
```
1. <html>
2. <head>
3.   <title>403 Forbidden</title>
4. </head>
5. <body>
6.
7. <p>Directory access is forbidden.</p>
8.
9. </body>
10. </html>
```

```
1. <html>
2. <head>
3.     <title>403 Forbidden</title>
4. </head>
5. <body>
6.
7. <p>Directory access is forbidden.</p>
8.
9. </body>
10. </html>
```

```
1. <?php
2.
3. class Tweets extends CI_Controller {
4.     public function __construct() {
5.         parent::__construct();
6.         $this->load->model('Tag');
7.         $this->load->model('Tweet');
8.     }
9.
10.    /**
11.     * Add a new tweet
12.     *
13.     */
14.    public function add() {
15.        // add new tweet
16.        $tweet = (object)$this->input->post();
17.        $result = $this->Tweet->add($tweet);
18.
19.        // make sure tweet was successfully added
20.        if (!$result) {
21.            echo json_encode(array(
22.                'success' => false,
23.            ));
24.            exit;
25.        }
26.
27.        // associate each tag with tweet
28.        foreach ($tweet->tags as $tag) {
29.            if ($tag) {
30.                $this->Tag->add(array(
31.                    'content' => $tag,
32.                    'tweet_id' => $result
33.                ));
34.            }
35.        }
36.
37.        echo json_encode(array(
38.            'success' => true,
39.            'id' => $result
40.        ));
41.    }
42.
43.    /**
44.     * TwitterNiter home page
45.     * /tweets
46.     *
47.     */
48.    public function index() {
```

```
49.     $tweets = $this->Tweet->get_newsfeed();
50.
51.     $this->load->view('tweets/index.php', array(
52.         'tweets' => $tweets
53.     ));
54. }
55.
56. /**
57.  * Search tweets by username or content
58.  * /tweets/search
59.  *
60.  */
61. public function search() {
62.     // get parameters from url
63.     $username = (isset($_GET['username'])) ? $_GET['username'] : '';
64.     $content = (isset($_GET['content'])) ? $_GET['content'] : '';
65.
66.     // search tweets
67.     $tweets = $this->Tweet->search($username, $content);
68.
69.     var_dump($tweets);
70.     exit;
71. }
72. }
```

```
1. <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2.
3. class Welcome extends CI_Controller {
4.
5.     /**
6.      * Index Page for this controller.
7.      *
8.      * Maps to the following URL
9.      *   http://example.com/index.php/welcome
10.     * - or -
11.     *   http://example.com/index.php/welcome/index
12.     * - or -
13.     * Since this controller is set as the default controller in
14.     * config/routes.php, it's displayed at http://example.com/
15.     *
16.     * So any other public methods not prefixed with an underscore will
17.     * map to /index.php/welcome/<method_name>
18.     * @see http://codeigniter.com/user_guide/general/urls.html
19.     */
20.     public function index()
21.     {
22.         $this->load->view('welcome_message');
23.     }
24. }
25.
26. /* End of file welcome.php */
27. /* Location: ./application/controllers/welcome.php */
```

```
1. <html>
2. <head>
3.     <title>403 Forbidden</title>
4. </head>
5. <body>
6.
7. <p>Directory access is forbidden.</p>
8.
9. </body>
10. </html>
```

```
1. <html>
2. <head>
3.     <title>403 Forbidden</title>
4. </head>
5. <body>
6.
7. <p>Directory access is forbidden.</p>
8.
9. </body>
10. </html>
```

```
1. <?php
2.
3. class Tag extends CI_Model {
4.     /**
5.      * Add a new tag
6.      * @param $tag Object with tweet_id and content
7.      *
8.      */
9.     public function add($tag) {
10.         if ($this->db->insert('tags', $tag))
11.             return $this->db->insert_id();
12.         else
13.             return false;
14.     }
15.
16.     /**
17.      * Get the tags associated with tweets
18.      * @param $tweet_ids Array of IDs of tweets to get tags for
19.      * @return Array of tags indexed by tweet_id
20.      *
21.      */
22.     public function get_for_tweets($tweet_ids) {
23.         // avoid sql error when given empty array
24.         if (empty($tweet_ids))
25.             return array();
26.
27.         // get tags from database
28.         $tags = $this->db->where_in('tweet_id', $tweet_ids)->get('tags')->result();
29.
30.         // group tags by tweet id
31.         $result = array();
32.         foreach ($tags as $tag)
33.             $result[$tag->tweet_id][] = $tag->content;
34.
35.         return $result;
36.     }
37. }
```



```
1. <?php
2.
3. class Tweet extends CI_Model {
4.     public function __construct() {
5.         parent::__construct();
6.         $this->load->model('Tag');
7.     }
8.
9.     /**
10.      * Add a new tweet
11.      * @param $tweet Object with contents and username
12.      *
13.      */
14.     public function add($tweet) {
15.         if ($this->db->insert('tweets', $tweet))
16.             return $this->db->insert_id();
17.         else
18.             return false;
19.     }
20.
21.     /**
22.      * Get the last 20 tweets to be posted
23.      *
24.      */
25.     public function get_newsfeed() {
26.         // get tweets
27.         $tweets = $this->db->limit(20)->order_by('timestamp desc')->get('tweets')->result();
28.
29.         // get tags associated with tweets
30.         $tweet_ids = array_map(function ($e) { return $e->id; }, $tweets);
31.         $tags = $this->Tag->get_for_tweets($tweet_ids);
32.
33.         // associate tags with tweets
34.         foreach ($tweets as &$tweet)
35.             $tweet->tags = (isset($tags[$tweet->id])) ? $tags[$tweet->id] : array();
36.
37.         return $tweets;
38.     }
39.
40.     /**
41.      * Search tweets by username and content
42.      * @param $username Username to search by
43.      * @param $content Content to search for
44.      *
45.      */
46.     public function search($username, $content) {
47.         if ($username)
48.             $this->db->or_like('username', $username);
```

```
49.         if ($content)
50.             $this->db->or_like('content', $content);
51.
52.         return $this->db->get('tweets')->result();
53.     }
54. }
```

```
1. <html>
2. <head>
3.     <title>403 Forbidden</title>
4. </head>
5. <body>
6.
7. <p>Directory access is forbidden.</p>
8.
9. </body>
10. </html>
```

```
1. <!doctype html>
2. <html>
3.     <head>
4.         <title>TwitterNiter - The Hippest App There Ever Was</title>
5.         <meta name="viewport" content="width=device-width, initial-scale=1">
6.
7.         <link rel="stylesheet" href="/css/jquery.mobile-1.0.1.min.css" />
8.
9.         <script src="/js/lib/jquery-1.7.1.min.js"></script>
10.        <script src="/js/lib/jquery.mobile-1.0.1.min.js"></script>
11.        <script src="/js/tweets/index.js"></script>
12.    </head>
13.    <body>
14.        <div data-role="page">
15.            <div data-role="header">
16.                <h1>TwitterNiter</h1>
17.            </div>
18.            <div data-role="content">
19.                <form method="post" action="/tweets/add">
20.                    <input type="text" id="txt-username" name="username" placeholder="Username" />
21.                    <br />
22.                    <textarea id="txt-content" name="content" placeholder="Tweety tweet tweet"></textarea>
23.                    <br />
24.                    <input type="text" id="txt-tag1" name="tag1" placeholder="Give your tweet a tag" />
25.                    <input type="text" id="txt-tag2" name="tag2" placeholder="Need another tag?" />
26.                    <input id="btn-submit" type="submit" value="Tweet!" />
27.                </form>
28.
29.                <ul id="tweets" data-role="listview" style="margin-top: 40px">
30.                    <?php foreach ($tweets as $tweet): ?>
31.                        <li>
32.                            <h4><?= $tweet->username ?></h4>
33.                            <?= $tweet->content; ?><br />
34.                            <small><?= implode(' ', $tweet->tags); ?></small>
35.                        </li>
36.                    <?php endforeach; ?>
37.                </ul>
38.            </div>
39.        </body>
40. </html>
```

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="utf-8">
5.     <title>Welcome to CodeIgniter</title>
6.
7.     <style type="text/css">
8.
9.         ::selection{ background-color: #E13300; color: white; }
10.        ::moz-selection{ background-color: #E13300; color: white; }
11.        ::webkit-selection{ background-color: #E13300; color: white; }
12.
13.        body {
14.            background-color: #fff;
15.            margin: 40px;
16.            font: 13px/20px normal Helvetica, Arial, sans-serif;
17.            color: #4F5155;
18.        }
19.
20.        a {
21.            color: #003399;
22.            background-color: transparent;
23.            font-weight: normal;
24.        }
25.
26.        h1 {
27.            color: #444;
28.            background-color: transparent;
29.            border-bottom: 1px solid #D0D0D0;
30.            font-size: 19px;
31.            font-weight: normal;
32.            margin: 0 0 14px 0;
33.            padding: 14px 15px 10px 15px;
34.        }
35.
36.        code {
37.            font-family: Consolas, Monaco, Courier New, Courier, monospace;
38.            font-size: 12px;
39.            background-color: #f9f9f9;
40.            border: 1px solid #D0D0D0;
41.            color: #002166;
42.            display: block;
43.            margin: 14px 0 14px 0;
44.            padding: 12px 10px 12px 10px;
45.        }
46.
47.        #body{
48.            margin: 0 15px 0 15px;
```

```
49.     }
50.
51.     p.footer{
52.         text-align: right;
53.         font-size: 11px;
54.         border-top: 1px solid #D0D0D0;
55.         line-height: 32px;
56.         padding: 0 10px 0 10px;
57.         margin: 20px 0 0 0;
58.     }
59.
60.     #container{
61.         margin: 10px;
62.         border: 1px solid #D0D0D0;
63.         -webkit-box-shadow: 0 0 8px #D0D0D0;
64.     }
65. </style>
66. </head>
67. <body>
68.
69. <div id="container">
70.     <h1>Welcome to CodeIgniter!</h1>
71.
72.     <div id="body">
73.         <p>The page you are looking at is being generated dynamically by CodeIgniter.</p>
74.
75.         <p>If you would like to edit this page you'll find it located at:</p>
76.         <code>application/views/welcome_message.php</code>
77.
78.         <p>The corresponding controller for this page is found at:</p>
79.         <code>application/controllers/welcome.php</code>
80.
81.         <p>If you are exploring CodeIgniter for the very first time, you should start by reading the <a href="user_guide/">User Guide</a>.</p>
82.     </div>
83.
84.     <p class="footer">Page rendered in <strong>{elapsed_time}</strong> seconds</p>
85. </div>
86.
87. </body>
88. </html>
```

```
1. <?php
2.
3. /*
4. *-----
5. * APPLICATION ENVIRONMENT
6. *-----
7. *
8. * You can load different configurations depending on your
9. * current environment. Setting the environment also influences
10. * things like logging and error reporting.
11. *
12. * This can be set to anything, but default usage is:
13. *
14. *     development
15. *     testing
16. *     production
17. *
18. * NOTE: If you change these, also change the error_reporting() code below
19. *
20. */
21.     define('ENVIRONMENT', 'development');
22. /*
23. *-----
24. * ERROR REPORTING
25. *-----
26. *
27. * Different environments will require different levels of error reporting.
28. * By default development will show errors but testing and live will hide them.
29. */
30.
31. if (defined('ENVIRONMENT'))
32. {
33.     switch (ENVIRONMENT)
34.     {
35.         case 'development':
36.             error_reporting(E_ALL);
37.             break;
38.
39.         case 'testing':
40.         case 'production':
41.             error_reporting(0);
42.             break;
43.
44.         default:
45.             exit('The application environment is not set correctly.');
```

```
49. /*
50. *-----
51. * SYSTEM FOLDER NAME
52. *-----
53. *
54. * This variable must contain the name of your "system" folder.
55. * Include the path if the folder is not in the same directory
56. * as this file.
57. *
58. */
59.     $system_path = '../system';
60.
61. /*
62. *-----
63. * APPLICATION FOLDER NAME
64. *-----
65. *
66. * If you want this front controller to use a different "application"
67. * folder then the default one you can set its name here. The folder
68. * can also be renamed or relocated anywhere on your server. If
69. * you do, use a full server path. For more info please see the user guide:
70. * http://codeigniter.com/user\_guide/general/managing\_apps.html
71. *
72. * NO TRAILING SLASH!
73. *
74. */
75.     $application_folder = '../application';
76.
77. /*
78. * -----
79. * DEFAULT CONTROLLER
80. * -----
81. *
82. * Normally you will set your default controller in the routes.php file.
83. * You can, however, force a custom routing by hard-coding a
84. * specific controller class/function here. For most applications, you
85. * WILL NOT set your routing here, but it's an option for those
86. * special instances where you might want to override the standard
87. * routing in a specific front controller that shares a common CI installation.
88. *
89. * IMPORTANT: If you set the routing here, NO OTHER controller will be
90. * callable. In essence, this preference limits your application to ONE
91. * specific controller. Leave the function name blank if you need
92. * to call functions dynamically via the URI.
93. *
94. * Un-comment the $routing array below to use this feature
95. *
96. */
```



```
97. // The directory name, relative to the "controllers" folder. Leave blank
98. // if your controller is not in a sub-folder within the "controllers" folder
99. // $routing['directory'] = '';
100.
101. // The controller class file name. Example: Mycontroller
102. // $routing['controller'] = '';
103.
104. // The controller function you wish to be called.
105. // $routing['function'] = '';
106.
107.
108. /*
109. * -----
110. * CUSTOM CONFIG VALUES
111. * -----
112. *
113. * The $assign_to_config array below will be passed dynamically to the
114. * config class when initialized. This allows you to set custom config
115. * items or override any default config values found in the config.php file.
116. * This can be handy as it permits you to share one application between
117. * multiple front controller files, with each file containing different
118. * config values.
119. *
120. * Un-comment the $assign_to_config array below to use this feature
121. *
122. */
123. // $assign_to_config['name_of_config_item'] = 'value of config item';
124.
125.
126.
127. // -----
128. // END OF USER CONFIGURABLE SETTINGS. DO NOT EDIT BELOW THIS LINE
129. // -----
130.
131. /*
132. * -----
133. * Resolve the system path for increased reliability
134. * -----
135. */
136.
137. // Set the current directory correctly for CLI requests
138. if (defined('STDIN'))
139. {
140.     chdir(dirname(__FILE__));
141. }
142.
143. if (realpath($system_path) !== FALSE)
144. {
```

```
145.     $system_path = realpath($system_path).'/';
146. }
147.
148. // ensure there's a trailing slash
149. $system_path = rtrim($system_path, '/').'/';
150.
151. // Is the system path correct?
152. if ( ! is_dir($system_path) )
153. {
154.     exit("Your system folder path does not appear to be set correctly. Please open the following file and correct this: ".pathinfo(
155.     __FILE__, PATHINFO_BASENAME));
156. }
157. /*
158. * -----
159. * Now that we know the path, set the main path constants
160. * -----
161. */
162. // The name of THIS file
163. define('SELF', pathinfo(__FILE__, PATHINFO_BASENAME));
164.
165. // The PHP file extension
166. // this global constant is deprecated.
167. define('EXT', '.php');
168.
169. // Path to the system folder
170. define('BASEPATH', str_replace("\\", "/", $system_path));
171.
172. // Path to the front controller (this file)
173. define('FCPATH', str_replace(SELF, '', __FILE__));
174.
175. // Name of the "system folder"
176. define('SYSDIR', trim(strchr(trim(BASEPATH, '/'), '/'), '/'));
177.
178.
179. // The path to the "application" folder
180. if (is_dir($application_folder))
181. {
182.     define('APPPATH', $application_folder.'/');
183. }
184. else
185. {
186.     if ( ! is_dir(BASEPATH.$application_folder.'/') )
187.     {
188.         exit("Your application folder path does not appear to be set correctly. Please open the following file and correct this: ".SELF);
189.     }
190.
191.     define('APPPATH', BASEPATH.$application_folder.'/');
```

```
192.     }
193.
194. /*
195.  * -----
196.  * LOAD THE BOOTSTRAP FILE
197.  * -----
198.  *
199.  * And away we go...
200.  *
201. */
202. require_once BASEPATH.'core/CodeIgniter.php' ;
203.
204. /* End of file index.php */
205. /* Location: ./index.php */
```