

source: [wechat-social-login](#) / [trunk](#) / [add-ons](#) / [social-qq](#) / [class-xh-social-channel-qq.php](#)
@ 2080785

Last change on this file since 2080785 was 2080785, checked in by xunhuweb, 5 years ago

- 1.去掉影响网站速度的两个php文件
- 2.新增保留emoji昵称的开关
- 3.注册邮箱验证码功能
- 4.注册时进行验证码验证防止机器注册

File size: 22.0 KB

Line	
1	<?php
2	if (! defined ('ABSPATH'))
3	exit (); // Exit if accessed directly
4	
5	/**
6	* 微信接口
7	*
8	* @since 1.0.0
9	* @author ranj
10	*/
11	class XH_Social_Channel_QQ extends Abstract_XH_Social_Settings_Channel{
12	/**
13	* Instance
14	* @var XH_Social_Channel_QQ
15	* @since 1.0.0
16	*/
17	private static \$_instance;
18	
19	/**
20	* @return XH_Social_Channel_QQ
21	* @since 1.0.0
22	*/
23	public static function instance() {
24	if (is_null(self::\$_instance))
25	self::\$_instance = new self();
26	return self::\$_instance;
27	}
28	
29	/**
30	* 初始化接口ID，标题等信息
31	*
32	* @since 1.0.0
33	*/
34	protected function __construct(){
35	\$this->id='social_qq';
36	
37	\$this->icon =XH_SOCIAL_URL.'/assets/image/qq-icon.png';
38	\$this->svg ='xh-qq';
39	\$this->title =__('QQ', XH_SOCIAL);
40	\$this->description=__('在QQ互联 <a >http:="" a>注册并创建应用。',xh_social);<="" connect.qq.com<="" href="http://connect.qq.com" target="_blank" td="">
41	\$this->init_form_fields();
42	
43	\$this->supports=array('login','share');
44	\$this->enabled ='yes'== \$this->get_option('enabled');
45	}
46	
47	

```

Line
48  /**
49  * 初始化设置项
50  *
51  * {@inheritdoc}
52  * @see Abstract_XH_Social_Settings::init_form_fields()
53  * @since 1.0.0
54  */
55  public function init_form_fields(){
56      $fields =array(
57          'enabled' => array (
58              'title' => __ ( 'Enable/Disable', XH_SOCIAL ),
59              'type' => 'checkbox',
60              'label' => __ ( 'Enable QQ connect', XH_SOCIAL ),
61              'default' => 'yes'
62          ),
63          'appid'=>array(
64              'title' => __ ( 'App ID', XH_SOCIAL ),
65              'type' => 'textbox',
66              'description'=>sprintf(__('QQ互联中，请设置网站回调域为:%s',XH_SOCIAL),admin_url('admin-
ajax.php'))
67          ),
68          'appsecret'=>array(
69              'title' => __ ( 'APP Key', XH_SOCIAL ),
70              'type' => 'textbox'
71          )
72      );
73
74      $fields['subtitle2']=array(
75          'title'=>__('Cross-domain Settings',XH_SOCIAL),
76          'type'=>'subtitle',
77          'description'=>__( '多网站共用一个QQ互联平台应用进行网页授权<span style="color:red;">(需要时才开启)
</span>',XH_SOCIAL)
78      );
79
80      $fields['enabled_cross_domain']=array(
81          'title'=>__('Cross-domain',XH_SOCIAL),
82          'type'=>'section',
83          'options'=>array(
84              'cross_domain_disabled'=>__('Disabled',XH_SOCIAL),
85              'cross_domain_enabled'=>__('Enabled',XH_SOCIAL)
86          )
87      );
88
89      $fields['cross_domain_url']=array(
90          'tr_css'=>'section-enabled_cross_domain section-cross_domain_enabled',
91          'title'=>__('Cross-domain Url',XH_SOCIAL),
92          'type'=>'text',
93          'placeholder'=>'http://other-domain.com/cross-domain-qq.php',
94          'description'=>__( '如果：你需要用自己的公众号（qq，微博）作为代理，请购买：<a
href="https://www.wpweixin.net/product/1211.html" target="_blank">跨域扩展</a>',XH_SOCIAL)
95      );
96
97      $this->form_fields=apply_filters('xh_social_channel_qq_form_fields', $fields,$this);
98  }
99
100 /**
101 * {@inheritdoc}
102 * @see Abstract_XH_Social_Settings_Channel::update_wp_user_info($ext_user_id,$wp_user_id=null)
103 */
104 public function update_wp_user_info($ext_user_id,$wp_user_id=null){
105     $ext_user_info = $this->get_ext_user_info($ext_user_id);
106     if(!$ext_user_info){
107         return XH_Social_Error::error_unknown();
108     }
109
110     global $wpdb;
111     if(!$wp_user_id){

```

```

Line
112         $user_login = XH_Social::instance()->WP->generate_user_login($ext_user_info['nickname']);
113         $userdata=apply_filters('wsocial_insert_user_Info',array(
114             'user_login'=>$user_login,
115             'user_nicename'=>$ext_user_info['nicename'],
116             'first_name'=>method_exists($this, 'filter_display_name')?$this-
>filter_display_name($ext_user_info['nickname']):$ext_user_info['nickname'],
117             'user_email'=>null,
118             'display_name'=>method_exists($this, 'filter_display_name')?$this-
>filter_display_name($ext_user_info['nickname']):$ext_user_info['nickname'],
119             'nickname'=>method_exists($this, 'filter_display_name')?$this-
>filter_display_name($ext_user_info['nickname']):$ext_user_info['nickname'],
120             'user_pass'=>str_shuffle(time())
121         ),$this);
122
123         $wp_user_id = $this->wp_insert_user_Info($ext_user_id, $userdata);
124         if($wp_user_id instanceof XH_Social_Error){
125             return $wp_user_id;
126         }
127     }
128
129     if($wp_user_id!=$ext_user_info['wp_user_id']){
130         // 若当前用户已绑定过其他微信号?那么解绑
131         $wpdb->query(
132             "delete from {$wpdb->prefix}xh_social_channel_qq
133             where user_id=$wp_user_id and id<>$ext_user_id; ");
134         if(!empty($wpdb->last_error)){
135             XH_Social_Log::error($wpdb->last_error);
136             return XH_Social_Error::err_code(500);
137         }
138
139         $result = $wpdb->query(
140             "update {$wpdb->prefix}xh_social_channel_qq
141             set user_id=$wp_user_id
142             where id=$ext_user_id;");
143         if(!$result||!empty($wpdb->last_error)){
144             XH_Social_Log::error("update xh_social_channel_qq failed.detail error:".$wpdb->last_error);
145             return XH_Social_Error::err_code(500);
146         }
147     }
148
149     $ext_user_info['wp_user_id']=$wp_user_id;
150
151     do_action('xh_social_channel_update_wp_user_info',$ext_user_info);
152     do_action('xh_social_channel_qq_update_wp_user_info',$ext_user_info);
153
154     try {
155         XH_Social::instance()->WP->convert_remoteimage_to_local($wp_user_id,$ext_user_info['user_img']);
156     } catch (Exception $e) {
157         $social_img = get_user_meta($wp_user_id, '_social_img',true);
158         if(!$social_img){
159             update_user_meta($wp_user_id, '_social_img', $ext_user_info['user_img']);
160         }
161     }
162     return $this->get_wp_user_info($ext_user_id);
163 }
164
165 /**
166  * {@inheritdoc}
167  * @see Abstract_XH_Social_Settings_Channel::get_wp_user_info($ext_user_id)
168  */
169 public function get_wp_user_info($ext_user_id){
170     $ext_user_id = intval($ext_user_id);
171     global $wpdb;
172     $user = $wpdb->get_row(
173         "select w.user_id
174         from {$wpdb->prefix}xh_social_channel_qq w
175         where w.id=$ext_user_id
176         limit 1;");

```

```

Line
177     if(!$user||!$user->user_id) {
178         return null;
179     }
180
181     return get_userdata($user->user_id);
182 }
183 /**
184  * {@inheritdoc}
185  * @see Abstract_XH_Social_Settings_Channel::get_ext_user_info_by_wp($wp_user_id)
186  */
187 public function get_ext_user_info_by_wp($wp_user_id){
188     $wp_user_id = intval($wp_user_id);
189
190     global $wpdb;
191     $user = $wpdb->get_row(
192         "select w.*
193         from {$wpdb->prefix}xh_social_channel_qq w
194         where w.user_id=$wp_user_id
195         limit 1;");
196
197     if(!$user) {
198         return null;
199     }
200     $guid = XH_Social_Helper_String::guid();
201     return array(
202         'wp_user_id'=>$user->user_id,
203         'ext_user_id'=>$user->id,
204         'nickname'=>$user->nickname,
205         'user-img'=>$user->img,
206         'user_img'=>$user->img,
207         'user_login'=>null,
208         'user_email'=>null,
209         'nickname'=>$guid,
210         'uid'=>$user->openid
211     );
212 }
213
214 /**
215  * {@inheritdoc}
216  * @see Abstract_XH_Social_Settings_Channel::get_share_link()
217  */
218 public function get_share_link(){
219     return array(
220         'link'=>"http://connect.qq.com/widget/shareqq/index.html?url={url}&title={title}&summary={summary}&pics={img}",
221         'width'=>770,
222         'height'=>580
223     );
224 }
225
226 /**
227  * {@inheritdoc}
228  * @see Abstract_XH_Social_Settings_Channel::remove_ext_user_info_by_wp($wp_user_id)
229  */
230 public function remove_ext_user_info_by_wp($wp_user_id){
231     global $wpdb;
232     $wpdb->query("delete from {$wpdb->prefix}xh_social_channel_qq where user_id={$wp_user_id};");
233     if(!empty($wpdb->last_error)){
234         return XH_Social_Error::error_custom($wpdb->last_error);
235     }
236
237     return XH_Social_Error::success();
238 }
239
240 /**
241  * {@inheritdoc}
242  * @see Abstract_XH_Social_Settings_Channel::get_ext_user_info($ext_user_id)

```

```

Line
243 */
244 public function get_ext_user_info($ext_user_id){
245     $ext_user_id = intval($ext_user_id);
246     global $wpdb;
247     $user = $wpdb->get_row(
248         "select w.*
249         from {$wpdb->prefix}xh_social_channel_qq w
250         where w.id=$ext_user_id
251         limit 1;");
252     if(!$user) {
253         return null;
254     }
255     $guid = XH_Social_Helper_String::guid();
256     return array(
257         'nickname'=>$user->nickname,
258         'user_login'=>null,
259         'user_email'=>null,
260         'user_img'=>$user->img,
261         'wp_user_id'=>$user->user_id,
262         'ext_user_id'=>$user->id,
263         'nickname'=>$guid,
264         'uid'=>$user->openid
265     );
266 }
267
268 public function process_authorization_callback($wp_user_id){
269     $login_location_uri=XH_Social::instance()->session->get('social_login_location_uri');
270     if(empty($login_location_uri)){
271         $login_location_uri = home_url('/');
272     }
273
274     $userdata=array();
275
276     if(isset($_POST['user_hash'])){
277         $userdata = isset($_POST['userdata'])? base64_decode($_POST['userdata']):null;
278         $user_hash = $_POST['user_hash'];
279         $userdata =json_decode($userdata,true):null;
280         if(!$userdata){
281             return $login_location_uri;
282         }
283
284         if($user_hash!=XH_Social_Helper::generate_hash($userdata, $this-
>get_option('appsecret'))&&$user_hash!=$this->__old_generate_hash($userdata, $this-
>get_option('appsecret'))){
285             XH_Social_Log::error(__('Please check cross-domain app secret config(equal to current
website app secret)!',XH_SOCIAL));
286             return $login_location_uri;
287         }
288
289         $userdata['last_update']=date_i18n('Y-m-d H:i');
290     }else{
291         if(!isset($_GET['code'])){
292             return $login_location_uri;
293         }
294
295         try {
296             //获取accesstoken
297             $userdata = $this->get_qq_user_by_code($_GET['code'],$this->get_option("appid"),$this-
>get_option("appsecret"));
298         } catch (Exception $e) {
299             XH_Social_Log::error($e);
300             $err_times = isset($_GET['err_times'])?intval($_GET['err_times']):3;
301
302             if($err_times>0){
303                 $err_times--;
304                 return $this->_login_get_authorization_uri($wp_user_id, $err_times);
305             }
306

```

```

Line
307         XH_Social::instance()->WP->set_wp_error($login_location_uri, $e->getMessage());
308         return $login_location_uri;
309     }
310 }
311
312 if(!$userdata||empty($userdata)){
313     return $login_location_uri;
314 }
315
316 // 获取到用户信息，存储且跳转
317 global $wpdb;
318 $ext_user_id = 0;
319 $wpdb->last_error='';
320
321 try {
322     $ext_user_info = $wpdb->get_row(
323         $wpdb->prepare(
324             "select id,
325                 user_id
326             from {$wpdb->prefix}xh_social_channel_qq
327             where openid=%s
328             limit 1;", $userdata['openid']));
329
330     if($wp_user_id
331         &&$wp_user_id>0
332         &&$ext_user_info
333         &&$ext_user_info->user_id
334         &&$ext_user_info->user_id!=$wp_user_id){
335         $wp_user = get_userdata($ext_user_info->user_id);
336         if($wp_user){
337             throw new Exception(sprintf(__("对不起，您的QQ已与账户(%s)绑定，请解绑后重
338 试！",XH_SOCIAL),$wp_user->nickname));
339         }
340     }
341
342     if($wp_user_id>0
343         &&(!empty($ext_user_info)||$ext_user_info->user_id<>$wp_user_id)){
344
345         $wpdb->query("delete from {$wpdb->prefix}xh_social_channel_qq where user_id=$wp_user_id ;");
346         if(!empty($wpdb->last_error)){
347             XH_Social_Log::error($wpdb->last_error);
348             throw new Exception(__($wpdb->last_error,XH_SOCIAL));
349         }
350     }
351
352     if(!$ext_user_info){
353         if($wp_user_id>0){
354             $userdata['user_id']=$wp_user_id;
355         }
356         $wpdb->insert("{$wpdb->prefix}xh_social_channel_qq", $userdata);
357         if(!empty($wpdb->last_error)){
358             throw new Exception($wpdb->last_error);
359         }
360
361         if($wpdb->insert_id<=0){
362             XH_Social_Log::error('insert qq user info failed'.print_r($userdata,true));
363             throw new Exception('insert qq user info failed');
364         }
365
366         $ext_user_id=$wpdb->insert_id;
367     } else{
368         //user_id
369         if($wp_user_id>0){
370             $userdata['user_id'] = $wp_user_id;
371         }
372
373         $wpdb->update("{$wpdb->prefix}xh_social_channel_qq", $userdata,

```

```

Line
373         array(
374             'id'=>$ext_user_info->id
375         ));
376
377         if(!empty($wpdb->last_error)){
378             XH_Social_Log::error($wpdb->last_error);
379             throw new Exception($wpdb->last_error);
380         }
381
382         $ext_user_id=$ext_user_info->id;
383     }
384
385     return $this->process_login($ext_user_id,$wp_user_id>0);
386 } catch (Exception $e) {
387     XH_Social_Log::error($e);
388     XH_Social::instance()->WP->set_wp_error($login_location_uri, $e->getMessage());
389     return $login_location_uri;
390 }
391 }
392
393 public function get_qq_user_by_code($code,$appid,$appsecret){
394     $params=array();
395
396 $redirect_uri = XH_Social_Helper_Uri::get_uri_without_params(XH_Social_Helper_Uri::get_location_uri(),$params);
397
398     if(isset($params['code'])) unset($params['code']);
399     if(isset($params['state'])) unset($params['state']);
400     $redirect_uri.="?".http_build_query($params);
401
402     $params=array(
403         'grant_type'=>'authorization_code',
404         'code'=>$code,
405         'client_id'=>$appid,
406         'client_secret'=>$appsecret,
407         'redirect_uri'=>$redirect_uri
408     );
409
410 $response = XH_Social_Helper_Http::http_get('https://graph.qq.com/oauth2.0/token?'.http_build_query($params));
411
412     if(!$response){
413         throw new Exception(__('Nothing callback when get user info!',XH_SOCIAL),500);
414     }
415
416     if(strpos($response, "callback")!==false){
417         throw new Exception($response,500);
418     }
419
420 //access_token=0030C012FB1355A1976846C544AE5DF9&expires_in=7776000&refresh_token=1985788C72B58754385737A919C
421 //D0E50
422
423     $access_token='';
424     $results=explode('&', $response);
425     foreach ($results as $param){
426         $params = explode('=', $param);
427         if(count($params)!==2){
428             continue;
429         }
430
431         if($params[0]=== 'access_token'){
432             $access_token=$params[1];
433             break;
434         }
435     }
436
437     if(empty($access_token)){
438         throw new Exception("error callback,details:$response",500);
439     }
440 }

```

```

Line
435     $response = XH_Social_Helper_Http::http_get("https://graph.qq.com/oauth2.0/me?access_token=
{$access_token}");
436     //callback( {"client_id":"101321683","openid":"A34E43E9A2E01B1BE9147A5C1B033BF7"} );
437     $openid='';
438     if(strpos($response, "callback")!==false){
439         $lpos = strpos($response, "(");
440         $rpos = strrpos($response, ")");
441         $response = substr($response, $lpos + 1, $rpos - $lpos -1);
442         $msg = json_decode($response);
443
444         if(isset($msg->error)){
445             throw new Exception("error:{$msg->error},desc:{$msg->error_description}",500);
446         }
447
448         if(isset($msg->openid)){
449             $openid=$msg->openid;
450         }
451     }
452
453     if(empty($openid)){
454         throw new Exception("get openid error callback,details:$response",500);
455     }
456
457     $response = XH_Social_Helper_Http::http_get("https://graph.qq.com/user/get_user_info?access_token=
{$access_token}&oauth_consumer_key={$appid}&openid={$openid}");
458     if(strpos($response, "callback")!==false){
459         throw new Exception($response,500);
460     }
461
462     $obj = json_decode($response,true);
463
464     $img =null;
465     if(isset($obj['figureurl_qq_2'])&&!empty($obj['figureurl_qq_2'])){
466         $img =$obj['figureurl_qq_2'];
467     }else if(isset($obj['figureurl_qq_1'])&&!empty($obj['figureurl_qq_1'])){
468         $img =$obj['figureurl_qq_1'];
469     }else{
470         $img =$obj['figureurl_2'];
471     }
472     return array(
473         'openid'=>$openid,
474         'nickname'=>XH_Social_Helper_String::remove_emoji($obj['nickname']),
475         'gender'=>$obj['gender'],
476         'province'=>$obj['province'],
477         'city'=>$obj['city'],
478         'img'=>str_replace('http://', '///', $img),
479         'last_update'=>date_il8n('Y-m-d H:i')
480     );
481 }
482
483 public function get_wp_user($field,$field_val){
484     if(!in_array($field, array(
485         'openid'
486     ))){
487         return null;
488     }
489
490     global $wpdb;
491     $ext_user_info =$wpdb->get_row($wpdb->prepare(
492         "select user_id
493         from {$wpdb->prefix}xh_social_channel_qq
494         where $field=%s
495         limit 1;", $field_val));
496     if($ext_user_info&&$ext_user_info->user_id){
497         return get_userdata($ext_user_info->user_id);
498     }
499
500     return null;

```

```

Line
501 }
502
503 public function get_ext_user($field,$field_val){
504     if(!in_array($field, array(
505         'openid'
506     ))) {
507         return null;
508     }
509
510     global $wpdb;
511     return $wpdb->get_row($wpdb->prepare(
512         "select *
513         from {$wpdb->prefix}xh_social_channel_qq
514         where $field=%s
515         limit 1;", $field_val));
516 }
517
518 /**
519  * {@inheritdoc}
520  * @see Abstract_XH_Social_Settings_Channel::generate_authorization_uri()
521  */
522 public function generate_authorization_uri($user_ID=0,$login_location_uri=null){
523     return $this->_login_get_authorization_uri(is_null($user_ID)?0:$user_ID,0);
524 }
525
526 /**
527  * 获取登录授权链接
528  * @param string $login_location_uri
529  * @param int $error_times
530  * @return string
531  * @since 1.0.0
532  */
533 private function _login_get_authorization_uri($user_ID=0,$error_times=null){
534     $params=array();
535     $api = XH_Social_Add_On_Social_QQ::instance();
536     $url=XH_Social_Helper_Uri::get_uri_without_params(XH_Social::instance()->ajax_url(
537         array(
538             'tab'=>'authorization',
539             'action'=>"xh_social_{$api->id}",
540             'uid'=>is_null($user_ID)?0:$user_ID
541         ),true,true
542         ),$params);
543
544     if(!is_null($error_times)){
545         $params['err_times']=$error_times;
546     }
547
548     $redirect_uri= $url."?".http_build_query($params);
549
550     if('cross_domain_enabled'==$this->get_option('enabled_cross_domain')){
551         $params = array(
552             'callback'=>$redirect_uri
553         );
554
555         $params['hash'] = XH_Social_Helper::generate_hash($params, $this->get_option('appsecret'));
556         $params_uri =array();
557         $cross_domain_uri = XH_Social_Helper_Uri::get_uri_without_params($this->get_option('cross_domain_url'),$params_uri);
558
559         return $cross_domain_uri."?".http_build_query(array_merge($params_uri,$params));
560     }else{
561         return $this->get_qq_authorize_url($this->get_option('appid'),$redirect_uri);
562     }
563 }
564
565 public function get_qq_authorize_url($appid,$redirect_uri){
566     $params=array(

```

Line	
567	'response_type'=>'code',
568	'client_id'=>\$appid,
569	'scope'=>'get_user_info',
570	'redirect_uri'=>\$redirect_uri,
571	'state'=>str_shuffle(time())
572	);
573	
574	return 'https://graph.qq.com/oauth2.0/authorize?'.http_build_query(\$params);
575	}
576	}
577	require_once XH_SOCIAL_DIR.'/includes/abstracts/abstract-xh-schema.php';
578	/**
579	* 微信接口
580	*
581	* @since 1.0.0
582	* @author ranj
583	*/
584	class XH_Social_Channel_QQ_Model extends Abstract_XH_Social_Schema{
585	
586	/**
587	* {@inheritdoc}
588	* @see Abstract_XH_Social_Schema::init()
589	*/
590	public function init () {
591	\$collate=\$this->get_collate();
592	global \$wpdb;
593	\$wpdb->query("CREATE TABLE IF NOT EXISTS `{ \$wpdb->prefix}xh_social_channel_qq` (
594	`id` INT(11) NOT NULL AUTO_INCREMENT,
595	`user_id` BIGINT(20) NULL,
596	`openid` VARCHAR(64) NULL DEFAULT NULL,
597	
598	`last_update` DATETIME NOT NULL,
599	`nickname` VARCHAR(512) NULL DEFAULT NULL,
600	`gender` VARCHAR(16) NULL DEFAULT NULL,
601	`province` VARCHAR(64) NULL DEFAULT NULL,
602	`city` VARCHAR(64) NULL DEFAULT NULL,
603	`img` TEXT NULL,
604	PRIMARY KEY (`id`),
605	UNIQUE INDEX `openid_unique` (`openid`),
606	UNIQUE INDEX `user_id_unique` (`user_id`)
607	)
608	\$collate;");
609	
610	if (!empty(\$wpdb->last_error)){
611	XH_Social_Log::error(\$wpdb->last_error);
612	throw new Exception(\$wpdb->last_error);
613	}
614	}
615	}

[About](#)

[News](#)

[Hosting](#)

[Donate](#)

[Swag](#)

[Documentation](#)

[Developers](#)

[Get Involved](#)

[Learn](#)

[Showcase](#)

[Plugins](#)

[Themes](#)

[Patterns](#)

[WordCamp](#)

[WordPress.TV](#)

[BuddyPress](#)

[bbPress](#)

[WordPress.com](#)

[Matt](#)

[Privacy](#)

[Public Code](#)