

ELEMENT_TYPES

Constants Summary

String	#BUTTON	Constant representing an element of the Button type.
String	#CALENDAR	Constant representing an element of the Calendar type.
String	#CHECK	Constant representing an element of the Check type.
String	#COMBOBOX	Constant representing a combobox element.
String	#GROUP	Constant representing a Group of elements.
String	#HTML_AREA	Constant representing a html area element.
String	#IMAGE_MEDIA	Constant representing an element of the ImageMedia type.
String	#LABEL	Constant representing an element of the Label type.
String	#PASSWORD	Constant representing an element of the Password type.
String	#PORTAL	Constant representing an element of the Portal type.
String	#RADIOS	Constant representing an element of the Radios type.
String	#RECTANGLE	Constant representing a rectangle element.
String	#RTF_AREA	Constant representing a rtf area of element.
String	#SPLITPANE	Constant representing a splitpane element.
String	#TABPANEL	Constant representing an element of the Tabpanel type.
String	#TEXT_AREA	Constant representing an element of the TextArea type.
String	#TEXT_FIELD	Constant representing an element of the TextField type.
String	#TYPE_AHEAD	Constant representing a typeahead element.

Constants Details

BUTTON

Constant representing an element of the Button type.

Returns

[String](#)

Sample

```
//the return value for an element of the Button type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.BUTTON)
{
    // element is a Button component
}
```

CALENDAR

Constant representing an element of the Calendar type.

Returns

[String](#)

Sample

```
//the return value for an element of the Calendar type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.CALENDAR)
{
    // element is a Calendar field
}
```

CHECK

Constant representing an element of the Check type.

Returns

[String](#)

Sample

```
//the return value for an element of the Check type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.CHECK)
{
    // element is a Check(box) field
}
```

COMBOBOX

Constant representing a combobox element.

Returns

[String](#)

Sample

```
//the return value for an element of the TextField type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.COMBOBOX)
{
    // element is a COMBOBOX element
}
```

GROUP

Constant representing a Group of elements.

Returns

[String](#)

Sample

```
//the return value for an element type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.GROUP)
{
    // element is a group element
}
```

HTML_AREA

Constant representing a html area element.

Returns

[String](#)

Sample

```
//the return value for an element of the TextField type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.HTML_AREA)
{
    // element is a HTML textarea
}
```

IMAGE_MEDIA

Constant representing an element of the ImageMedia type.

Returns

[String](#)

Sample

```
//the return value for an element of the ImageMedia type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.IMAGE_MEDIA)
{
    // element is a Image Media field
}
```

LABEL

Constant representing an element of the Label type.

Returns

[String](#)

Sample

```
//the return value for an element of the Label type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.LABEL)
{
    // element is a Label component
}
```

PASSWORD

Constant representing an element of the Password type.

Returns

[String](#)

Sample

```
//the return value for an element of the Password type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.PASSWORD)
{
    // element is a Password component
}
```

PORTAL

Constant representing an element of the Portal type.

Returns

[String](#)

Sample

```
//the return value for an element of the Portal type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.PORTAL)
{
    // element is a Portal component
}
```

RADIOS

Constant representing an element of the Radios type.

Returns

[String](#)

Sample

```
//the return value for an element of the Radios type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.RADIOS)
{
    // element is a Radios field.
}
```

RECTANGLE

Constant representing a rectangle element.

Returns

[String](#)

Sample

```
//the return value for an element of the TextField type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.RECTANGLE)
{
    // element is a rectangle element
}
```

RTF_AREA

Constant representing a rtf area of element.

Returns

[String](#)

Sample

```
//the return value for an element of the TextField type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.RTF_AREA)
{
    // element is a RTF textarea.
}
```

SPLITPANE

Constant representing a splitpane element.

Returns

[String](#)

Sample

```
//the return value for an element type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.SPLITPANE)
{
    // element is a splitpane element
}
```

TABPANEL

Constant representing an element of the Tabpanel type.

Returns

[String](#)

Sample

```
//the return value for an element of the Tabpanel type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.TABPANEL)
{
    // element is a Tabpanel component
}
```

TEXT_AREA

Constant representing an element of the TextArea type.

Returns

[String](#)

Sample

```
//the return value for an element of the TextArea type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.TEXT_AREA)
{
    // element is a TextArea field
}
```

TEXT_FIELD

Constant representing an element of the TextField type.

Returns

[String](#)

Sample

```
//the return value for an element of the TextField type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.TEXT_FIELD)
{
    // element is a text field
}
```

TYPE_AHEAD

Constant representing a typeahead element.

Returns

[String](#)

Sample

```
//the return value for an element of the TextField type, as returned by the following code
var etype = elements.elementName.getElementType();
if (etype == ELEMENT_TYPES.TYPE_AHEAD)
{
    // element is a type ahead element
}
```