```php
<?php

class Router {
    public $site = null;
    public $theme = null;

    public $public_routes = [];
    public $private_routes = [];

    public $json = false;

    public function __construct($site, $theme) {
        $this->site = $site;
        $this->theme = $theme;

        $this->public_routes = [
            '/search' => [$this, 'search'],

            '/<user>/:([^/]+)/posts.tsv' => [$this, 'feed_show_tsv'],
            '/<user>/:([^/]+)' => [$this, 'feed_show'],

            '/<user>/\+([^/]+)/posts.tsv' => [$this, 'tribune_show_tsv'],
            '/<user>/\+([^/]+)' => [$this, 'tribune_show'],

            '/<user>' => [$this, 'user_show'],

            '/tribune/([^/]+)/post' => [$this, 'legacy_tribune_show_anonymous_post'],
            '/tribune/([^/]+)/tsv' => [$this, 'legacy_tribune_show_anonymous_tsv'],
            '/tribune/([^/]+)/xml' => [$this, 'legacy_tribune_show_anonymous_xml'],
            '/tribune/([^/]+)' => [$this, 'legacy_tribune_show_anonymous'],

            '/%2B([^/]+)/posts.tsv' => [$this, 'tribune_show_anonymous_tsv'],
            '/%2B([^/]+)/posts.xml' => [$this, 'tribune_show_anonymous_xml'],
            '/%2B([^/]+)/edit' => [$this, 'tribune_edit_anonymous'],
            '/%2B([^/]+)' => [$this, 'tribune_show_anonymous'],

            '/\+([^/]+)/posts.tsv' => [$this, 'tribune_show_anonymous_tsv'],
            '/\+([^/]+)/posts.xml' => [$this, 'tribune_show_anonymous_xml'],
```

```
39                '/\+([^/]+)/edit' => [$this, 'tribune_edit_anonymous'],
40                '/\+([^/]+)' => [$this, 'tribune_show_anonymous'],
41                '/\+' => [$this, 'tribune_create_anonymous'],
42
43                '/:mine' => [$this, 'feed_show_anonymous'],
44            ];
45
46            $this->private_routes = [
47                '/new-feed' => [$this, 'feed_edit'],
48                '/<user>/:([^/]+)/edit' => [$this, 'feed_edit'],
49                '/<user>/\+([^/]+)/edit' => [$this, 'tribune_edit'],
50            ];
51
52            if (isset($_SERVER['HTTP_ACCEPT']) and $_SERVER['HTTP_ACCEPT'] == 'application/json') {
53                $this->json = true;
54            } else if (isset($_SERVER['HTTP_ACCEPT']) and $_SERVER['HTTP_ACCEPT'] == 'application/js
55            }
56        }
57
58        public function handle($uri, $get, $post) {
59            $uri = strtok($uri, '?');
60
61            if ($this->site->user) {
62                foreach ($this->private_routes as $pattern => $function) {
63                    $pattern = str_replace('<user>', '@[^/]*', $pattern);
64
65                    if (preg_match('/^' . str_replace('/', '\/', $pattern) . '$/', $uri, $matches))
66                        return $function(explode('/', $uri), $get, $post, $matches);
67                    }
68                }
69            }
70
71            foreach ($this->public_routes as $pattern => $function) {
72                $pattern = str_replace('<user>', '@[^/]*', $pattern);
73
74                if (preg_match('/^' . str_replace('/', '\/', $pattern) . '$/', $uri, $matches)) {
75                    return $function(explode('/', $uri), $get, $post, $matches);
76                }
77            }
78
79            //header("HTTP/1.0 404 Not Found");
80        }
81
82        public function search($parts, $get, $post) {
83            $query = '';
84            if (!empty($get['q'])) {
85                $query = trim($get['q']);
86            } else if (!empty($post['q'])) {
87                $query = trim($post['q']);
88            }
89
90            $tribunes = [];
91
92            if ($query) {
93                $tribunes = Tribune::select(['title LIKE' => '%'.$query.'%'], "id DESC", 10);
94                $feeds = Feed::select(['title LIKE' => '%'.$query.'%'], "id DESC", 10);
95            }
```

```
96
97          $this->theme->content = $this->theme->tribunes_list($tribunes, 'search-results');
98          $this->theme->content .= $this->theme->feeds_list($feeds, 'search-results');
99          $this->theme->sidebar = $this->site->sidebar();
100         $this->theme->topbar = $this->site->topbar();
101         print $this->theme->html();
102
103         return true;
104     }
105
106     public function feed_edit($parts, $get, $post) {
107         $user = User::load_from_url($parts[1]);
108         $feed = Feed::load_from_url($parts[2]);
109
110         if (!$feed) {
111             $feed = new Feed();
112             $feed->user_id = $this->site->user->id;
113         }
114
115         if ($user && $feed && $user->id === $feed->user_id) {
116             if (!empty($post) && $feed->fill($post)) {
117                 if ($feed->id) {
118                     $feed->update();
119                 } else {
120                     $feed->insert();
121                 }
122             }
123
124             $this->theme->content = $feed->edit();
125         }
126
127         $this->theme->sidebar = $this->site->sidebar();
128         $this->theme->topbar = $this->site->topbar();
129         print $this->theme->html();
130
131         return true;
132     }
133
134     public function feed_show($parts, $get, $post) {
135         $user = User::load_from_url($parts[1]);
136         $feed = Feed::load_from_url($parts[2]);
137
138         if ($user && $feed && $user->id === $feed->user_id) {
139             $board = new Board($feed->tribunes(), $feed->url());
140
141             if ($this->json) {
142                 $this->theme->content = $board->json();
```

miaoli / inc / router.inc.php

Top

Code Blame 535 lines (427 loc) · 14.5 KB

Raw

```
134         public function feed_show($parts, $get, $post) {
148             if ($this->json) {
149                 header('Content-Type: application/json;charset=UTF-8');
150                 print $this->theme->bare();
151             } else {
152                 $this->theme->sidebar = $this->site->sidebar();
153                 $this->theme->topbar = $this->site->topbar();
154                 print $this->theme->html();
155             }
156
157             return true;
158         }
159
160         public function feed_show_tsv($parts, $get, $post) {
161             $user = User::load_from_url($parts[1]);
162             $feed = Feed::load_from_url($parts[2]);
163
164             if ($user && $feed && $user->id === $feed->user_id) {
165                 $board = new Board($feed->tribunes(), $feed->url());
166                 $this->theme->content = $board->tsv();
167             }
168
169             header('Content-Type: text/tab-separated-values;charset=UTF-8');
170             print $this->theme->bare();
171
172             return true;
173         }
174
175         public function feed_show_anonymous($parts, $get, $post) {
176             if (!isset($_SESSION['anonymous_tribunes'])) {
177                 $_SESSION['anonymous_tribunes'] = [];
178             }
179
180             $board = new Board($_SESSION['anonymous_tribunes'], trim(implode("/", $parts), "/"));
181
182             $this->theme->title = "plop";
183             $this->theme->content = $board->show();
184
185             $this->theme->sidebar = $this->site->sidebar();
186             $this->theme->topbar = $this->site->topbar();
187             print $this->theme->html();
188
189             return true;
190         }
191
192         public function tribune_edit_anonymous($parts, $get, $post) {
193             $tribune = Tribune::load_from_url($parts[1]);
194
195             if ($tribune && $tribune->temporary) {
196                 if (isset($_SESSION['anonymous_tribunes']) and in_array($tribune, $_SESSION['anonymo
197                     if (!empty($post) && $tribune->fill($post)) {
198                         if ($tribune->id) {
199                             $tribune->update();
200                         } else {
201                             $tribune->insert();
202                         }
203
```

```
204                     header('HTTP/1.0 302 Found');
205                     header('Location: /'.$tribune->url());
206                 }
207
208                 $this->theme->title = $tribune->title;
209                 $this->theme->content = $tribune->edit();
210             }
211         }
212
213         $this->theme->sidebar = $this->site->sidebar();
214         $this->theme->topbar = $this->site->topbar();
215         print $this->theme->html();
216
217         return true;
218     }
219
220     public function tribune_create_anonymous($parts, $get, $post) {
221         $tribune = new Tribune();
222         if (!isset($_SESSION['anonymous_tribunes'])) {
223             $_SESSION['anonymous_tribunes'] = [];
224         }
225         $tribune->temporary = 1;
226         $tribune->created = time();
227         $tribune->generate_title();
228         $tribune->generate_color();
229         $tribune->generate_id();
230         if ($tribune->insert()) {
231             $_SESSION['anonymous_tribunes'][] = $tribune;
232         }
233
234         $this->theme->title = $tribune->title;
235         $this->theme->content = $tribune->edit();
236
237         $this->theme->sidebar = $this->site->sidebar();
238         $this->theme->topbar = $this->site->topbar();
239         print $this->theme->html();
240
241         return true;
242     }
243
244     public function legacy_tribune_show_anonymous_tsv($parts, $get, $post) {
245         $tribune = Tribune::load_from_title($parts[2]);
246
247         if (!$tribune) {
248             $tribune = new Tribune();
249             if (!isset($_SESSION['anonymous_tribunes'])) {
250                 $_SESSION['anonymous_tribunes'] = [];
251             }
252             $tribune->temporary = 1;
253             $tribune->created = time();
254             $tribune->title = $parts[2];
255             $tribune->generate_color();
256             $tribune->generate_id();
257             if ($tribune->insert()) {
258                 $_SESSION['anonymous_tribunes'][] = $tribune;
259             }
260         }
```

```
261
262             if (!$tribune->temporary or $tribune->deleted) {
263                 header('HTTP/1.0 404 Not Found');
264             } else {
265                 $board = new Board([$tribune], $tribune->url());
266                 $this->theme->content = $board->tsv();
267             }
268
269             header('Content-Type: text/tab-separated-values;charset=UTF-8');
270             print $this->theme->bare();
271
272             return true;
273         }
274
275         public function legacy_tribune_show_anonymous_xml($parts, $get, $post) {
276             $tribune = Tribune::load_from_title($parts[2]);
277
278             if (!$tribune) {
279                 $tribune = new Tribune();
280                 if (!isset($_SESSION['anonymous_tribunes'])) {
281                     $_SESSION['anonymous_tribunes'] = [];
282                 }
283                 $tribune->temporary = 1;
284                 $tribune->created = time();
285                 $tribune->title = $parts[2];
286                 $tribune->generate_color();
287                 $tribune->generate_id();
288                 if ($tribune->insert()) {
289                     $_SESSION['anonymous_tribunes'][] = $tribune;
290                 }
291             }
292
293             if (!$tribune->temporary or $tribune->deleted) {
294                 header('HTTP/1.0 404 Not Found');
295             } else {
296                 $board = new Board([$tribune], $tribune->url());
297                 $this->theme->content = $board->xml();
298             }
299
300             header('Content-Type: text/tab-separated-values;charset=UTF-8');
301             print $this->theme->bare();
302
303             return true;
304         }
305
306         public function legacy_tribune_show_anonymous_post($parts, $get, $post) {
307             if (!$tribune = Tribune::load_from_title($parts[2])) {
308                 header('HTTP/1.0 404 Not Found');
309             }
310
311             $board = new Board([$tribune], $tribune->url());
312
313             if ($new_post = $tribune->post($post, $this->site) and $this->json) {
314                 $this->theme->content = $board->json([$new_post]);
315             } else if ($this->json) {
316                 $this->theme->content = $board->json();
317             } else {
```

```
318                 $this->theme->title = $tribune->title;
319                 $this->theme->content = $board->show();
320             }
321 
322             if ($this->json) {
323                 header('Content-Type: application/json;charset=UTF-8');
324                 print $this->theme->bare();
325             } else {
326                 $this->theme->sidebar = $this->site->sidebar();
327                 $this->theme->topbar = $this->site->topbar();
328                 print $this->theme->html();
329             }
330 
331             return true;
332         }
333 
334         public function legacy_tribune_show_anonymous($parts, $get, $post) {
335             if (!$tribune = Tribune::load_from_title($parts[2])) {
336                 $tribune = new Tribune();
337                 if (!isset($_SESSION['anonymous_tribunes'])) {
338                     $_SESSION['anonymous_tribunes'] = [];
339                 }
340                 $tribune->temporary = 1;
341                 $tribune->created = time();
342                 $tribune->title = $parts[2];
343                 $tribune->generate_color();
344                 $tribune->generate_id();
345                 if ($tribune->insert()) {
346                     $_SESSION['anonymous_tribunes'][] = $tribune;
347                 }
348             }
349 
350             $board = new Board([$tribune], $tribune->url());
351 
352             if ($new_post = $tribune->post($post, $this->site) and $this->json) {
353                 $this->theme->content = $board->json([$new_post]);
354             } else if ($this->json) {
355                 $this->theme->content = $board->json();
356             } else {
357                 $this->theme->title = $tribune->title;
358                 $this->theme->content = $board->show();
359             }
360 
361             if ($this->json) {
362                 header('Content-Type: application/json;charset=UTF-8');
363                 print $this->theme->bare();
364             } else {
365                 $this->theme->sidebar = $this->site->sidebar();
366                 $this->theme->topbar = $this->site->topbar();
367                 print $this->theme->html();
368             }
369 
370             return true;
371         }
372 
373         public function tribune_show_anonymous($parts, $get, $post, $matches) {
374             if (!$tribune = Tribune::load_from_url($matches[1]) or !$tribune->temporary or $tribune-
```

```
375                 header('HTTP/1.0 404 Not Found');
376             } else {
377                 $board = new Board([$tribune], $tribune->url());
378
379                 if (!empty($post['message']) and $new_post = $tribune->post($post, $this->site)) {
380                     header("X-Post-Id: {$new_post->post_id}");
381
382                     if ($this->json) {
383                         $this->theme->content = $board->json([$new_post]);
384                     }
385                 } else if ($this->json) {
386                     $this->theme->content = $board->json();
387                 } else {
388                     $this->theme->title = $tribune->title;
389                     $this->theme->content = $board->show();
390                 }
391             }
392
393             if ($this->json) {
394                 header('Content-Type: application/json;charset=UTF-8');
395                 print $this->theme->bare();
396             } else {
397                 $this->theme->sidebar = $this->site->sidebar();
398                 $this->theme->topbar = $this->site->topbar();
399                 print $this->theme->html();
400             }
401
402             return true;
403         }
404
405         public function tribune_show_anonymous_tsv($parts, $get, $post, $matches) {
406             $tribune = Tribune::load_from_url($matches[1]);
407
408             if (!$tribune or !$tribune->temporary or $tribune->deleted) {
409                 header('HTTP/1.0 404 Not Found');
410             } else {
411                 $board = new Board([$tribune], $tribune->url());
412                 $this->theme->content = $board->tsv();
413             }
414
415             header('Content-Type: text/tab-separated-values;charset=UTF-8');
416             print $this->theme->bare();
417
418             return true;
419         }
420
421         public function tribune_show_anonymous_xml($parts, $get, $post, $matches) {
422             $tribune = Tribune::load_from_url($matches[1]);
423
424             if (!$tribune or !$tribune->temporary or $tribune->deleted) {
425                 header('HTTP/1.0 404 Not Found');
426             } else {
427                 $board = new Board([$tribune], $tribune->url());
428                 $this->theme->content = $board->xml();
429             }
430
431             header('Content-Type: text/tab-separated-values;charset=UTF-8');
```

```
432             print $this->theme->bare();
433
434             return true;
435         }
436
437         public function tribune_edit($parts, $get, $post) {
438             $user = User::load_from_url($parts[1]);
439             $tribune = Tribune::load_from_url($parts[2]);
440
441             if ($user && $tribune && $user->id === $tribune->user_id) {
442                 if (!empty($post) && $tribune->fill($post)) {
443                     if ($tribune->id) {
444                         $tribune->update();
445                     } else {
446                         $tribune->insert();
447                     }
448                 }
449
450                 $this->theme->title = $tribune->title;
451                 $this->theme->content = $tribune->edit();
452             }
453
454             $this->theme->sidebar = $this->site->sidebar();
455             $this->theme->topbar = $this->site->topbar();
456             print $this->theme->html();
457
458             return true;
459         }
460
461         public function tribune_show($parts, $get, $post) {
462             $user = User::load_from_url($parts[1]);
463             $tribune = Tribune::load_from_url($parts[2]);
464
465             if ($user && $tribune && $user->id === $tribune->user_id) {
466                 $board = new Board([$tribune], $tribune->url());
467
468                 if (!empty($post['message']) and $new_post = $tribune->post($post, $this->site)) {
469                     header("X-Post-Id: {$new_post->post_id}");
470
471                     if ($this->json) {
472                         $this->theme->content = $board->json([$new_post]);
473                     }
474                 } else if ($this->json) {
475                     $this->theme->content = $board->json();
476                 } else {
477                     $this->theme->title = $tribune->title;
478                     $this->theme->content = $board->show();
479                 }
480             }
481
482             if ($this->json) {
483                 header('Content-Type: application/json;charset=UTF-8');
484                 print $this->theme->bare();
485             } else {
486                 $this->theme->sidebar = $this->site->sidebar();
487                 $this->theme->topbar = $this->site->topbar();
488                 print $this->theme->html();
```

```
489             }
490
491             return true;
492         }
493
494         public function tribune_show_tsv($parts, $get, $post) {
495             $user = User::load_from_url($parts[1]);
496             $tribune = Tribune::load_from_url($parts[2]);
497
498             if ($user && $tribune && $user->id === $tribune->user_id) {
499                 $board = new Board([$tribune], $tribune->url());
500                 $this->theme->content = $board->tsv();
501             }
502
503             header('Content-Type: text/tab-separated-values;charset=UTF-8');
504             print $this->theme->bare();
505
506             return true;
507         }
508
509         public function user_show($parts, $get, $post) {
510             $user = User::load_from_url($parts[1]);
511
512             if ($user) {
513                 $tribunes = Tribune::select(['user_id' => $user->id]);
514                 $board = new Board($tribunes, $user->home_url());
515
516                 if ($this->json) {
517                     $this->theme->content = $board->json();
518                 } else {
519                     $this->theme->title = $tribune->title;
520                     $this->theme->content = $board->show();
521                 }
522             }
523
524             if ($this->json) {
525                 header('Content-Type: application/json;charset=UTF-8');
526                 print $this->theme->bare();
527             } else {
528                 $this->theme->sidebar = $this->site->sidebar();
529                 $this->theme->topbar = $this->site->topbar();
530                 print $this->theme->html();
531             }
532
533             return true;
534         }
535     }
```