

JSDNDEvent

Constants Summary

String	#ACTION	Constant returned by JSEvent.
String	#DATACHANGE	Constant returned by JSEvent.
String	#DOUBLECLICK	Constant returned by JSEvent.
String	#FOCUSGAINED	Constant returned by JSEvent.
String	#FOCUSLOST	Constant returned by JSEvent.
String	#FORM	Constant returned by JSEvent.
Number	#MODIFIER_ALT	Constant for the ALT modifier that can be returned by JSEvent.
Number	#MODIFIER_CTRL	Constant for the CTRL modifier that can be returned by JSEvent.
Number	#MODIFIER_META	Constant for the META modifier that can be returned by JSEvent.
Number	#MODIFIER_SHIFT	Constant for the SHIFT modifier that can be returned by JSEvent.
String	#NONE	Constant returned by JSEvent.
String	#ONDRAG	Constant returned by JSEvent.
String	#ONDRAGEND	Constant returned by JSEvent.
String	#ONDRAGOVER	Constant returned by JSEvent.
String	#ONDROP	Constant returned by JSEvent.
String	#RIGHTCLICK	Constant returned by JSEvent.

Property Summary

Object	#data	A data object that specific events can set, a user can set data back to the system for events that supports this.
String	#dataMimeType	The event data mime type.

Method Summary

Number	#getDragResult()	Returns the result of the drag action.
String	#getElementName()	returns the name of the element, can be null if the form was the source of the event.
String	#getFormName()	returns the name of the form the element was placed on.
Number	#getModifiers()	Returns the modifiers of the event, see JSEvent.
JSRecord	#getRecord()	Returns the record of the event.
Object	#getSource()	returns the source component/element of the event.
Date	#getTimestamp()	Returns the time the event occurred.
String	#getType()	returns the event type see the JSEvents constants what it can return.
Number	#getX()	Returns the x position of the event, relative to the component that fired it, if applicable.
Number	#getY()	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](#)

Also see

[.getModifiers](#)

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](#)

Also see

[.getModifiers](#)

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](#)

Also see

[.getModifiers](#)

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](#)

Also see

[.getModifiers](#)

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.
}
```

ONDRAG

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

Returns

[String](#)

Sample

```
if (event.getType() == JSEvent.ONDRAG)
{
    // its an ondrag event
    if (event.getElementName() == 'todragelement')
        return DRAGNDROP.COPY
}
```

ONDRAGEND

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

Returns

[String](#)

Sample

```
if (event.getType() == JSEvent.ONDRAGEND)
{
    // its an on drag end event.
    // return true if the drop has been completed successfully
    return event.isDropSuccess();
}
```

ONDRAGOVER

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

Returns

[String](#)

Sample

```
if (event.getType() == JSEvent.ONDRAGOVER)
{
    // its an on drag over event.
    // return true if it over the right element.
    return event.getElementName() == 'candroponelement';
}
```

ONDROP

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

Returns

[String](#)

Sample

```
if (event.getType() == JSEvent.ONDROP)
{
    // its a on drop event.
    var element = elements[event.getElementName()];
    // do drop on element
    return true;
}
```

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;
elements[event.getElementName()].setText(data);
return true;
```

`dataMimeType`

The event data mime type.

Returns

[String](#)

Sample

```
// only accept drag if data is a servoy record
function onDragOver(event)
{
    if(event.dataMimeType.indexOf("application/x-servoy-record-object") == 0) return true;
    else return false;
}
```

Method Details

`getDragResult`

[Number](#) **getDragResult()**

Returns the result of the drag action.

Returns

[Number](#) – a DRAGNDROP constant, representing the result of the drag action

Sample

```
function onDragEnd(event)
{
    var dragResult = event.getDragResult();
    if(dragResult == DRAGNDROP.NONE)
    {
        // the drag was canceled
    }
    else if(dragResult == DRAGNDROP.COPY)
    {
        // the drag ended with a copy action
    }
    else if(dragResult == DRAGNDROP.MOVE)
    {
        // the drag ended with a move action
    }
}
```

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
}
```

getRecord

JSRecord **getRecord()**

Returns the record of the event.

Returns

JSRecord – Record of the event

Sample

```
event.Record();
```

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

```
var sourceDataProvider = event.getSource().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 there 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;
```