the jQuery interface:

*$(“#myWidgetElement”).dxMyWidget(“methodName”);*

#### 1.4.3 Using DOM Element:

* First get raw widget instance,
* Then get the dx instance of that widget,
* And the it is as same as 1.4.1.

*var widgetElement = document.getElementById(“myWidgetElement”);*

*var widgetInstance = DevExpress.ui.getWidgetInstance(widgetElement);*

*widgetInstance.methodName();*

#### 1.4.3 Using global widget instance method

For some widget, there are global method available on ‘DevExpress.ui’, which can be used to interact with widgets.

E.g., DevExpress.ui.notify to show notification for some widget.

### 1.5 DevExpress.ui namespace

* DevExpress.ui is a namespace that contains global utility methods and properties related to UI components.
* Task performed by DevExpress.ui:
  1. Notification
  2. Widget Registraction
  3. Widget Lookup
  4. Widget Theme

### 1.6 Accessing custom handlers

*let buttonInstance = $("#container").dxButton("instance");*

*buttonInstance.\_options.\_optionManager.\_options.onClick*

*document.getElementById(“#container”).jQuery351039286975644576462.dxButton.\_options.\_optionManager.\_options.onClick*

* Another way to register an event in devextreme is using .on method. It allows you to subscribe to events at runtime and even to attach several handlers to a single event. E.g., myWidgetInstance.on(“click”, function(){  
   alert(“click1”);  
  });

myWidgetInstance.on(“click”, function(){  
 alert(“click2”);  
});

* To manually locate this handlers:

*document.getElementById(“#container”).jQuery351039286975644576462*.dxButton.\_eventsStrategy.\_events.click.\_list

#### Unsubscribing from an event handler

To unsubscribe from an event handler attached using .on() method. You have to invoke .off() method. (Both invokes \_eventStrategy methods only).

myWidgetInstance.off(“click”) => empty the \_list attached to click event in \_eventStrategy instance.

myWidgetInstance.off(“click”, handlerRef) => pops out the handlerRef from the \_list attached to click event in \_eventStrategy instance.

To unsubscribe from handler attached using .onEventName property, u have to do, myWidgetInstance.option(“eventName”, undefined);

## 2 Editors

### DevExtreme Components Options (Version 21.1)

#### Common Options

Data Binding

* dataSource: Specifies the data source for the component.
* value: Specifies the current value.
* valueExpr: Specifies which data field provides the component's value.
* displayExpr: Specifies which data field provides the display text.
* valueChangeEvent: Specifies the DOM events that trigger updates to the value.

Appearance

* elementAttr: Specifies the global attributes to be attached to the component's root element.
* height: Specifies the component's height.
* width: Specifies the component's width.
* visible: Specifies whether the component is visible.
* rtlEnabled: Switches the component to a right-to-left representation.

Behavior

* disabled: Specifies whether the component is disabled.
* readOnly: Specifies whether the component is read-only.
* tabIndex: Specifies the tab index of the component.

Events

* onValueChanged: A function that is executed after the component's value is changed.
* onInitialized: A function that is executed after the component is initialized.
* onDisposing: A function that is executed before the component is disposed of.

#### Check Box

##### Options

**accessKey**: Specifies the shortcut key that sets focus on the widget.

**activeStateEnabled**: Specifies whether the widget changes its state when interacting with a user.

**focusStateEnabled**: Specifies whether the widget can be focused.

**height**: Specifies the widget's height.

**hoverStateEnabled**: Specifies whether the widget changes its state when a user pauses on it.

**name**: Specifies the name attribute for the underlying input element.

**readOnly**: Specifies whether the widget is read-only.

**tabIndex**: Specifies the number of the element when the Tab key is used for navigating.

**text**: The text displayed by the widget.

**validationError**: Specifies the message displayed in the tooltip in case of validation error.

##### Methods

**blur**(): Removes focus from the widget.

**focus**(): Sets focus to the widget.

##### Events

**onValueChanged**: Occurs when the widget's value changes.

#### Date Box

##### Options

**accessKey**: Specifies the shortcut key that sets focus on the widget.

**activeStateEnabled**: Specifies whether the widget changes its state when interacting with a user.

**applyButtonText**: The text displayed on the button that applies the changes made in the drop-down calendar.

**cancelButtonText**: The text displayed on the button that cancels the changes made in the drop-down calendar.

**calendarOptions**: Specifies options for the drop-down calendar.

**dateSerializationFormat**: Specifies the format string that is used to serialize date values.

**displayFormat**: Specifies the date display format.

**format**: Specifies the format string that is used to display the date.

**interval**: Specifies the interval between adjacent values in the popup list in minutes.

**max**: The maximum date that can be selected within the widget.

**min**: The minimum date that can be selected within the widget.

**opened**: Specifies whether the drop-down calendar is opened.

**placeholder**: The text displayed by the widget when the widget is empty.

**readOnly**: Specifies whether the widget is read-only.

**showAnalogClock**: Specifies whether or not an analog clock is displayed in the drop-down calendar.

**text**: The text displayed by the widget.

##### Methods

**blur**(): Removes focus from the widget.

**close**(): Closes the drop-down calendar.

**focus**(): Sets focus to the widget.

**open**(): Opens the drop-down calendar.

**repaint**(): Redraws the widget.

**reset**(): Resets the widget's value to null.

**setDate**(date): Sets a new date to the widget.

Events

**onValueChanged**: Occurs when the widget's value changes.

**onOpened**: Occurs when the drop-down calendar is opened.

**onClosed**: Occurs when the drop-down calendar is closed.

#### Drop Down Box

##### Options

**acceptCustomValue**: Specifies whether the widget allows an end-user to enter a custom value.

**accessKey**: Specifies the shortcut key that sets focus on the widget.

**activeStateEnabled**: Specifies whether the widget changes its state when interacting with a user.

**contentTemplate**: Specifies a custom template for the drop-down content.

**dataSource**: Specifies the data source for the widget.

**deferRendering**: Specifies whether to render the drop-down content when it is displayed.

**displayExpr**: Specifies the data field whose values should be displayed.

**dropDownOptions**: Specifies options for the drop-down list.

**fieldTemplate**: Specifies a custom template for the text field.

**focusStateEnabled**: Specifies whether the widget can be focused.

**height**: Specifies the widget's height.

**hint**: Specifies the text displayed when the mouse pointer is over the widget.

**items**: An array of items to be displayed by the widget.

**opened**: Specifies whether the drop-down is opened.

**placeholder**: The text displayed by the widget when the widget is empty.

**readOnly**: Specifies whether the widget is read-only.

**searchEnabled**: Specifies whether or not the widget supports searching.

**showClearButton**: Specifies whether the widget displays a button that clears the value.

**tabIndex**: Specifies the number of the element when the Tab key is used for navigating.

##### Methods

**blur**(): Removes focus from the widget.

**close**(): Closes the drop-down.

**focus**(): Sets focus to the widget.

**open**(): Opens the drop-down.

**repaint**(): Redraws the widget.

##### Events

**onValueChanged**: Occurs when the widget's value changes.

**onOpened**: Occurs when the drop-down is opened.

**onClosed**: Occurs when the drop-down is closed.

#### **Number** Box

##### Options

**accessKey**: Specifies the shortcut key that sets focus on the widget.

**activeStateEnabled**: Specifies whether the widget changes its state when interacting with a user.

**buttons**: Specifies options for the spin buttons.

**format**: Specifies the format string that is used to format the value displayed by the widget.

**invalidValueMessage**: Specifies the message displayed in the tooltip if the widget's value is not a valid number.

**max**: The maximum value that can be entered into the widget.

**min**: The minimum value that can be entered into the widget.

**mode**: Specifies how the widget's value can be changed using the spin buttons.

**placeholder**: The text displayed by the widget when the widget is empty.

**readOnly**: Specifies whether the widget is read-only.

**showSpinButtons**: Specifies whether the spin buttons are visible.

**step**: Specifies the step by which the widget's value changes when using spin buttons.

**value**: Specifies the value of the widget.

##### Methods

**blur**(): Removes focus from the widget.

**focus**(): Sets focus to the widget.

##### Events

**onValueChanged**: Occurs when the widget's value changes.

#### Select Box

##### Options

**accessKey**: Specifies the shortcut key that sets focus on the widget.

**activeStateEnabled**: Specifies whether the widget changes its state when interacting with a user.

**dataSource**: Specifies the data source for the widget.

**displayExpr**: Specifies which data field provides display values for items.

**disabled**: Specifies whether the widget is disabled.

**dropDownOptions**: Specifies options for the drop-down list.

**focusStateEnabled**: Specifies whether the widget can be focused.

**grouped**: Specifies whether or not the drop-down items should be grouped.

**groupTemplate**: Specifies a custom template for group headers.

**height**: Specifies the widget's height.

**hint**: Specifies the text displayed when the mouse pointer is over the widget.

**itemTemplate**: Specifies a custom template for items.

**items**: An array of items to be displayed by the widget.

**placeholder**: The text