

## Basics of Devextreme

DevExtreme is UI (user interface) components for building modern web and mobile applications. Developed by DevExpress.

It is for react , angular , vue.

It is in include topic

Data grid,

Charts

Form control

Navigation menu etc.

## Installation – NuGet Package

Cdn files

Local files:

```
<head>
  <!-- ... -->
  <script type="text/javascript" src="js/jquery-3.5.1.min.js"></script>

  <!-- DevExtreme theme -->
  <link rel="stylesheet" href="css/dx.light.css">

  <!-- DevExtreme library -->
  <script type="text/javascript" src="js/dx.all.js"></script>
  <!-- <script type="text/javascript" src="js/dx.web.js"></script> -->
  <!-- <script type="text/javascript" src="js/dx.viz.js"></script> -->
</head>
<body class="dx-viewport">
  <!-- ... -->
</body>
```

## **Widget Basics - jQuery**

A widget in DevExtreme is a reusable UI component that encapsulates a specific functionality or behavior.

Follow steps.

Initialization -> options -> method -> events

### **Create and Configure a Widget:**

Include the Required DevExtreme Files

Create an HTML Element for the Widget

Initialize the Widget Using jQuery

Configure Widget Options

### **Get a Widget Instance**

get a widget instance, you can use `dxWidgetName("instance")` method, where `WidgetName` is the name of the widget you are working with (e.g., `dxDataGrid`, `dxTextBox`, `dxButton`, etc.).

### **Get and Set Options:**

You can get the current value of a widget's option or set a new value to it programmatically using the `option()` method.

### **Call Methods:**

To call a method on a widget, you need to access the widget instance using the `dxWidgetName("instance")` method. Once you have the instance, you can call any available method on that widget.

## **Handle Events**

you can handle events by passing an event handler function as part of the widget's configuration or by using the `onEventname` option in the widget initialization.

## **Destroy a Widget:**

To destroy a widget, call the `dispose()` method on the widget's instance.

## **2 . Editors – Overview:**

It is collection to use to take a user input field

It is used to include data entry , validation, grouping easy handle

### **2.1 Check Box:**

`dxCheckBox` widget in DevExtreme is a user interface element that allows the user to select or deselect an option.

disable or enable the checkbox, and programmatically get or set the checkbox value.

Event: `optionChanged` `valueChanged`  
`registerKeyHandler`

Options:

`text`, `value`, `hint`, `focusstateenabled`, `accesskey`, `tabindex`, `rtlenabled`, `beginupdate`, `endupdate` etc.

## 2.2 Date Box:

The `dxDateBox` widget in DevExtreme is used to provide a date input box where users can select or input a date.

It can display a calendar for easy date selection and can be customized to handle different date formats, validation rules, and more.

To initialize the **`dxDateBox`** widget, target an HTML element and use the `dxDateBox` method to configure it.

`type ,value ,dateSerializationFormat, invalidDateMessage  
acceptCustomValue,accessKey,activeStateEnabled: true,  
applyButtonText,cancelButtonText,applyValueMode,height  
width, maxLength,hoverStateEnabled`

## 2.3 Drop Down Box

The **`dxDropDownBox`** widget in DevExtreme provides a drop-down list from which users can select a value. It is useful when you need a list of items to be presented for the user to pick one from.

`Name,placeholder, acceptCustomValue,displayExpr,  
showDropDownButton,contentTemplate,  
onClosed,hoverStateEnabled,showClearButton,  
optionChanged,focus,blur,keydown`

## 2.4 Number Box

The **dxNumberBox** widget allows users to input a number. It includes support for setting minimum and maximum values, step increments, and more.

Used all off content:

type,placeholder,activeStateEnabled,  
disabled,elementAttr,max,min,mode,  
showClearButton,showSpinButtons,step,  
visible,useLargeSpinButtons,value,format and other event

## 2.5 Select Box

The **dxSelectBox** widget is a drop-down list where users can select a single value from a list. It can display a list of strings or objects.

Used:

dataSource,valueExpr,displayExpr  
searchEnabled,searchMode,onOpened,  
grouped,placeholder,hint,contains,startswith

## 2.6 Text Area

The **dxTextArea** widget allows the user to input multiline text. It's commonly used for input fields that require a longer response, like comments or descriptions.

Used:

accessKey,activeStateEnabled,autoResizeEnabled,  
maxHeight,maxLength,minHeight,onCopy,onCut,stylingMode  
value,spellcheck,

## 2.7 Text Box

The **dxTextBox** widget is a simple text input field. It supports single-line input and can handle various options like placeholders, validation, and input restrictions.

Used:

Placeholder,mask,maskchar,maskInvalidMessage,  
maxLength,showMaskMode,useMaskedValue,onFieldClick,  
stylingMode,mode,buttons

## 2.8 Button:

The **dxButton** widget represents a button control that can trigger actions when clicked.

Types of button:

Normal,success,default,danger

stylingMode,text,stylingMode,icon

## 2.9 FileUploader

The **dxFileUploader** widget allows users to upload files to the server.

It supports features like drag and drop, multiple file uploads, and progress tracking.

abortUpload,allowCanceling,  
allowedFileExtensions,Accept,selectButtonText,  
invalidFileExtensionMessage,  
invalidMaxFileSizeMessage,  
invalidMinFileSizeMessage  
accessKey,uploadFailedMessage,  
labelText,maxFileSize,minFileSize  
multiple,showFileList,onProgress,  
readyToUploadMessage,uploadMethod,uploadMode  
uploadUrl,dropZone,uploadMode

## 2.10 Validation

Validation can be done using the **validationRules** option, and you can handle validation events like **onValidated** to take action when the validation passes or fails.

It is array type for validationrules.

Type of validation:

Stringlength, number , email , password , compare , custom