

Plugin Directory

source: [wp-seo-structured-data-schema](#) / [trunk](#) / [lib](#) / [classes](#) / [KcSeoHelper.php](#)

Last change on this file was [3289009](#), checked in by [kcseopro](#), 31 hours ago

Release version 2.8.0

File size: 18.0 KB

Line	
1	<?php
2	
3	if (!class_exists('KcSeoSettings')):
4	
5	class KcSeoHelper
6	{
7	function verifyNonce() {
8	\$nonce = !empty(\$_REQUEST['_kcseo_nonce']) ? sanitize_text_field(\$_REQUEST['_kcseo_nonce']) : null;
9	if (!wp_verify_nonce(\$nonce, \$this->nonceText())) {
10	return false;
11	}
12	
13	return true;
14	}
15	
16	function nonceText() {
17	return "kcseo_nonce_secret_text";
18	}
19	
20	function isValidBase64(\$string = null) {
21	<i>// Check if input is not a string</i>
22	if (!is_string(\$string)) {
23	return false;
24	}
25	\$decoded = @base64_decode(\$string, true);
26	<i>// Check if there is no invalid character in string</i>
27	if (!@preg_match('/^[a-zA-Z0-9\\\/\\r\\n+]*={0,2}\$/ ', \$string)) {
28	return false;
29	}
30	
31	<i>// Decode the string in strict mode and send the response</i>
32	if (!@base64_decode(\$string, true)) {
33	return false;
34	}
35	
36	<i>// Encode and compare it to original one</i>
37	if (@base64_encode(\$decoded) != \$string) {
38	return false;
39	}
40	
41	return true;
42	}
43	
44	function get_post_type_list() {
45	\$post_types = get_post_types(
46	array(
47	'public' => true
48	)
49	);

```

50     $exclude = array('attachment', 'revision', 'nav_menu_item');
51     foreach ($exclude as $ex) {
52         unset($post_types[$ex]);
53     }
54
55     return $post_types;
56 }
57
58 function kcSeoPostTypes() {
59     $post_types = get_post_types(
60         array(
61             '_builtin' => true
62         )
63     );
64     $exclude = array('attachment', 'revision', 'nav_menu_item');
65     foreach ($exclude as $ex) {
66         unset($post_types[$ex]);
67     }
68
69     return $post_types;
70 }
71
72 /**
73  * Sanitize field value
74  *
75  * @param string $fieldId
76  * @param array $field
77  * @param null $value
78  *
79  * @return array|null
80  * @internal param $value
81  */
82 function sanitize($fieldId, $field = array(), $value = null) {
83     $newValue = null;
84     $type = (!empty($field['type'])) ? sanitize_text_field($field['type']) : 'text';
85     if (is_array($field) && $value) {
86         if (in_array($fieldId, ['description', 'articleBody', 'review_body'])) {
87             $newValue = KcSeoHelper::filter_content($value);
88         } else {
89             if ($type == 'url') {
90                 $newValue = esc_url($value);
91             } else if ($type == 'textarea') {
92                 $newValue = wp_kses($value, []);
93             } else if ($field['type'] == 'group' && !empty($field['fields'])) {
94                 $newGValue = [];
95                 $groupValue = is_array($value) ? $value : [];
96                 foreach ($groupValue as $gId => $gValue) {
97                     $newVItem = [];
98                     foreach ($field['fields'] as $gFid => $fieldItem) {
99                         if (isset($gValue[$gFid])) {
100                             $newVItem[$gFid] = $this->sanitize($gFid, $fieldItem, $gValue[$gFid]);
101                         }
102                     }
103                     array_push($newGValue, $newVItem);
104                 }
105                 $newValue = $newGValue;
106             } else {
107                 $newValue = sanitize_text_field($value);
108             }
109         }
110     }
111 }
112

```

```

113         return $newValue;
114     }
115
116
117     function sanitizeOutPut($value, $type = 'text') {
118         $newValue = null;
119         if ($value) {
120             if ($type == 'text') {
121                 $newValue = esc_html(stripslashes($value));
122             } elseif ($type == 'url') {
123                 $newValue = esc_url(stripslashes($value));
124             } elseif ($type == 'textarea') {
125                 $newValue = esc_textarea(stripslashes($value));
126             } else {
127                 $newValue = esc_html(stripslashes($value));
128             }
129         }
130
131         return $newValue;
132     }
133
134     static function get_same_as($value) {
135         $sameAs = null;
136         if ($value) {
137             $sameAsRaw = preg_split('/\r\n|\r|\n/', $value);
138             $sameAsRaw = !empty($sameAsRaw) ? array_filter($sameAsRaw) : array();
139             if (!empty($sameAsRaw) && is_array($sameAsRaw)) {
140                 if (1 < count($sameAsRaw)) {
141                     $sameAs = $sameAsRaw;
142                 } else {
143                     $sameAs = $sameAsRaw[0];
144                 }
145             }
146         }
147
148         return $sameAs;
149     }
150
151
152     function imageInfo($attachment_id) {
153         $data = array();
154         $imgData = wp_get_attachment_metadata($attachment_id);
155         $data['url'] = wp_get_attachment_url($attachment_id);
156         $data['width'] = !empty($imgData['width']) ? absint($imgData['width']) : 0;
157         $data['height'] = !empty($imgData['height']) ? absint($imgData['height']) : 0;
158
159         return $data;
160     }
161
162     static function filter_content($content, $limit = 0) {
163         $content = preg_replace('#\[([^\]]+)\]#', '', wp_filter_nohtml_kses($content));
164         $content = stripslashes($content);
165         $content = convert_invalid_entities($content);
166         $content = self::characterToHTMLEntity($content);
167         if ($limit && strlen($content) > $limit) {
168             $content = mb_substr($content, 0, $limit, "utf-8");
169             $content = preg_replace('/\W\w+\s*(\W*)$/', '$1', $content);
170         }
171
172         $content = str_replace('&nbsp;', ' ', $content);
173         $content = str_replace(PHP_EOL, ' ', $content);
174         $content = str_replace("\r", '', $content);
175

```

Line	
176	return \$content;
177	}
178	
179	static function characterToHTMLEntity(\$str) {
180	\$replace = array(
181	" ",
182	'&',
183	'<',
184	'>',
185	'€',
186	'\'',
187	'\"',
188	'“',
189	'”',
190	'—',
191	'–',
192	'ı',
193	'¢',
194	'£',
195	'¤',
196	'¥',
197	'¦',
198	'§',
199	'¨',
200	'©',
201	'ª',
202	'«',
203	'¬',
204	'®',
205	'¯',
206	'°',
207	'±',
208	'²',
209	'³',
210	'´',
211	'µ',
212	'¶',
213	'·',
214	'¸',
215	'¹',
216	'º',
217	'»',
218	'¼',
219	'½',
220	'¾',
221	'¿',
222	'À',
223	'Á',
224	'Â',
225	'Ã',
226	'Ä',
227	'Å',
228	'Æ',
229	'Ç',
230	'È',
231	'É',
232	'Ê',
233	'Ë',
234	'Ì',
235	'Í',
236	'Î',
237	'Ï',
238	'Ð',

Line	
239	'Ñ',
240	'Ò',
241	'Ó',
242	'Ô',
243	'Õ',
244	'Ö',
245	'×',
246	'Ø',
247	'Ù',
248	'Ú',
249	'Û',
250	'Ü',
251	'Ý',
252	'Þ',
253	'ß',
254	'à',
255	'á',
256	'â',
257	'ã',
258	'ä',
259	'å',
260	'æ',
261	'ç',
262	'è',
263	'é',
264	'ê',
265	'ë',
266	'ì',
267	'í',
268	'î',
269	'ï',
270	'ð',
271	'ñ',
272	'ò',
273	'ó',
274	'ô',
275	'õ',
276	'ö',
277	'÷',
278	'ø',
279	'ù',
280	'ú',
281	'û',
282	'ü',
283	'ý',
284	'þ',
285	'ÿ',
286	'Œ',
287	'œ',
288	'',
289	'"',
290	'...',
291	'™',
292	'•',
293	'~',
294	);
295	
296	\$search = array(
297	'’',
298	'&',
299	'<',
300	'>',
301	'€',

Line	
302	'‘',
303	'’',
304	'“',
305	'”',
306	'–',
307	'—',
308	'¡',
309	'¢',
310	'£',
311	'¤',
312	'¥',
313	'¦',
314	'§',
315	'¨',
316	'©',
317	'ª',
318	'«',
319	'¬',
320	'®',
321	'¯',
322	'°',
323	'±',
324	'²',
325	'³',
326	'´',
327	'µ',
328	'¶',
329	'·',
330	'¸',
331	'¹',
332	'º',
333	'»',
334	'¼',
335	'½',
336	'¾',
337	'¿',
338	'À',
339	'Á',
340	'Â',
341	'Ã',
342	'Ä',
343	'Å',
344	'Æ',
345	'Ç',
346	'È',
347	'É',
348	'Ê',
349	'Ë',
350	'Ì',
351	'Í',
352	'Î',
353	'Ï',
354	'Ð',
355	'Ñ',
356	'Ò',
357	'Ó',
358	'Ô',
359	'Õ',
360	'Ö',
361	'×',
362	'Ø',
363	'Ù',
364	'Ú',

```

Line
365         '&Ucirc;',
366         '&Uuml;',
367         '&Yacute;',
368         '&THORN;',
369         '&szlig;',
370         '&agrave;',
371         '&aacute;',
372         '&acirc;',
373         '&atilde;',
374         '&auml;',
375         '&aring;',
376         '&aelig;',
377         '&ccedil;',
378         '&egrave;',
379         '&eacute;',
380         '&ecirc;',
381         '&euml;',
382         '&igrave;',
383         '&iacute;',
384         '&icirc;',
385         '&iuml;',
386         '&eth;',
387         '&ntilde;',
388         '&ograve;',
389         '&oacute;',
390         '&ocirc;',
391         '&otilde;',
392         '&ouml;',
393         '&divide;',
394         '&oslash;',
395         '&ugrave;',
396         '&uacute;',
397         '&ucirc;',
398         '&uuml;',
399         '&yacute;',
400         '&thorn;',
401         '&yuml;',
402         '&OElig;',
403         '&oelig;',
404         '&sbquo;',
405         '&bdquo;',
406         '&hellip;',
407         '&trade;',
408         '&bull;',
409         '&asymp;'
410     );
411
412     //REPLACE VALUES
413     $str = str_replace($search, $replace, $str);
414
415     //RETURN FORMATED STRING
416     return $str;
417 }
418
419 function fix1_2DataMigration() {
420     global $KcSeoWPSchema;
421     $fix_1_2 = get_option($KcSeoWPSchema->options['1_2_fix']);
422     $installed_version = get_option($KcSeoWPSchema->options['installed_version']);
423     if (empty($fix_1_2) && empty($installed_version)) {
424         global $wpdb;
425         $ids = array();
426         $posts = $wpdb->get_results("
427             SELECT posts.ID FROM {$wpdb->prefix}posts as posts

```

```

Line
428         WHERE posts.post_type = 'post'
429         AND EXISTS (
430             SELECT * FROM `{$wpdb->prefix}postmeta` as postmeta
431             WHERE postmeta.`meta_key` LIKE '_schema_%'
432             AND postmeta.`post_id`=posts.ID
433         ));
434     if ($posts) {
435         foreach ($posts as $id) {
436             $ids[] = $id->ID;
437         }
438     }
439     $pages = $wpdb->get_results("
440         SELECT posts.ID FROM {$wpdb->prefix}posts as posts
441         WHERE posts.post_type = 'page'
442         AND EXISTS (
443             SELECT * FROM `{$wpdb->prefix}postmeta` as postmeta
444             WHERE postmeta.`meta_key` LIKE '_schema_%'
445             AND postmeta.`post_id`=posts.ID
446         ));
447     if ($pages) {
448         foreach ($pages as $id) {
449             $ids[] = $id->ID;
450         }
451     }
452
453
454     if (!empty($ids)) {
455         $schemaFields = KcSeoOptions::getSchemaTypes();
456         foreach ($ids as $id) {
457             foreach ($schemaFields as $schemaID => $schema) {
458                 $schemaMetaId = $KcSeoWPSchema->KcSeoPrefix . $schemaID;
459                 $getRawMeta = get_post_meta($id, $schemaMetaId, true);
460                 $metaData = array();
461                 if (!empty($getRawMeta)) {
462                     if ($KcSeoWPSchema->isValidBase64($getRawMeta)) {
463                         $metaData = @unserialize(base64_decode($getRawMeta));
464                     } else {
465                         $metaData = @unserialize($getRawMeta);
466                     }
467                 }
468                 $metaData = (!empty($metaData) ? $metaData : array());
469                 if (!empty($metaData) && is_array($metaData)) {
470                     $metaData['active'] = true;
471                     update_post_meta($id, $schemaMetaId, $metaData);
472                 }
473             }
474         }
475     }
476
477     update_option($KcSeoWPSchema->options['1_2_fix'], true);
478 }
479
480
481 function advertisingBlock() {
482     ?>
483     <div class='kc-get-pro'>
484         <h3><?php _e("Pro Version Features", "wp-seo-structured-data-schema") ?></h3>
485         <ol>
486             <li><?php _e("Includes Auto-fill function <---Popular", "wp-seo-structured-data-schema") ?>
487             <li><?php _e("Supports Custom Post Types beyond default page and posts", "wp-seo-structured-data-schema") ?></li>
488             <li><?php _e("Supports WordPress Multisite", "wp-seo-structured-data-schema") ?></li>

```


Line	
489	<?php _e("Supports more schema types:", "wp-seo-structured-data-schema") ?>
490	
491	<?php _e("Books", "wp-seo-structured-data-schema") ?>
492	<?php _e("Courses", "wp-seo-structured-data-schema") ?>
493	<?php _e("Person", "wp-seo-structured-data-schema") ?>
494	<?php _e("Job Postings", "wp-seo-structured-data-schema") ?>
495	<?php _e("Movies", "wp-seo-structured-data-schema") ?>
496	<?php _e("Music", "wp-seo-structured-data-schema") ?>
497	<?php _e("Recipe", "wp-seo-structured-data-schema") ?>
498	<?php _e("TV Episode", "wp-seo-structured-data-schema") ?>
499	<?php _e("QAPage", "wp-seo-structured-data-schema") ?>
500	<?php _e("Item List", "wp-seo-structured-data-schema") ?>
501	
502	
503	
504	<div class="kc-pro-action"><a class='button button-primary'
505	href='https://wpsemplugins.com/downloads/wordpress-schema-
506	plugin/' target='_blank'><?php _e("Get the Pro Version", "wp-seo-
507	structured-data-schema") ?>
508	</div>
509	<?php
510	}
511	
512	function fix2_5_7_to_2_5_8() {
513	global \$KcSeoWPSchema;
514	\$installed_version = get_option(\$KcSeoWPSchema->options['installed_version']);
515	if (version_compare(\$installed_version, '2.5.7', '<=')) {
516	\$settings = get_option(\$KcSeoWPSchema->options['settings']);
517	if (isset(\$settings['contact']['contactType'])) {
518	\$settings['contact']['contactType'] = strtolower(\$settings['contact']['contactType']);
519	update_option(\$KcSeoWPSchema->options['settings'], \$settings);
520	}
521	if (isset(\$settings['area_served']) && !empty(\$settings['area_served'])) {
522	
523	\$cList = KcSeoOptions::getCountryList();
524	\$areas = array();
525	foreach (\$settings['area_served'] as \$area) {
526	\$key = array_search(\$area, \$cList);
527	if (\$key) {
528	\$areas[] = \$key;
529	}
530	}
531	if (!empty(\$areas)) {
532	\$settings['area_served'] = \$areas;
533	update_option(\$KcSeoWPSchema->options['settings'], \$settings);
534	}
535	}
536	
537	}
538	}
539	}
540	
541	endif;

[News](#)

[Hosting](#)

[Donate](#)

[Swag](#)

[Documentation](#)

[Developers](#)

[Get Involved](#)

[Learn](#)

[Showcase](#)

[Plugins](#)

[Themes](#)

[Patterns](#)

[WordCamp](#)

[WordPress.TV](#)

[BuddyPress](#)

[bbPress](#)

[WordPress.com](#)

[Matt](#)

[Privacy](#)

[Public Code](#)