



summary refs log tree commit diff stats

log msg

search

path: root/extra/tribune_weather/tribune_weather.module

blob: 55fcc6ff23847b80e81c7bfeab21014af05e76c2 (plain)

```
1 <?php
2
3 /**
4  * Implements hook_tribune_after_post().
5  */
6 function tribune_weather_tribune_after_post($node, $post) {
7   $enabled = variable_get('tribune_weather_enabled_' . $node->nid, FALSE);
8
9   if ($enabled) {
10    $message = $post['text'];
11    if (strpos($message, t('/weather')) === 0) {
12     $place = trim(str_replace(t('/weather'), '', $message));
13
14     $units = variable_get('tribune_weather_units_' . $node->nid, 'metric');
15
16     $weather = tribune_weather_get_condition($place, $units);
17
18     $clock = tribune_format_time($node, $post['timestamp']);
19     tribune_post($node, $clock . ' ' . $weather, user_load(0), 'Meteo');
20   } elseif (strpos($message, t('/forecast')) === 0) {
21     $place = trim(str_replace(t('/forecast'), '', $message));
22
23     $units = variable_get('tribune_weather_units_' . $node->nid, 'metric');
24
25     $weather = tribune_weather_get_forecast($place, $units);
26
27     $clock = tribune_format_time($node, $post['timestamp']);
28     tribune_post($node, $clock . ' ' . $weather, user_load(0), 'Meteo');
29   }
30 }
31
32
33 function tribune_weather_get_condition($place, $units) {
34   $url = 'http://api.openweathermap.org/data/2.1/find/name?q=' . urlencode($place);
35
36   switch ($units) {
37     case 'imperial':
38       $url .= '&units=imperial';
39       $degree_unit = t('%F');
40       $wind_unit = t('mph');
41       break;
42     case 'si':
43       $url .= '&units=si';
44       $degree_unit = t('%K');
45       $wind_unit = t('km/h');
46       break;
47     case 'metric':
48     default:
49       $url .= '&units=metric';
50       $degree_unit = t('%C');
51       $wind_unit = t('km/h');
52   }
53
54   $result = drupal_http_request($url);
55
56   if (isset($result->error) or !($data = drupal_json_decode($result->data)) or !$data['count']) {
57     return t('Unable to retrieve data for !place', array('!place' => $place));
58   } else {
59     foreach ($data['list'] as $city) {
60       $name = $city['name'];
61       $temp = round($city['main']['temp']);
62       $humidity = round($city['main']['humidity']);
63       $wind = round($city['wind']['speed']);
64       $condition = $city['weather'][0]['main'];
65
66       return t('Weather in <b>!city</b>: !condition, !degrees!degreeunit, !humidity% humidity, !wind !windunit', array(
67         '!city' => $name,
68         '!condition' => $condition,
69         '!degrees' => $temp,
70         '!degreeunit' => $degree_unit,
71         '!humidity' => $humidity,
72         '!wind' => $wind,
73         '!windunit' => $wind_unit,
74       ));
75     }
76   }
77 }
78
79 function tribune_weather_get_forecast($place, $units) {
80   $code_url = 'http://api.openweathermap.org/data/2.1/find/name?q=' . urlencode($place);
81   $url = 'http://api.openweathermap.org/data/2.2/forecast/city/%code?mode=daily_compact';
82
83   switch ($units) {
84     case 'imperial':
85       $url .= '&units=imperial';
86       $degree_unit = t('%F');
87       $wind_unit = t('mph');
88       $date_format = 'm/d';
89       break;
90     case 'si':
91       $url .= '&units=si';
92       $degree_unit = t('%K');
93       $wind_unit = t('km/h');
94       $date_format = 'Y-m-d';
95       break;
96     case 'metric':
97     default:
98       $url .= '&units=metric';
99       $degree_unit = t('%C');
100      $wind_unit = t('km/h');
101      $date_format = 'd/m';
102   }
103
104   $result = drupal_http_request($code_url);
105
106   if (isset($result->error) or !($data = drupal_json_decode($result->data)) or !$data['count']) {
107     return t('Unable to retrieve data for !place', array('!place' => $place));
108   } else {
109     $code = $data['list'][0]['id'];
110     $result = drupal_http_request(str_replace('%code', $code, $url));
111     $data = drupal_json_decode($result->data);
112
113     $name = $data['city']['name'];
114     $string = t('Forecast for <b>!city</b> - ', array('!city' => $name));
115     $days = array();
116     foreach ($data['list'] as $point) {
117       $date = date($date_format, $point['dt']);
118
119       if (isset($days[$date]) or count($days) > 4) {
120         continue;
121       }
122
123       $temp_min = round($point['morn']);
124       $temp_max = round($point['temp']);
125       $humidity = round($point['main']['humidity']);
126       $wind = round($point['wind']['speed']);
127       $condition = $point['weather'][0]['description'];
128
129       $days[$date] = t('!date: !condition !degrees_min»!degrees_max!degreeunit', array(
130         '!date' => $date,
131         '!condition' => $condition,
132         '!degrees_min' => $temp_min,
133         '!degrees_max' => $temp_max,
134         '!degreeunit' => $degree_unit,
135       ));
136     }
137   }
138
139   $string .= join(' ', $days);
140
141   return $string;
142 }
143
144 /**
145  * Implements hook_tribune_form().
146  */
147 function tribune_weather_tribune_form($node, $form_state) {
148   $default_value = 'NULL';
149   $default_value_units = 'metric';
150
151   if (!empty($node->nid)) {
152     $default_value = variable_get('tribune_weather_enabled_' . $node->nid, FALSE);
153     $default_value_units = variable_get('tribune_weather_units_' . $node->nid, 'metric');
154   }
155
156   return array(
157     'tribune_weather_details' => array(
158       '#weight' => 10,
159       '#type' => 'fieldset',
160       '#title' => t('Weather'),
161       '#description' => t('Weather condition and forecast is taken from http://openweathermap.org.'),
162
163     'tribune_weather_enabled' => array(
164       '#type' => 'checkbox',
165       '#title' => t('Enable weather commands'),
166       '#default_value' => $default_value,
167       '#description' => t('Weather condition for a place will be available by typing <code>weather &lt;city>&lt;/code>, forecast with <code>/forecast &lt;city>&lt;/code>.'),
168     ),
169     'tribune_weather_units' => array(
170       '#type' => 'select',
171       '#title' => t('Unit system use'),
172       '#default_value' => $default_value_units,
173       '#options' => array(
174         'metric' => t('Metric'),
175         'imperial' => t('Imperial'),
176         'si' => t('SI'),
177       ),
178     ),
179   ),
180 );
181 }
182
183 /**
184  * Implements hook_node_update().
185  */
186 function tribune_weather_node_update($node) {
187   if ($node->type == 'tribune' and isset($node->tribune_weather_enabled)) {
188     variable_set('tribune_weather_enabled_' . $node->nid, $node->tribune_weather_enabled);
189     variable_set('tribune_weather_units_' . $node->nid, $node->tribune_weather_units);
190   }
191 }
192
193 /**
194  * Implements hook_node_insert().
195  */
196 function tribune_weather_node_insert($node) {
197   if ($node->type == 'tribune' and isset($node->tribune_weather_enabled)) {
198     variable_set('tribune_weather_enabled_' . $node->nid, $node->tribune_weather_enabled);
199     variable_set('tribune_weather_units_' . $node->nid, $node->tribune_weather_units);
200   }
201 }
202
203 /**
204  * Implements hook_node_delete().
205  */
206 function tribune_weather_node_delete($node) {
207   if ($node->type == 'tribune') {
208     variable_del('tribune_weather_enabled_' . $node->nid);
209     variable_del('tribune_weather_units_' . $node->nid);
210   }
211 }
212 }
```