

source: [formality](#) / [trunk](#) / [public](#) / [class-formality-form.php](#)

Last change on this file was [2654495](#), checked in by [michelegiorgi](#), 3 years ago

release v1.5

Property svn:eol-style set to native

Property svn:executable set to *

File size: 8.7 KB

Line	
1	<code><?php</code>
2	
3	<code>/**</code>
4	<code> * Form rendering functions</code>
5	<code> *</code>
6	<code> * @link https://formality.dev</code>
7	<code> * @since 1.0</code>
8	<code> * @package Formality</code>
9	<code> * @subpackage Formality/public</code>
10	<code> * @author Michele Giorgi <hi@giorgi.io></code>
11	<code> */</code>
12	
13	<code>class Formality_Form {</code>
14	
15	<code> private \$formality;</code>
16	<code> private \$version;</code>
17	
18	<code> public function __construct(\$formality, \$version) {</code>
19	<code> \$this->formality = \$formality;</code>
20	<code> \$this->version = \$version;</code>
21	<code> \$this->form_id = get_the_ID();</code>
22	<code> \$this->form_meta = get_post_meta(\$this->form_id);</code>
23	<code> }</code>
24	
25	<code> public function option(\$single, \$default = "") {</code>
26	<code> return \$single ? (isset(\$this->form_meta["_formality_" . \$single]) && \$this->form_meta["_formality_" . \$single][0] ? \$this->form_meta["_formality_" . \$single][0] : \$default) : \$this->form_meta;</code>
27	<code> }</code>
28	
29	<code> public function fields() {</code>
30	<code> \$render = new Formality_Fields(\$this->formality, \$this->version);</code>
31	<code> \$fields = "";</code>
32	<code> \$index = 0;</code>
33	<code> if(has_blocks()) {</code>
34	<code> \$blocks = parse_blocks(get_the_content());</code>
35	<code> foreach (\$blocks as \$block) {</code>
36	<code> if(\$block['blockName']) {</code>
37	<code> \$index++;</code>
38	<code> \$type = str_replace("formality/", "", \$block['blockName']);</code>
39	<code> \$field = \$render->field(\$type, \$block['attrs'], \$this->option("type"), \$index);</code>
40	<code> \$fields .= \$field;</code>
41	<code> }</code>
42	<code> }</code>
43	<code> }</code>
44	<code> \$fields = '<div class="fo__main">' . \$fields . '</section></div>';</code>

```

Line
45     return $fields;
46 }
47
48 public function actions() {
49     $buttons = '<div class="fo__actions">';
50     $buttons .= '<button type="button" class="fo__btn fo__btn--prev">'. __( 'Previous', 'formality' ) . '</button>';
51     $buttons .= '<button type="button" class="fo__btn fo__btn--next">'. __( 'Next', 'formality' ) . '</button>';
52     $buttons .= '<button type="submit" class="fo__btn fo__btn--submit">'. ( $this->option("send_text") ? sanitize_text_field($this->option("send_text")) : __( 'Send', 'formality' ) ) . '</button>';
53     $buttons .= '</div>';
54     return $buttons;
55 }
56
57 public function nav() {
58     $nav = '<nav class="fo__nav"><ul class="fo__nav__list"></ul><ul class="fo__nav__hints fo__nav__hints--less"></ul></nav>';
59     return apply_filters('formality_form_nav', $nav, $this->form_id);
60 }
61
62 public function result() {
63     $thankyou_default = __("Thank you", "formality");
64     $thankyou_message_default = __("Your data has been successfully submitted. You are very important to us, all information received will always remain confidential. We will contact you as soon as possible.", "formality");
65     $error_default = __("Error", "formality");
66     $error_message_default = __("Something went wrong and we couldn't save your data. Please retry later or contact us by e-mail or phone.", "formality");
67     $result = '<div class="fo__result">';
68     $result .= '<div class="fo__result__success"><h3>' . ( $this->option("thankyou") ? $this->option("thankyou") : $thankyou_default ) . '</h3><p>' . ( $this->option("thankyou_message") ? $this->option("thankyou_message") : $thankyou_message_default ) . '</p></div>';
69     $result .= '<div class="fo__result__error"><h3>' . ( $this->option("error") ? $this->option("error") : $error_default ) . '</h3><p>' . ( $this->option("error_message") ? $this->option("error_message") : $error_message_default ) . '</p></div>';
70     $result .= '</div>';
71     return apply_filters('formality_form_result', $result, $this->form_id);
72 }
73
74 public function header($hide_title=false) {
75     $logo_id = $this->option("logo_id");
76     $logo = $logo_id ? wp_get_attachment_image($logo_id, "full") : '';
77     $header = '<header class="fo__header">';
78     $header .= '<div class="fo__logo">' . $logo . '</div>';
79     if(!$hide_title) { $header .= '<h2 class="fo__title">' . get_the_title() . '</h2>'; }
80     $header .= '</header>';
81     return apply_filters('formality_form_header', $header, $this->form_id);
82 }
83
84 public function body() {
85     $body = '<div class="fo__body">' . $this->fields() . $this->nav() . '</div>';
86     return apply_filters('formality_form_body', $body, $this->form_id);
87 }
88
89 public function footer() {
90     $footer = '<footer class="fo__footer">' . $this->actions() . $this->result() . $this->credits() . '</footer>';
91     return apply_filters('formality_form_footer', $footer, $this->form_id);
92 }
93
94 public function credits() {
95     $credits = nl2br($this->option("custom_credits"));
96     if($this->option("enable_credits")) {
97         if($credits) { $credits .= '<br>'; };

```

```

98     $credits .= str_replace("Formality", '<a rel="noopener noreferrer" target="_blank"
href="https://formality.dev"><svg width="45px" height="30px" viewBox="0 0 45 30" version="1.1"><path
d="M26.2434082,6 L36.7565918,6 C41.3093004,6 45,9.69069957 45,14.2434082 C45,18.7961168 41.3093004,22.4868164
36.7565918,22.4868164 L26.2434082,22.4868164 C21.6906996,22.4868164 18,18.7961168 18,14.2434082 C18,9.69069957
21.6906996,6 26.2434082,6 Z M26.2212891,19.3434082 C29.0379413,19.3434082 31.3212891,17.0600604
31.3212891,14.2434082 C31.3212891,11.426756 29.0379413,9.1434082 26.2212891,9.1434082 C23.4046368,9.1434082
21.1212891,11.426756 21.1212891,14.2434082 C21.1212891,17.0600604 23.4046368,19.3434082 26.2212891,19.3434082
Z"></path><polygon points="0 22.078125 0 0 15.65625 0 15.65625 3.640625 3.96875 3.640625 3.96875 9.21875
14.28125 9.21875 14.28125 12.859375 3.96875 12.859375 3.96875 22.078125"></polygon></svg></a>', __( 'Made with
Formality', 'formality' ));

99     $credit_bg = sprintf(__( "Photo by %s on Unsplash", "formality"), '<a rel="noopener noreferrer"
target="_blank" href="' . $this->option("credits_url") . '>' . $this->option("credits") . '</a>');

100     $credits .= $this->option("credits") ? ( '&nbsp; – &nbsp;' . str_replace("Unsplash", '<a rel="noopener
noreferrer" href="https://unsplash.com" target="_blank">Unsplash</a>', $credit_bg) ) : '';

101 }

102 $credits = $credits ? '<div class="fo__credits">' . $credits . '</div>' : '';

103 return apply_filters('formality_form_credits', $credits, $this->form_id);

104 }

105

106 public function style($embed=false, $id=true) {

107     $style = '<style>' . ( $embed ? ( $id ? '#' : '.' ) . 'formality-' . $this->form_id : ':root' ) . ' {';

108     $style .= '--formality_coll: ' . $this->option("color1") . ' ';

109     $style .= '--formality_col2: ' . $this->option("color2") . ' ';

110     $style .= '--formality_col3: ' . $this->option("color3") . ' ';

111     $style .= '--formality_bg: ' . $this->option("color2") . ' ';

112     $style .= '--formality_fontsize: ' . $this->option("fontsize") . 'px;';

113     $style .= '--formality_border: ' . ( $this->option("fontsize") < 18 ? 1 : 2 ) . 'px;';

114     $style .= '--formality_logo_height: ' . $this->option("logo_height") . 'em;';

115     $style .= '--formality_radius: ' . $this->option("border_radius") . 'px;';

116     $bg = $this->option("bg_id");

117     if($bg && (! $this->option("template"))) {

118         $bg = wp_get_attachment_image_src($bg, "full");

119         $bg_url = $bg[0];

120     } else {

121         $bg_url = $this->option("bg");

122     }

123     if($bg_url) {

124         $opacity = sprintf('%02d', $this->option("overlay_opacity"));

125         $style .= '--formality_bg_url: url(' . $bg_url . ');';

126         $style .= '--formality_bg_opacity: ' . ( $opacity == '100' ? '1' : '0.' . $opacity ) . ' ';

127         $style .= '--formality_bg_position: ' . $this->option("position") . ' ';

128         $style .= '</style>';

129         $style .= $embed ? '' : '<div class="fo__bg"></div>';

130     } else {

131         $style .= '</style>';

132     }

133     return apply_filters('formality_form_style', $style, $this->form_id);

134 }

135

136 public function sidebar($cta_label="", $invert_colors=false, $align="left", $hidden=false) {

137     $sidebar = '<div class="fo__cta-wrap fo__cta-wrap--align-' . $align . ( $hidden ? ' fo__cta-wrap--
hidden' : '' ) . '"><a id="formality-' . $this->form_id . '" href="'. get_permalink($this->form_id) . '"
class="formality-' . $this->form_id . ' fo__cta" . ($invert_colors ? " fo__cta--
invert" : "") . '">' . $cta_label . '</a></div>';

138     return apply_filters('formality_form_sidebar', $sidebar, $this->form_id);

139 }

140

141 public function print($embed=false, $attributes = array() ) {

142     $include_bg = isset($attributes['include_bg']) ? $attributes['include_bg'] : false;

143     $hide_title = isset($attributes['hide_title']) ? $attributes['hide_title'] : false;

144     if(isset($attributes['sidebar'])&&$attributes['sidebar']) {

145         $form = $this-
>sidebar($attributes['cta_label'], $attributes['invert_colors'], $attributes['align'], $attributes['disable_but
ton']) . $this->style($embed, false);

146     } else {

147         $form = '<form id="formality-' . $this->form_id . '" data-id="' . $this->form_id . '" data-
uid="' . uniqid() . '" class="fo fo--first-loading fo--' . $this->option("type") . ( $include_bg ? " fo--

```

Line	
	<pre>includebg" : "") . ' fo--' . \$this->option("style") . ' fo--layout-' . \$this- >option("bg_layout", "standard") . '" autocomplete="off" novalidate><div class="fo__wrap">' . \$this- >header(\$hide_title) . \$this->body() . \$this->footer() . '</div></form>' . \$this->style(\$embed);</pre>
148	<pre> }</pre>
149	<pre> return \$form;</pre>
150	<pre> }</pre>
151	
152	<pre>}</pre>

[About](#)

[News](#)

[Hosting](#)

[Donate](#)

[Swag](#)

[Documentation](#)

[Developers](#)

[Get Involved](#)

[Learn](#)

[Showcase](#)

[Plugins](#)

[Themes](#)

[Patterns](#)

[WordCamp](#)

[WordPress.TV](#)

[BuddyPress](#)

[bbPress](#)

[WordPress.com](#)

[Matt](#)

[Privacy](#)

