

Button

Event Summary

[onAction](#)
The method that is executed when the component is clicked.

Property Summary

	dataIcon Icon for a button, this must be one of: alert arrow-d arrow-l arrow-r arrow-u back bars check delete edit forward gear grid home info minus plus refresh search star
String	dataProvider The dataprovider of the component.
Boolean	displaysTags Flag that enables or disables merging of data inside components using tags (placeholders).
Boolean	enabled The enable state of the component, default true.
String	format The format that should be applied when displaying data(using dataProviderID) in the label/button.
String	location The x and y position of the component, in pixels, separated by a comma.
String	name The name of the component.
String	styleClass The name of the style class that should be applied to this component.
String	text The text that is displayed inside the component.
Boolean	visible The visible property of the component, default true.

Event Details

onAction
The method that is executed when the component is clicked.

Property Details

dataIcon

Icon for a button, this must be one of:

alert
arrow-d
arrow-l
arrow-r
arrow-u
back
bars
check
delete
edit
forward
gear
grid
home
info
minus
plus
refresh
search
star

Returns

[String](#)

dataProvider

The dataprovider of the component.

Returns

[String](#)

displaysTags

Flag that enables or disables merging of data inside components using tags (placeholders).

Tags (or placeholders) are words surrounded by %%on each side. There are data tags and standard tags. Data tags consist in names of dataproviders surrounded by%%. Standard tags are a set of predefined tags that are made available by the system.

See the "Merging data" section for more details about tags.

The default value of this flag is "false", that is merging of data is disabled by default.

Returns

[Boolean](#)

enabled

The enable state of the component, default true.

Returns

[Boolean](#) - enabled state

format

The format that should be applied when displaying data(using dataProviderID) in the label/button.

Some examples are "%%", "dd-MM-yyyy", "MM-dd-yyyy", etc.

Returns

[String](#)

location

The x and y position of the component, in pixels, separated by a comma.

Returns

[String](#)

name

The name of the component. Through this name it can also accessed in methods.

Returns

[String](#)

styleClass

The name of the style class that should be applied to this component.

When defining style classes for specific component types, their names must be prefixed according to the type of the component. For example in order to define a class names 'fancy' for fields, in the style definition the class must be named 'field.fancy'. If it would be intended for labels, then it would be named 'label.fancy'. When specifying the class name for a component, the prefix is dropped however. Thus the field or the label will have its styleClass property set to 'fancy' only.

Returns

[String](#)

text

The text that is displayed inside the component.

Returns

[String](#)

visible

The visible property of the component, default true.

Returns

[Boolean](#) - visible property