CHECK BOX

msCheckbox :

The Kendo UI for jQuery CheckBox enables you to style input type="checkbox" elements. The component allows the user to toggle between checked and unchecked states.

In this demo, you can see the CheckBox in different states, such as disabled, enabled, and checked.

CHECKBOX STATES:

The Kendo UI for jQuery Checkbox supports checked, unchecked and indeterminate states and allows users to select only one item.

Properties of Checkbox are mentioned below,

- ✓ Id*
- ✓ Savefield*

ID*:

It specifies the input id of a particular widget. Each and every widget must have an UNIQUE id. Id should accept only the numbers, alphabets and underscore.

It does not allow to type special characters in an ID and not allow to type negative integers.

Mandatory : TRUE

Example:

id - Test_123 (Will work)

id - Test-@@ (Will not accept as a widget id)

id = -123 or -test (Will not accept as a widget id)

SAVEFIELD*:

This savefield property accept only the boolean values like, True or False. It is a mandatory field.

True - All the values are saved in database

False - Values are not saved in database

LOOP:

User can use this loop with in the Checkbox then, it should mention the loop id in a particular Checkbox's loop property.

User can mention the loop in ui and rule,

Syntax:

UI --> loop (Loop id is unique and user defined)

RULE --> screenid_widgetid of loop (Like this mentioned in rule)

Checkbox widget works within the loop.

PARENT GROUP:

Checkbox widget is also used within the parent group, but should follow some rules like parent group id is properly mentioned in ui and rule.

User can mention the parent group in ui and rule,

Syntax:

UI --> pgroup (Parent group id is unique and user defined)

RULE --> screenid_parentgroupid (Like this mentioned in rule)

Checkbox widget will works within the parent group.

ACTION NAME:

Actions available for the Checkbox widget in framework 2.0 are mentioned below,

- Mandatory
- Optional
- Show
- Hide
- Enable
- Disable
- Setval

Mandatory*:

User must select a picture. They cannot skip this field, if they can try to skip means cannot allowed to submit the form.

Mandatory field is mentioned as red asterisk symbol.

Syntax:

APPLY [Mandatory] ON [#screenid_widgetid]

Example:

APPLY [Mandatory] ON [#config_test_Checkbox1]

Optional:

User can select or they can skip a particular widget, Which is allowed to submit a form.

Syntax:

APPLY [Optional] ON [#screenid_widgetid]

Example:

APPLY [Optional] ON [#config_test_Checkbox1]

Show:

This Checkbox will show/display to the user, so that the user can select the option from the list.

Syntax:

APPLY [Show] ON [#screenid_widgetid]

Example:

APPLY [Show] ON [#config_test_Checkbox1]

Hide:

This rule is to hide a mentioned widget. So it is not visible to the user.

Syntax:

APPLY [Hide] ON [#screenid_widgetid]

Example:

APPLY [Hide] ON [#config_test_Checkbox1]

Enable:

It enables the Checkbox, so the user can able to select the option from the list and an enable action is used to activate the widget from disabled state.

Syntax:

APPLY [Enable] ON [#screenid_widgetid]

Example:

APPLY [Enable] ON [#config_test_Checkbox1]

Disable:

The Kendo UI for Angular Checkbox provides a clear visual indication when the component is disabled, which can be set with a single configuration option.

It disables the Checkbox widget. The user cannot select option from the list in this disable rule, and they appear blurred.

Syntax:

APPLY [Disable] ON [#screenid_widgetid]

Example:

APPLY [Disable] ON [#config_test_Checkbox1]

Setvalue:

1. The user can set the value to "static," which means that when the form is initially opened, it displays the mentioned image in a specific Checkbox.

But this is not constant, user can change/add the image by manually at run time.

Syntax:

Setvalue = “”

APPLY [SetValue] ON [#screenid_widgetid] VALUE [""];

Example:

Setvalue = “” (So, this value is reflected in the initial page of the form)

APPLY [SetValue] ON [#config_test_Checkbox1] VALUE [""];

2. User can pass the values through variable,

Syntax:

APPLY [SetValue] ON [$tempvariable] VALUE [""];

APPLY [SetValue] ON [#screenid_widgetid] VALUE [$tempvariable];

Example:

APPLY [SetValue] ON [$test] VALUE [""];

APPLY [SetValue] ON [#config_test_Checkbox1] VALUE [$test];

3. User can get values from the another widget,

Syntax:

APPLY [SetValue] ON [#screenid_widgetid] VALUE [#screenid_widgetid];

Example:

APPLY [SetValue] ON [#config_test_Checkbox1] VALUE [#config_test_Combo2];

EVENT NAME:

Events in rule files are,

- ✓ Load
- ✓ Change

Load:

This event is should display all the values while loading on an initial page.

Syntax:

APPLY [Enable] ON [#screenid_widgetid] VALUE ["Checkbox1"];

Example:

```
FIELD_BEGIN [NAME = "config_test"]
RULE_BEGIN [NAME = "Initial Load", ORDER = "1"]
APPLY [Enable] ON [#config_test_1] VALUE ["Checkbox1"];
RULE_END
FIELD_END
```

Change:

Change event is triggered when the value of the widget is changed by the user.

Syntax:

```
IF ((#screenid_widgetid != ""))
BEGIN
APPLY [Hide] ON [#screenid_widgetid];
END
ELSE
```

```
BEGIN
APPLY [Show] ON [#screenid_widgetid];
END
```

Example:

```
FIELD_BEGIN [NAME = "config_test_1"]
RULE_BEGIN [NAME = "condition1", ORDER = "1"]
IF ((#config_test_1 != ""))
BEGIN
APPLY [Hide] ON [#config_test_2];
END
ELSE
BEGIN
APPLY [Show] ON [#config_test_2];
END
RULE_END
FIELD_END
```

ORIENTATION:

In an application displays the form in 2 ways. They are,

- ✓ Vertical orientation
- ✓ Horizontal orientation

Vertical orientation:

An application shows in portrait mode.

Horizontal orientation:

An application shows in landscape mode.