

## 1 Rotator

### 1.1 Introduction

The Essential JavaScript Rotator control displays a set of slides. Each slide may contain images or images with content or content with user-defined transition between them. It supports Data binding, Thumbnail, Pager, Dynamic number of slide move options and also supports all custom easing animation. It supports all types of image formats (jpg, gif, etc...).

### 1.2 Control structure

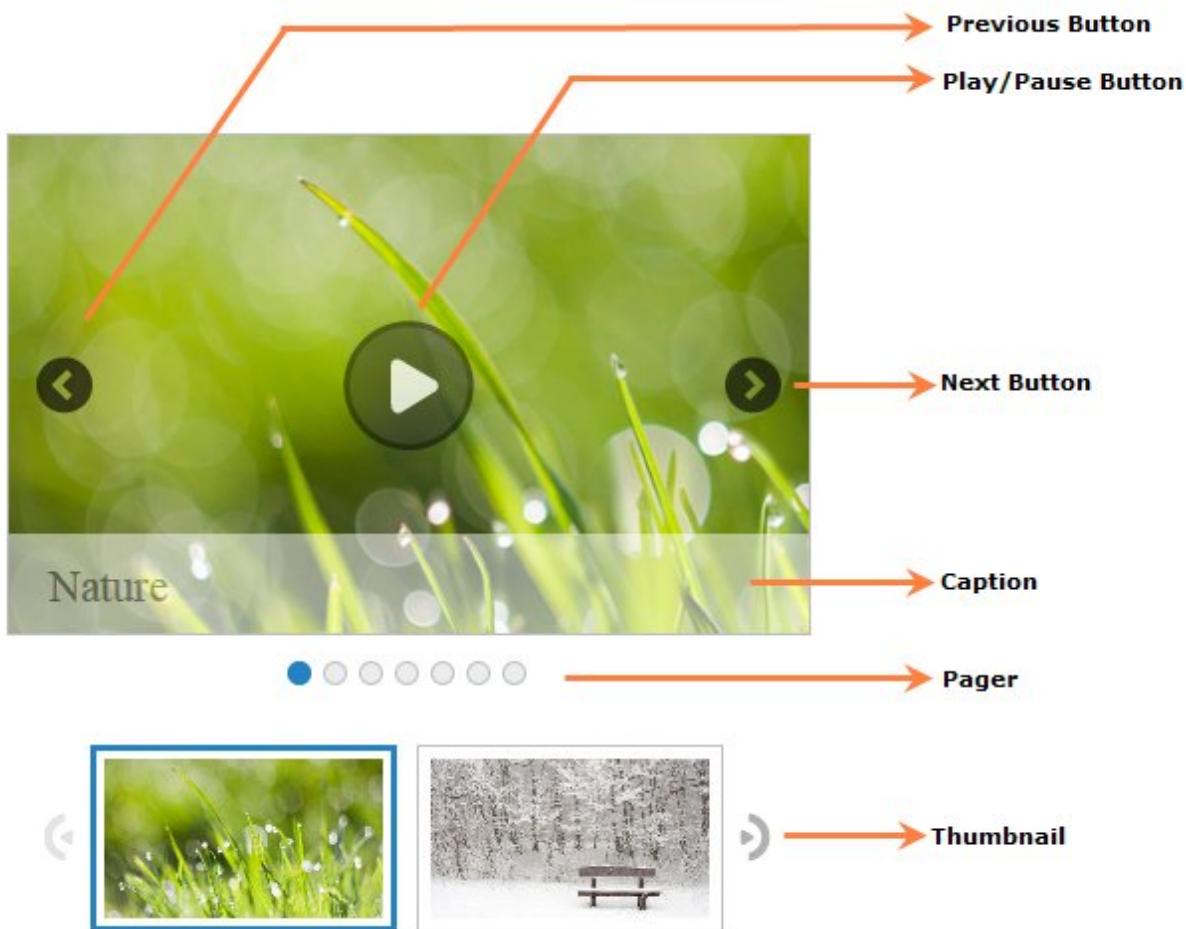


Figure 1: Essential JavaScript Rotator

### 1.3 Key Features

- Data binding: Support for data binding with JSON data as well as remote data.
- Image with Content: Supports to render with an image, content, or image with content.
- Display dynamic number of slide: Supports dynamic (1 or more) number of slides.

## User Guide Template for Concepts and Features Section

- Move dynamic number of slide: Supports to move dynamic (1 or more) number of slides at a time.
- Autoplay: Supports auto-play mode for slide transition
- Pager and Thumbnail : Supports to navigate between slides.

## 1.4 Getting Started

Essential JavaScript Rotator widget basically renders with built-in features. A user can convert the UL and LI template as a Rotator as follows:

### 1.4.1 Rotator Creation

Step 1 : Create an HTML file and paste the following template into the html file for ejRotator creation.

```
<!DOCTYPE html>

<html>

<head>
    <title>Getting Started Essential JS</title>

    <!-- style sheet for default theme(flat azure) -->
    <link href="http://cdn.syncfusion.com/js/flat-azure/ej.widgets.all-latest.min.css" rel="stylesheet" />

    <!--scripts-->
    <script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>

    <script src="http://cdnjs.cloudflare.com/ajax/libs/jquery-easing/1.3/jquery.easing.min.js"></script>

    <script src="http://cdn.syncfusion.com/js/ej.widgets.all-latest.min.js"></script>

</head>

<body>
    <!--add Rotator control template element here-->
</body>

</html>
```

## User Guide Template for Concepts and Features Section

Step 2: Add the Rotator template elements given below into the body tag to render the Rotator.

Step 3: Refer images from any location. In the following example, images are referred from images / rotator folder in sample directory.

```
<ul id="sliderContent">  
    <li></li>  
  
    <li></li>  
  
    <li></li>  
  
    <li></li>  
  
    <li></li>  
  
</ul>
```

Step 4: Initialize ejRotator in script as follows,

```
$(function () {  
    // document ready  
    // simple Rotator creation  
    $("#sliderContent").ejRotator();  
  
});
```

Step 6 : Rotator is created.

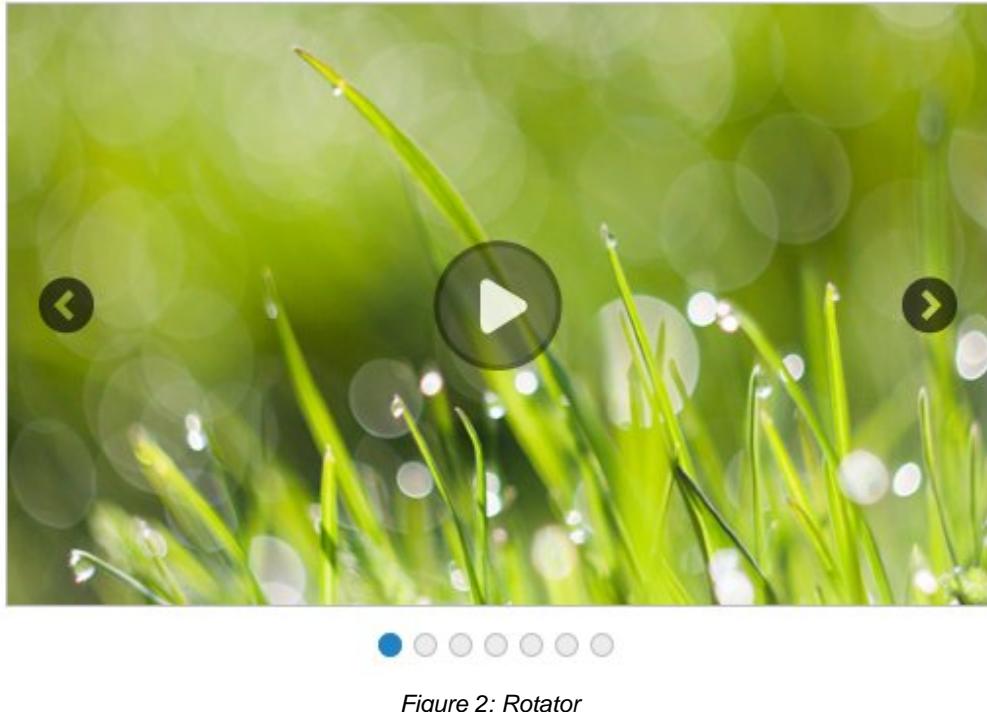


Figure 2: Rotator

## 1.5 Properties

Essential JavaScript Rotator has the following properties :

1.5.1

Name	Type	Default value
dataSource	JSON data	null

This property contains the list of data which contains a set of data fields. Each data value is used to render an item for the rotator.

The datasource property contains a subset of properties such as text, url and **htmlAttributes**. For example, the value could be an image location or a caption text for the image.

Name	Type	Default value
Text	String	null

Defines the caption of image

Name	Type	Default value

## User Guide Template for Concepts and Features Section

url	String	null
-----	--------	------

Defines the url link to set images

Name	Type	Default value
htmlAttributes	Object	null

Defines the html attributes such as id, class, styles for the items.

**Note :** The user can change the default dataField names text,url and htmlAttributes to any name by mapping datasource to fields property.

The code snippet below illustrates the usage of default data field names.

### Code sample:

Initialize the Rotator with dataSource value as shown below:

```
var items = [
    { text: "Nature", url: ".../images/rotator/nature.jpg"},
    { text: "Bird", url: ".../images/rotator/bird.jpg"},
    { text: "Sculpture", url: ".../images/rotator/sculpture.jpg"}
];
$("#sliderContent").ejRotator({dataSource: items});
```

The code snippet below illustrates the usage of user-defined data field names by mapping.

### Code sample:

Initialize the Rotator with the dataSource value as shown below

```
var items = [
    { content: "Nature", imageUrl: ".../images/rotator/nature.jpg"},
    { content: "Bird", imageUrl: ".../images/rotator/bird.jpg"},
    { content: "Sculpture", imageUrl: ".../images/rotator/sculpture.jpg"}
];
$("#sliderContent").ejRotator({
    dataSource: items,
    fields: {text: "content", url: "imageUrl"}}
```

## User Guide Template for Concepts and Features Section

```
});
```

The highlighted code snippet illustrates how to map user-defined datafield name.

Get or set the **dataSource** API, after initialization:

```
//Get the dataSource API value  
$("#sliderContent").ejRotator('option', 'dataSource');  
  
//Set the dataSource API  
$("#sliderContent").ejRotator('option', 'dataSource', items);
```

Fields	Object	null
--------	--------	------

Defines the mapping fields for the data items of the Rotator. When JSON data is used in the dataSource, fields are used to map the columns to the corresponding values.

### fields:

Name	Type	Default
Text	String	null

Defines the caption of image

Name	Type	Default
url	String	null

Defines the url link to set images

Name	Type	Default
htmlAttributes	Object	null

Defines the html attributes such as id, class, styles for the item.

### Code sample:

Initialize the Rotator with the field value as shown below :

```
var items = [
```

## User Guide Template for Concepts and Features Section

```
{ text: "Nature", url: ".../images/rotator/nature.jpg"},  
 { text: "Bird", url: ".../images/rotator/bird.jpg"},  
 { text: "Sculpture", url: ".../images/rotator/sculpture.jpg"}  
];  
  
$("#sliderContent").ejRotator({  
    dataSource: items,  
    fields: {text: "text", url: "url"}  
});
```

**Note:** In the above code snippet, we have explicitly mapped the frequency field as {text: "text"}. Since it is the default field name, it can also be omitted.

Get or set the fields API, after initialization:

```
//Get the fields API value  
$("#sliderContent").ejRotator('option', 'fields');  
  
//Set the fields API  
$("#sliderContent").ejRotator('option', 'fields', {text: "text", url: "url"});
```

Query	String	null
-------	--------	------

Used to retrieve the data from remote data. This property is used only when the remote dataSource is used.

### Code sample:

Initialize the Rotator with the query value as shown below :

```
var dataManger = ejDataManager({  
    url: http://mvc.syncfusion.com/Services/Northwnd.svc/  
});  
  
var queryString = ej.Query().from("Suppliers").select("SupplierName", "SupplierUrl");  
  
$("#sliderContent").ejRotator({  
    dataSource: dataManger,  
    query: queryString,  
    fields: {text: "SupplierName", url: "SupplierUrl"}  
});
```

## User Guide Template for Concepts and Features Section

Get or set the query API, after initialization:

```
//Get the query API value  
$("#sliderContent").ejRotator('option', 'query');  
  
//Set the query API  
$("#sliderContent").ejRotator('option', 'query', queryString);
```

<b>allowKeyboardNavigation</b>	Boolean	true
--------------------------------	---------	------

This property enables the keyboard interaction with Rotator items. The following keys can be used when the property is set to “True”.

Keys	Functions
<b>Alt+j</b>	Focus the control
<b>Down</b>	Move to previous slide
<b>Up</b>	Move to next slide
<b>Right</b>	Move to next slide
<b>Left</b>	Move to previous slide
<b>Space</b>	Play/Pause slide
<b>Alt+left</b>	Move thumb item to right and select item
<b>Alt+right</b>	Move thumb item to left and select item
<b>Enter</b>	Move to selected thumbnail item

### Code sample:

Initialize the Rotator with the allowKeyboardNavigation value as shown below :

```
$("#sliderContent").ejRotator({allowKeyboardNavigation: false});
```

## User Guide Template for Concepts and Features Section

Get or set the allowKeyboardNavigation API, after initialization:

```
//Get the allowKeyboardNavigation API value  
$("#sliderContent").ejRotator('option', 'allowKeyboardNavigation');  
  
//Set the allowKeyboardNavigation API  
$("#sliderContent").ejRotator('option', 'allowKeyboardNavigation', false);
```

<b>allowResize</b>	Boolean	true
--------------------	---------	------

This property allows to resize the rotator when the browser is resized

**Code sample:**

Initialize the Rotator with the allowResize value as shown below :

```
$("#sliderContent").ejRotator({allowResize: false});
```

Get or set the allowResize API, after initialization:

```
//Get the allowResize API value  
$("#sliderContent").ejRotator('option', 'allowResize');  
  
//Set the allowResize API  
$("#sliderContent").ejRotator('option', 'allowResize', false);
```

<b>animation</b>	string	null
------------------	--------	------

Sets the animation type for the rotator items. Animation values are “slide”, “fastslide”, “slowslide” and also all types of custom easing animations.

**Code sample:**

Initialize the Rotator with the animation value as shown below :

## User Guide Template for Concepts and Features Section

```
$("#sliderContent").ejRotator({animation: 'slide'});
```

Get or set the animation API, after initialization:

```
//Get the animation API value  
$("#sliderContent").ejRotator('option', 'animation');  
  
//Set the animation API  
$("#sliderContent").ejRotator('option', 'animation', 'slowSlide');
```

<b>autoPlay</b>	Boolean	true
-----------------	---------	------

Allows enabling / disabling the autoplay mode. Autoplay rotates the items continuously without user interference.

### Code sample:

Initialize the Rotator with the `autoPlay` value as shown below :

```
$("#sliderContent").ejRotator({autoPlay: false});
```

Get or set the `autoPlay` API, after initialization:

```
//Get the autoPlay API value  
$("#sliderContent").ejRotator('option', 'autoPlay');  
  
//Set the autoPlay API  
$("#sliderContent").ejRotator('option', 'autoPlay', false);
```

<b>caption</b>	Boolean	true
----------------	---------	------

Allows to enable/disable the caption when the rotator item source is an image. Caption text for each item should be given as `title` attribute value in the respective image tag. The caption doesn't appear when multiple items are to be displayed.

## User Guide Template for Concepts and Features Section

### **Code sample:**

Initialize the Rotator with the `caption` value as shown below :

```
$("#sliderContent").ejRotator({caption: false});
```

Get or set the `caption` API, after initialization:

```
//Get the caption API value  
$("#sliderContent").ejRotator('option', 'caption');  
  
//Set the caption API  
$("#sliderContent").ejRotator('option', 'caption', false);
```

<b>cssClass</b>	String	null
-----------------	--------	------

Sets the root class for the Rotator theme. This API helps to provide custom skinning option for Rotator control. To define the root class using this API, we need to include this root class in CSS initially.

### **Code sample:**

Initialize the Rotator with the `cssClass` value as shown below :

```
$("#sliderContent").ejRotator({ cssClass: 'gradient-time' });
```

Get or set the `cssClass` API, after initialization:

```
//Get the cssClass API value  
$("#sliderContent").ejRotator('option', 'cssClass');  
  
//Set the cssClass API  
$("#sliderContent").ejRotator('option', 'cssClass', 'gradient-time');
```

## User Guide Template for Concepts and Features Section

<b>enabled</b>	Boolean	true
----------------	---------	------

Enable or disable the rotator control.

**Code sample:**

Initialize the Rotator with the `enabled` value as shown below :

```
$("#sliderContent").ejRotator({enabled: false});
```

Get or set the `enabled` API, after initialization:

```
//Get the enabled API value  
$("#sliderContent").ejRotator('option', 'enabled');  
  
//Set the enabled API  
$("#sliderContent").ejRotator('option', 'enabled', false);
```

<b>frameSpace</b>	String or number	“10” pixel
-------------------	------------------	------------

Allows to set the space between rotator items

**Code sample:**

Initialize the Rotator with the `frameSpace` value as shown below :

```
$("#sliderContent").ejRotator({frameSpace: 10});
```

Get or set the `frameSpace` API, after initialization:

```
//Get the frameSpace API value  
$("#sliderContent").ejRotator('option', 'frameSpace');  
  
//Set the frameSpace API  
$("#sliderContent").ejRotator('option', 'frameSpace', 10);
```

## User Guide Template for Concepts and Features Section

<b>itemDisplay</b>	String or number	“1” pixel
Allows to set the number of rotator items to be displayed		
<b>Code sample:</b> Initialize the Rotator with the itemDisplay value as shown below :		
<pre>\$("#sliderContent").ejRotator({itemDisplay: 1});</pre>		
Get or set the itemDisplay API, after initialization:		
<pre>//Get the itemDisplay API value \$("#sliderContent").ejRotator('option', 'itemDisplay');  //Set the itemDisplay API \$("#sliderContent").ejRotator('option', 'itemDisplay', 1);</pre>		
<b>itemMove</b>	String or number	“1” pixel
Specifies the number of items to be navigated at single click (next / previous / play buttons) . ‘itemMove’ value should be less than or equal to itemDisplay property value.		
<b>Code sample:</b> Initialize the Rotator with the itemMove value as shown below :		
<pre>\$("#sliderContent").ejRotator({itemMove: 1});</pre>		
Get or set the itemMove API, after initialization:		
<pre>//Get the itemMove API value \$("#sliderContent").ejRotator('option', 'itemMove');  //Set the itemMove API \$("#sliderContent").ejRotator('option', 'itemMove', 1);</pre>		

## User Guide Template for Concepts and Features Section

<b>orientation</b>	String / Enum	"horizontal" (or) ej.orientation.Horizontal
--------------------	---------------	---

Specifies the orientation of slideshow and also Rotator control rendering direction either in horizontal or vertical direction.

### Code sample:

Initialize the Rotator with the orientation value as shown below :

```
$("#sliderContent").ejRotator({orientation: "vertical"});
```

Get or set the orientation API, after initialization:

```
//Get the orientation API value  
$("#sliderContent").ejRotator('option', 'orientation');  
  
//Set the orientation API  
$("#sliderContent").ejRotator('option', 'orientation', "vertical");
```

<b>pager</b>	Boolean	true
--------------	---------	------

Allows to enable /disable the pager support. Pager helps to navigate the rotator items.

### Code sample:

Initialize the Rotator with the pager value as shown below :

```
$("#sliderContent").ejRotator({pager: false});
```

Get or set the pager API, after initialization:

```
//Get the pager API value  
$("#sliderContent").ejRotator('option', 'pager');  
  
//Set the pager API  
$("#sliderContent").ejRotator('option', 'pager', false);
```

<b>pagerPosition</b>	String / Enum	outside
----------------------	---------------	---------

Pagerposition is used for positioning the pager on the rotator item. This property accepts string values such as (outside, topleft, topright ,bottomleft or bottomright ) or ej.Rotator.pagerPosition enum type.

outside: - Pager positioned in the outside of an item.

topleft: - Pager positioned in the top left corner of an item.

topright: - Pager positioned in the top right corner of an item.

bottomleft: - Pager positioned in the bottom left corner of an item.

bottomright : - Pager positioned in the bottom right corner of an item.

**Code sample:**

Initialize the Rotator with the pagerPosition value as shown below :

```
$("#sliderContent").ejRotator({pagerPosition: "topleft"});
```

Get or set the pagerPosition API, after initialization:

```
//Get the pagerPosition API value
$("#sliderContent").ejRotator('option', 'pagerPosition');

//Set the pagerPosition API
$("#sliderContent").ejRotator('option', 'pagerPosition', "topleft");
```

<b>rtl</b>	Boolean	False
------------	---------	-------

Specifies the direction of slide transition. The direction of slide transition changes based on orientation and rtl values. The possible directions of slide transition is listed below.

rtl	orientation	direction
false	horizontal	left to right
	vertical	top to bottom
true	horizontal	right to left
	vertical	bottom to top

## User Guide Template for Concepts and Features Section

### **Code sample:**

Initialize the Rotator with the rtl value as shown below :

```
$("#slideContent").ejRotator({rtl: true});
```

Get or set the rtl API, after initialization:

```
//Get the rtl API value  
$("#slideContent").ejRotator('option', 'rtl');  
  
//Set the rtl API  
$("#slideContent").ejRotator('option', 'rtl', true);
```

<b>slideButton</b>	Boolean	False
--------------------	---------	-------

Allows to enable / disable the slide buttons(next /previous buttons) placed on the rotator items. Slide buttons are used to navigate the rotator items.

### **Code sample:**

Initialize the Rotator with the slideButton value as shown below :

```
$("#slideContent").ejRotator({slideButton: false});
```

Get or set the slideButton API, after initialization:

```
//Get the slideButton API value  
$("#slideContent").ejRotator('option', 'slideButton');  
  
//Set the slideButton API  
$("#slideContent").ejRotator('option', 'slideButton', false);
```

<b>slideWidth</b>	String or number	"300" pixel
-------------------	------------------	-------------

## User Guide Template for Concepts and Features Section

Allow to set the width of each rotator item.

### **Code sample:**

Initialize the Rotator with the `slideWidth` value as shown below :

```
$("#sliderContent").ejRotator({slideWidth: 300});
```

Get or set the `slideWidth` API, after initialization:

```
//Get the slideWidth API value  
$("#sliderContent").ejRotator('option', 'slideWidth');  
  
//Set the slideWidth API  
$("#sliderContent").ejRotator('option', 'slideWidth', 300);
```

<b>slideHeight</b>	String or number	“300” pixel
--------------------	------------------	-------------

Allow to set the height of each rotator item.

### **Code sample:**

Initialize the Rotator with the `slideHeight` value as shown below :

```
$("#sliderContent").ejRotator({slideHeight: 300});
```

Get or set the `slideHeight` API, after initialization:

```
//Get the slideHeight API value  
$("#sliderContent").ejRotator('option', 'slideHeight');  
  
//Set the slideHeight API  
$("#sliderContent").ejRotator('option', 'slideHeight', 300);
```

## User Guide Template for Concepts and Features Section

<b>speed</b>	String or number	“300”
--------------	------------------	-------

Enable to set the slide transition speed.

### **Code sample:**

Initialize the Rotator with the speed value as shown below :

```
$("#sliderContent").ejRotator({speed: 300});
```

Get or set the speed API, after initialization:

```
//Get the speed API value  
$("#sliderContent").ejRotator('option', 'speed');  
  
//Set the speed API  
$("#sliderContent").ejRotator('option', 'speed', 300);
```

<b>startIndex</b>	String or number	“2”
-------------------	------------------	-----

Used to set the slide that needs to be displayed initially.

### **Code sample:**

Initialize the Rotator with the startIndex value as shown below :

```
$("#sliderContent").ejRotator({startIndex: 2});
```

Get or set the startIndex API, after initialization:

```
//Get the startIndex API value  
$("#sliderContent").ejRotator('option', 'startIndex');  
  
//Set the startIndex API  
$("#sliderContent").ejRotator('option', 'startIndex', 2);
```

## User Guide Template for Concepts and Features Section

<b>thumbItem</b>	Boolean	False
------------------	---------	-------

Allows to enable / disable the thumbnail support. Thumbnail is used to navigate between slides. Thumbnail supports only for single slide transition. User should give source to thumbnail element through thumbSource property.

### Code sample:

Initialize the Rotator with the thumbItem value as shown below :

```
$("#slideContent").ejRotator({thumbItem: false});
```

Get or set the thumbItem API, after initialization:

```
//Get the thumbItem API value  
$("#slideContent").ejRotator('option', 'thumbItem');  
  
//Set the thumbItem API  
$("#slideContent").ejRotator('option', 'thumbItem', false);
```

<b>thumbSource</b>	String	"thumbElement"
--------------------	--------	----------------

Allows to set the source for thumbnail elements. The source has to be created as <ul><li> elements with "id" , which is given as a value for thumbSource property. Thumbnail elements are rendered with specified source.

### Example:

```
<ul id="thumbElement">  
  <li></li>  
  <li></li>  
  <li></li>  
  <li></li>  
  <li></li>  
</ul>
```

### Code sample:

Initialize the Rotator with the thumbSource value as shown below :

```
$( "#sliderContent" ).ejRotator({ thumbSource: "thumbElement" });
```

Get or set the thumbSource API, after initialization:

```
90
//Get the thumbSource API value
$( "#sliderContent" ).ejRotator('option', 'thumbSource');

//Set the thumbSource API
$( "#sliderContent" ).ejRotator('option', 'thumbSource', "thumbElement");
```

## 1.6 Methods

Essential JavaScript Rotator has flexible public methods. These methods can be accessed in different ways as follows :

- Using the object of Rotator we can call the methods of ejRotator.

1. Initialize the Rotator object

```
//initialize the Rotator object
var rotObj = $( "#sliderContent" ).data("ejRotator");
```

2. Using that object call the methods.

3. //calls the getIndex method of Rotator to get current slide index

```
rotObj.getIndex();
```

- Another way of accessing the methods is without creating object as follows

```
// to get current slide index of Rotator using getIndex method
$( "#sliderContent" ).ejRotator("getIndex");
```

Name	Parameters	Return type
getIndex()	--	int

## User Guide Template for Concepts and Features Section

This method is used to get current slide index.

Code sample:

```
// To get the current slide index value  
$("#slideContent").ejRotator("getIndex");
```

goToIndex()	(index)	--
-------------	---------	----

This method is used to move slide to specified index.

Code sample:

```
// To move slide to specified index  
$("#slideContent").ejRotator("goToIndex", 5);
```

enable()	--	--
----------	----	----

This method is used to enable the Rotator control.

Code sample:

```
// To enable the Rotator control  
$("#slideContent").ejRotator("enable");
```

disable()	--	--
-----------	----	----

This method is used to disable the Rotator control.

Code sample:

```
// To disable the Rotator control  
$("#slideContent").ejRotator("disable");
```

nextSlide()	--	--
-------------	----	----

## User Guide Template for Concepts and Features Section

This method is used to move to the next slide from the current slide. If current slide is the last slide then the first slide becomes the next slide.

Code sample:

```
// To move next slide  
$("#sliderContent").ejRotator("nextSlide");
```

previousSlide()	--	--
-----------------	----	----

This method is used to move to the previous slide from the current slide. If current slide is the first slide then last slide becomes the previous slide.

Code sample:

```
// To move previous slide  
$("#sliderContent").ejRotator("previousSlide");
```

play()	--	--
--------	----	----

This method is used to move slides continuously (or start auto play) with specified autoplay direction.

Code sample:

```
// To start auto play  
$("#sliderContent").ejRotator("play");
```

pause()	--	--
---------	----	----

This method is used to pause the auto play.

Code sample:

```
// To pause auto play
```

```
$("#sliderContent").ejRotator("pause");
```

## 1.7 Events

Name	Description
<b>Create</b>	This event is triggered when the Rotator control is initialized

### Code sample:

```
$("#sliderContent").ejRotator({
    create: function(args) {
        // args.cancel - returns the cancel option value
        // args.model - returns the Rotator model
        // args.type - returns the name of the event

        //handle the event
    }
});
```

<b>change</b>	This event is triggered when the Rotator slides are changed
---------------	---

### Code sample:

```
$("#sliderContent").ejRotator({
    change: function(args) {
        // args.cancel - returns the cancel option value
        // args.model - returns the Rotator model
        // args.id - returns the current rotator id
        // args.type - returns the name of the event
        // args.index - returns the current slide index

        //handle the event
    }
});
```

## User Guide Template for Concepts and Features Section

### **start**

This event is triggered when the autoplay start

#### **Code sample:**

```
$("#slideContent").ejRotator({
    start: function(args) {
        // args.cancel - returns the cancel option value
        // args.model - returns the Rotator model
        // args.id - returns the current rotator id
        // args.type - returns the name of the event
        // args.index - returns the current slide index

        //handle the event
    }
});
```

### **stop**

This event is triggered when the autoplay is stopped / paused

#### **Code sample:**

```
$("#slideContent").ejRotator({
    stop: function(args) {
        // args.cancel - returns the cancel option value
        // args.model - returns the Rotator model
        // args.id - returns the current rotator id
        // args.type - returns the name of the event
        // args.index - returns the current slide index

        //handle the event
    }
});
```

### **pagerClick**

This event is triggered when pager is clicked

## User Guide Template for Concepts and Features Section

### Code sample:

```
$("#sliderContent").ejRotator({  
    pagerClick: function(args) {  
        // args.cancel - returns the cancel option value  
        // args.model - returns the Rotator model  
        // args.id - returns the current rotator id  
        // args.type - returns the name of the event  
        // args.index - returns the clicked pager index  
  
        //handle the event  
    }  
});
```

### thumbClick

This event is triggered when thumbnail pager is clicked

### Code sample:

```
$("#sliderContent").ejRotator({  
    thumbClick: function(args) {  
        // args.cancel - returns the cancel option value  
        // args.model - returns the Rotator model  
        // args.id - returns the current rotator id  
        // args.type - returns the name of the event  
        // args.index - returns the clicked thumbnail item index  
  
        //handle the event  
    }  
});
```

### destroy

This event is triggered while the Rotator control is destroyed

### Code sample:

## User Guide Template for Concepts and Features Section

```
$("#sliderContent").ejRotator({
    destroy: function(args) {
        // args.cancel - returns the cancel option value
        // args.model - returns the Rotator model
        // args.type - returns the name of the event

        //handle the event
    }
});
```

## User Guide Template for Concepts and Features Section