**8 Dialogues And Notification**

# **LoadIndicator**

The LoadIndicator is a UI element notifying the viewer that a process is in progress.

**Specify Dimensions**

LoadIndicator allows us to specify its dimensions. We can set the [*height*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadIndicator/Configuration/#height) and [*width*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadIndicator/Configuration/#width) properties to the size we need.

**Apply a Custom Indicator**

If we want to apply a custom indicator, we can assign the path to a new image or GIF to the *[indicatorSrc](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadIndicator/Configuration/" \l "indicatorSrc" \t "_blank)* property.

We can use *LoadIndicator* within other UI components also.

# **LoadPanel**

The LoadPanel is an overlay component used to notify users that a process is in progress.

**Show and Hide the Indicator and Pane**

LoadPanel elements *(a message and an animated load indicator)* are displayed on a *pane*. If we want to hide it, we can set the *[showPane](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/" \l "showPane" \t "_blank)* property to  *false*. We can also disable the *[showIndicator](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/" \l "showIndicator" \t "_blank)* property to *hide* the animated *load indicator*. In this case, the LoadPanel displays only the [message](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/#message).

**Configure the Background Shade**

When LoadPanel is displayed, it shades the background. We can use the *[shadingColor](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/" \l "shadingColor" \t "_blank)* property to specify the color of the shade. We can also specify the element that should be shaded. For this, assign the element's CSS selector to the [container](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/#container) property. If we don’t want to shade the background, *disable* the [*shading*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/#shading) property.

**Show and Hide LoadPanel**

We can change the LoadPanel visibility by setting the [*visible*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/#visible) property. Alternatively, we can call the [*show()*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Methods/#show) and [*hide()*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Methods/#hide) or [*toggle(showing)*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Methods/#toggleshowing) methods.

Users can hide LoadPanel when they click outside it if you enable the *[hideOnOutsideClick](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/" \l "hideOnOutsideClick" \t "_blank)* property.

LoadPanel also allows you to handle the show and hide events. We can use the *[onShowing](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/" \l "onShowing" \t "_blank)* and *[onHiding](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/" \l "onHiding" \t "_blank)* functions to handle the events before they occur and possibily cancel them. We can use the *[onShown](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/" \l "onShown" \t "_blank)* and *[onHidden](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxLoadPanel/Configuration/" \l "onHidden" \t "_blank)* functions to perform required actions after the events are raised.

# **Popup**

The JavaScript Popup is a pop-up window overlaying the current view.

**Show and Hide the Popup**

We can call the [*show()*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Methods/#show) method to display the popup. We can close popup, by choosing one of the following options :

*Built-in close button* (Enable the *[showCloseButton](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/" \l "showCloseButton" \t "_blank)* property to display the Close button in a popup’s top toolbar)

*Custom close button*

On outside click (Enable the *[hideOnOutsideClick](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/" \l "hideOnOutsideClick" \t "_blank)* property to allow users to hide the popup by clicking outside the component)

**Configure the popup**

The popup inner area is divided into three parts :

*Top toolbar*  
We can set *[showTitle](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/" \l "showTitle" \t "_blank)* to *true* and use the [*title*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/#title) property to specify the *caption*. The *Close button* will appear if you do *not disable* the *[showCloseButton](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/" \l "showCloseButton" \t "_blank)* property.  
Also we can add *[toolbarItems](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/toolbarItems/" \t "_blank)* markup and set each item's [toolbar](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/toolbarItems/#toolbar) property to top.

*Content area*  
To populate the popup with content, we can use the *[contentTemplate](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/" \l "contentTemplate" \t "_blank)* property.

*Bottom toolbar*  
To enable the bottom toolbar, we can declare the *[toolbarItems](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/toolbarItems/" \t "_blank)* array. We can set each item's [toolbar](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/toolbarItems/#toolbar) property to bottom.

**Resize and Position**

To specify popup size, we can use the [*height*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/#height) and [*width*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/#width) properties.

We can set the *my, at,* and *of* properties of the [*position*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/#position) object.

We uses the [*container*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/#container) property to select the container in which you want to render the popup. If we set the [*container*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/#container) property to an element on the page, the [*shading*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/#shading) applies to this element.

We can turn on the *[dragEnabled](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopup/Configuration/" \l "dragEnabled" \t "_blank)* option to allow users to *move* the popup around the page.

# **Popover**

The Popover component shows pop-up notifications within a box with an arrow that points to a specified UI element.

**Attach Popover to a Page Element**

Popover displays an arrow that points to a page element. To specify the element, set the [*target*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/#target) property to a CSS selector.

We can use the [*position*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/#position) property to position Popover relative to the target element. If we don’t specify this property, Popover is *displayed under the element*.

**Show and Hide Popover**

The *[showEvent](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/showEvent/" \t "_blank)* and *[hideEvent](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/hideEvent/" \t "_blank)* properties allow us to *show* and *hide* Popover in response to certain events. These properties can accept *one or multiple names of DOM events* or *DevExtreme UI events separated by a space character*.

We can also specify a *delay* before the events occur. We can Set the *showEvent* and *hideEvent* properties to an object with the [name](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/showEvent/#name) (one or multiple event names) and [*delay*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/showEvent/#delay) properties.

**Specify Content**

Popover consists of *content area* and a *title*. We can specify static content in the HTML markup. If Popover should display dynamic content, we can use the *[contentTemplate](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/" \l "contentTemplate" \t "_blank)* property to specify a template. To display the title, we can enable the *[showTitle](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/" \l "showTitle" \t "_blank)* property and set the [*title*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/#title) property to the title text.

**Animate Popover**

We can show and hide Popover with animation effects, by assigning an object with the *show* and *hide* fields to the [*animation*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/animation/) property. Each of these fields accepts an object that configures the animation [*type*](https://js.devexpress.com/jQuery/Documentation/ApiReference/Common/Object_Structures/animationConfig/#type) and [other properties](https://js.devexpress.com/jQuery/Documentation/ApiReference/Common/Object_Structures/animationConfig/).

**Shade the Background**

We can show Popover with a shaded background. To do this, we can enable the [*shading*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/#shading) property and specify a *[shadingColor](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxPopover/Configuration/" \l "shadingColor" \t "_blank)*.

# **Popup vs Popover**

**Usage** **:** Popups are used for more significant interactions requiring user focus, whereas Popovers are used for supplementary information related to a specific element.

**Modal vs Non-Modal** **:** Popups can be modal, blocking other interactions; Popovers are non-modal.

**Positioning :** Popups can be positioned anywhere on the screen, while Popovers are positioned relative to a specific element.

**Content Complexity :** Popups can handle complex content; Popovers are typically for simpler, more concise content.

**Overlay :** Popups usually include an overlay; Popovers do not.

# **Toast**

The Toast is a UI component that displays pop-up notifications.

We can specify one of the four predefined types of notifications, depending on the *mode* of the message :

*'info'*(A toast with a message), *'warning'*(yellow toast), *'error'*(red toast), *'success'*(green toast).

We can also customize the Toast appearance by setting the [type](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxToast/Configuration/#type) property to *'custom'* and use a *[contentTemplate](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxToast/Configuration/" \l "contentTemplate" \t "_blank)*.

If we need to specify other Toast properties in addition to [*type*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxToast/Configuration/#type) and *[displayTime](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxToast/Configuration/" \l "displayTime" \t "_blank)*, we can set other Toast properties, such as [*shading*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxToast/Configuration/#shading)*,*[*position*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxToast/Configuration/#position)*,*[*width*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxToast/Configuration/#width)*,*[*height*](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxToast/Configuration/#height)*,* and [others](https://js.devexpress.com/jQuery/Documentation/ApiReference/UI_Components/dxToast/Configuration/).