
# Workflow Designer

## custom Frontend Actions

15.03.2025

You are able to implement your own simple Frontend Actions, which you can select within “Frontend Actions” block.
This actions must be implemented within JavaScript and registered to a Plugin Manager Workflow Designer.

Create a file **<customname>.inc.php** in the
folder **modules/Workflow2/extends/frontendactions/**

Every frontend action you want to use, must registered within this file in the following way:

```
PluginFrontendAction::registerSimple('Action Label', 'Action Key',
'Callback to generate Javascript Code', array(
        // ... config fields
), 'Description of Action');
```

Example from **core.inc.php**

```
PluginFrontendAction::registerSimple('Set Value to Input Element',
'inputvalue', '\Workflow\Plugins\FrontendActions\Core::inputValue',
array(
    'inputele' => array(
        'type' => 'templatefield',
        'label' => 'Input Selector you want to set',
    ),
    'value' => array(
        'type' => 'templatefield',
        'label' => 'New value',
    ),
), 'Can be used to set a value into a input field, by directory use a
jQuery/CSS Selector. <a
href="https://www.w3schools.com/jquery/jquery_ref_selectors.asp"
class="btn btn-default" target="_blank">More information</a>');
```

| | | |
|---|---|---|
| Action Label | Label of this Action, shown in Frontend Actions configuration | string |
| Action Key | Unique Action Key for internal relationship | string |
| config fields | See next section | array |
| Description of Action | A short descriptin, what this action does, shown in Frontend Actions configuration | string |

## Configuration fields

The registration can get an array with configuration variables, you can input during Frontend Action configuration.

The **array key** is the variable name, you will use to access this value. The key "**type**", within the config field array, define the type of the input field. Possible values are:

| | |
|---|---|
| templatefield | Default text field with support for variables |
| field | picklist for selection of a field in current moduleYou can limit possible fields by use "**fieldtype**" and define a type of fields you want to select (For example "picklist") |
| checkbox | Checkbox for yes / no switch |
| colorpicker | Allow to pick a color and contain Hex code |
| picklist | A selectbox to select predefined valuesWill use Value→Label store in "**options**" array key |

## Callback

The callback only generate the JavaScript code, which is used in frontend to do the task. You get all configuration values within "**config**" variable.

**Example**

```
public function inputvalue() {
    ob_start();
    ?>
    fieldEle = jQuery(config.inputele);
    value = config.value;
    fieldEvents = fieldEle.data('events');
    if(fieldEle.hasClass('autoComplete')) {
        return;
    }
    if(fieldEle.hasClass('dateField')) {
        fieldEle.val(value).DatePickerSetDate(value, true);
    }
    if(fieldEle.hasClass('chzn-select')) {
        fieldEle.val(value).trigger('liszt:updated');
    }

    if(fieldEle.attr('type') == 'checkbox') {
        fieldEle.prop('checked', value == '1');
    }
    if(fieldEle.hasClass('sourceField')) {
```

 

```
        var obj = Vtiger_Edit_Js.getInstance();
        obj.setReferenceFieldValue(allFieldEleParent, {
            id: value,
            name: newRecord.record[field + '_display']
        });
    }
    if(fieldEle.attr('type') == 'checkbox') {
        fieldEle.prop('checked', value == '1');
    }
    if(fieldEle.attr('type') == 'text' || fieldEle.prop("tagName") ==
'TEXTAREA') {
        fieldEle.val(value);
    }

    fieldEle.trigger('keyup');
    fieldEle.trigger('focusout');

    <?php
    return ob_get_clean();
}
```

 