

# JSEvent

## Constants Summary

String	<a href="#">#ACTION</a>	Constant returned by JSEvent.
String	<a href="#">#DATACHANGE</a>	Constant returned by JSEvent.
String	<a href="#">#DOUBLECLICK</a>	Constant returned by JSEvent.
String	<a href="#">#FOCUSGAINED</a>	Constant returned by JSEvent.
String	<a href="#">#FOCUSLOST</a>	Constant returned by JSEvent.
String	<a href="#">#FORM</a>	Constant returned by JSEvent.
Number	<a href="#">#MODIFIER_ALT</a>	Constant for the ALT modifier that can be returned by JSEvent.
Number	<a href="#">#MODIFIER_CTRL</a>	Constant for the CTRL modifier that can be returned by JSEvent.
Number	<a href="#">#MODIFIER_META</a>	Constant for the META modifier that can be returned by JSEvent.
Number	<a href="#">#MODIFIER_SHIFT</a>	Constant for the SHIFT modifier that can be returned by JSEvent.
String	<a href="#">#NONE</a>	Constant returned by JSEvent.
String	<a href="#">#RIGHTCLICK</a>	Constant returned by JSEvent.

## Property Summary

Object	<a href="#">#data</a>	A data object that specific events can set, a user can set data back to the system for events that supports this.
--------	-----------------------	---

## Method Summary

String	<a href="#">#getElementName()</a>	returns the name of the element, can be null if the form was the source of the event.
String	<a href="#">#getFormName()</a>	returns the name of the form the element was placed on.
Number	<a href="#">#getModifiers()</a>	Returns the modifiers of the event, see JSEvent.
Object	<a href="#">#getSource()</a>	returns the source component/element of the event.
Date	<a href="#">#getTimestamp()</a>	Returns the time the event occurred.
String	<a href="#">#getType()</a>	returns the event type see the JSEvents constants what it can return.
Number	<a href="#">#getX()</a>	Returns the x position of the event, relative to the component that fired it, if applicable.
Number	<a href="#">#getY()</a>	Returns the y position of the event, relative to the component that fired it, if applicable.

## Constants Details

#### ACTION

Constant returned by `JSEvent.getType()` in a method that is attached to an `onAction` event.

#### Returns

[String](#)

#### Sample

```
if (event.getType() == JSEvent.ACTION)
{
    // its an action event.
}
```

#### DATACHANGE

Constant returned by `JSEvent.getType()` in a method that is attached to an `onDataChange` event.

#### Returns

[String](#)

#### Sample

```
if (event.getType() == JSEvent.DATACHANGE)
{
    // its a data change event
}
```

#### DOUBLECLICK

Constant returned by `JSEvent.getType()` in a method that is attached to an `onDoubleClick` event.

#### Returns

[String](#)

#### Sample

```
if (event.getType() == JSEvent.DOUBLECLICK)
{
    // its a double click event.
}
```

#### FOCUSGAINED

Constant returned by `JSEvent.getType()` in a method that is attached to an `onFocusGained` or the forms `onElementFocusGained` event.

#### Returns

[String](#)

#### Sample

```
if (event.getType() == JSEvent.FOCUSGAINED)
{
    // its a focus gained event.
}
```

#### FOCUSLOST

Constant returned by `JSEvent.getType()` in a method that is attached to an `onFocusLost` or the forms `onElementFocusLost` event.

#### Returns

[String](#)

#### Sample

```
if (event.getType() == JSEvent.FOCUSLOST)
{
    // its a focus lost event.
}
```

## FORM

Constant returned by `JSEvent.getType()` in a method that is attached to a form event (like `onShow`) or command (like `onDeleteRecord`)

### Returns

[String](#)

### Sample

```
if (event.getType() == JSEvent.FORM)
{
    // its a form event or command
}
```

## MODIFIER\_ALT

Constant for the ALT modifier that can be returned by `JSEvent.getModifiers()`;

### Returns

[Number](#)

### Sample

```
//test if the SHIFT modifier is used.
if (event.getModifiers() & JSEvent.MODIFIER_SHIFT)
{
    //do shift action
}
```

## MODIFIER\_CTRL

Constant for the CTRL modifier that can be returned by `JSEvent.getModifiers()`;

### Returns

[Number](#)

### Sample

```
//test if the SHIFT modifier is used.
if (event.getModifiers() & JSEvent.MODIFIER_SHIFT)
{
    //do shift action
}
```

## MODIFIER\_META

Constant for the META modifier that can be returned by `JSEvent.getModifiers()`;

### Returns

[Number](#)

### Sample

```
//test if the SHIFT modifier is used.
if (event.getModifiers() & JSEvent.MODIFIER_SHIFT)
{
    //do shift action
}
```

## MODIFIER\_SHIFT

Constant for the SHIFT modifier that can be returned by `JSEvent.getModifiers()`;

### Returns

[Number](#)

### Sample

```
//test if the SHIFT modifier is used.
if (event.getModifiers() & JSEvent.MODIFIER_SHIFT)
{
    //do shift action
}
```

## NONE

Constant returned by `JSEvent.getType()` if the event is not used in a known event or command.

### Returns

[String](#)

### Sample

```
if (event.getType() == JSEvent.NONE)
{
    // type is not set.
}
```

## RIGHTCLICK

Constant returned by `JSEvent.getType()` in a method that is attached to an `onRightClick` event.

### Returns

[String](#)

### Sample

```
if (event.getType() == JSEvent.RIGHTCLICK)
{
    // its a right click event.
}
```

## Property Details

### data

A data object that specific events can set, a user can set data back to the system for events that supports this.

### Returns

[Object](#)

### Sample

```
// A client design method that handles ondrag
if (event.getType() == JSEvent.ONDRAG)
{
    // the data is the selected elements array
    var elements = event.data;
    // only start a client design drag when there is 1 element
    if (elements.length == 1)
    {
        return true;
    }
}

// code for a data drag method
event.data = "drag me!";
return DRAGNDROP.COPY;

// code for a data drop method
var data = event.data;
elementments[event.getElementName()].setText(data);
return true;
```

## Method Details

### getElementName

[String](#) **getElementName()**

returns the name of the element, can be null if the form was the source of the event.

### Returns

[String](#) – a String representing the element name.

### Sample

```
if (event.getElementName() == 'myElement')
{
    elements[event.getElementName()].bgcolor = '#ff0000';
}
```

### getFormName

#### String getFormName()

returns the name of the form the element was placed on.

#### Returns

String – a String representing the form name.

### Sample

```
forms[event.getFormName()].myFormMethod();
```

### getModifiers

#### Number getModifiers()

Returns the modifiers of the event, see JSEvent.MODIFIER\_XXXX for the modifiers that can be returned.

#### Returns

Number – an int which holds the modifiers as a bitset.

### Sample

```
//test if the SHIFT modifier is used.
if (event.getModifiers() & JSEvent.MODIFIER_SHIFT)
{
    //do shift action
}
```

### getSource

#### Object getSource()

returns the source component/element of the event.

If it has a name the getElementName() is the name of this component.

#### Returns

Object – an Object representing the source of this event.

### Sample

```
// cast to runtime text field (change to another kind of type if you know the type)
/** @type {RuntimeTextField} */
var source = event.getSource();
var sourceDataProvider = source.getDataProviderID();
```

### getTimestamp

#### Date getTimestamp()

Returns the time the event occurred.

#### Returns

Date – a Date when this event happened.

### Sample

```
event.getTimestamp();
```

### getType

#### String getType()

returns the event type see the JSEvents constants what it can return.

Plugins can create events with their own types.

#### Returns

String – a String representing the type of this event.

**Sample**

```
if (event.getType() == JSEvent.ACTION)
{
    // its an action event.
}
```

getX

**Number** **getX()**

Returns the x position of the event, relative to the component that fired it, if applicable.  
For example drag'n'drop events will set the x,y positions.

**Returns**

**Number** – an int representing the X position.

**Sample**

```
var x = event.getX();
var xPrevious = previousEvent.getX();
var movedXPixels = x -xPrevious;
```

getY

**Number** **getY()**

Returns the y position of the event, relative to the component that fired it, if applicable.  
For example drag'n'drop events will set the x,y positions.

**Returns**

**Number** – an int representing the Y position.

**Sample**

```
var y = event.getY();
var yPrevious = previousEvent.getY();
var movedYPixels = y -yPrevious;
```