

Using the HelperOptions class

Table of contents

- [Using the HelperOptions class](#)
 - [Options declaration](#)
 - [Basic declaration](#)

Using the HelperOptions class

This helper is used to generate a configuration form, the values of which are stored in the `configuration` table. Example: the "Preferences" page.

Options declaration

Fields inside [brackets] are optional.

Values between {curly braces} list the possible values for this field.

```

$this->fields_options = array(
  'general' => array(
    ['title'] => $this->l('Carrier options'), // The title of the fieldset. If missing,
    default is 'Options'.
    ['top'] => $this->l('Text to display before the fieldset'), // This text is display right above the
    first. Rarely used.
    ['image'] => 'url to icon', // If missing, will use the default icon for
    the tab.
    ['description'] => $this->l('Display as description'), // Displays an informational box above the
    fields.
    ['info'] => $this->l('Display as info'), // Displays an unstyled text above the
    fields.
    'fields' => array( // The various option fields.
      'PS_CARRIER_DEFAULT' => array( // The array is named after the option's ID.
        It must be the // same name as the value stored in the
        ps_configuration table.
        ['title'] => $this->l('Default carrier:'), // The name of the option.
        ['desc'] => $this->l('The default carrier used in shop'), // The description of the option.
        ['cast'] => 'intval', // Using this option, you can cast the
        variable's content // into a known value. You can use boolval,
        floatval, intval // or strval depending on value type you
        want to receive.
        'type' => {'text', 'hidden', 'select', 'bool', 'radio', // The kind of input field you want to use.
          'checkbox', 'password', 'textarea', 'file', 'textLang',
          'textareaLang', 'selectLang'},
        ['suffix'] => 'kg', // Display after the field (ie. currency).
        // For text fields or password fields only.
        ['identifier'] => 'id_carrier', // The unique ID for the form.
        ['list'] => array(list do display as options), // For select field only.
        ['empty_message'] => $this->l('Display if list is empty'), // For select field only
        ['cols'] => 40, // For textarea fields only.
        ['rows'] => 5, // For textarea fields only.
        ['thumb'] => 'url to thumb image', // For file fields only.
        ['is_invisible'] => {true, false} // Disable the field depending on shop
        context.
      ),
      'another_field' => array(
        ...
      ),
    ),
    'submit' => array()
  ),
  'another fieldset' => ...
);

```

Basic declaration

Removing all the optional fields, this is how to build a basic HelperOptions element:

```

$this->fields_options = array(
  'general' => array(
    'title' => $this->l('Parameters'),
    'fields' => array(
      'PS_MYMODULE_OPTION1' => array(
        'title' => $this->l('Choose one'),
        'desc' => $this->l('Choose between Yes and No.'),
        'cast' => 'boolval',
        'type' => 'bool'
      ),
      'PS_MYMODULE_OPTION2' => array(
        'title' => $this->l('Add some text'),
        'desc' => $this->l('This is where you can add some text'),
        'cast' => 'strval',
        'type' => 'text',
        'size' => '10'
      )
    )
  )
);

```

This specific code generates this HTML code (simplified here for readability reasons):

```

<form id="_form" name="_form">
<fieldset>
  <legend>Parameters</legend>

  <div>
    <label>Choose one</label>
    <div>
      <label></label>
      <input type="radio" value="1" />
      <label>Yes</label>
      <label></label>
      <input type="radio" value="0" checked="checked" />
      <label>No</label>
      <p>Choose between Yes and No.</p>
    </div>
  </div>

  <div class="clear"></div>

  <div>
    <label>Add some text</label>
    <div>
      <input type="text" value="" />
      <p>This is where you can add some text</p>
    </div>
  </div>
</fieldset>
</form>

```