

master ▾

...

[ICEcoder](#) / [classes](#) / **Settings.php** / [Jump to ▾](#)**mattpass** Dynamic assets path, tweak and fix in settings, ignore .idea[History](#)[1 contributor](#)

245 lines (221 sloc) | 8.43 KB

...

```
1  <?php declare(strict_types=1);
2
3  namespace ICEcoder;
4
5  class Settings
6  {
7      public function __construct()
8      {
9          // Set version number and document root as core settings
10         // Defaults to the right
11         $this->versionNo = "8.1";           // "8.1";
12         $this->docRoot = $_SERVER['DOCUMENT_ROOT']; // $_SERVER['DOCUMENT_ROOT']
13         $this->assetsRoot = "assets";       // "assets" (relative or absolute)
14     }
15
16     public function getCoreDetails()
17     {
18         return [
19             "versionNo" => $this->versionNo,
20             "docRoot" => $this->docRoot,
21             "assetsRoot" => $this->assetsRoot,
22         ];
23     }
24
25     // =====
26     // DATA DIR
27     // =====
28
29     public function getDataDirDetails()
```

```

30 {
31     clearstatcache();
32
33     // Return details about the data dir
34     $fullPath = dirname(__FILE__) . "../data/";
35     $exists = file_exists($fullPath);
36     $readable = is_readable($fullPath);
37     $writable = is_writable($fullPath);
38     return [
39         "fullPath" => $fullPath,
40         "exists" => $exists,
41         "readable" => $readable,
42         "writable" => $writable,
43     ];
44 }
45
46 // =====
47 // GLOBAL CONFIG
48 // =====
49
50 public function getConfigGlobalTemplate($asArray)
51 {
52     // Return the serialized global config template
53     $fileName = 'template-config-global.php';
54     $fullPath = dirname(__FILE__) . "../lib/" . $fileName;
55     if (function_exists('opcache_invalidate')) {
56         opcache_invalidate($fullPath, true);
57     }
58     $settings = file_get_contents($fullPath);
59     if ($asArray) {
60         $settings = $this->serializedFileData("get", $fullPath);
61     }
62     return $settings;
63 }
64
65 public function getConfigGlobalFileDetails()
66 {
67     clearstatcache();
68
69     // Return details about the global config file
70     $fileName = 'config-global.php';
71     $fullPath = dirname(__FILE__) . "../data/" . $fileName;
72     $exists = file_exists($fullPath);
73     $readable = is_readable($fullPath);
74     $writable = is_writable($fullPath);
75     $filemtime = filemtime($fullPath);
76     return [
77         "fileName" => $fileName,
78         "fullPath" => $fullPath,

```

```

79         "exists" => $exists,
80         "readable" => $readable,
81         "writable" => $writable,
82         "filemtime" => $filemtime
83     ];
84 }
85
86 public function getConfigGlobalSettings()
87 {
88     // Start an array with version number and document root
89     $settings = $this->getCoreDetails();
90     // Get global config file details
91     $fullPath = $this->getConfigGlobalFileDetails()['fullPath'];
92     $settingsFromFile = $this->serializedFileData("get", $fullPath);
93     // Merge that with the array we started with and return
94     $settings = array_merge($settings, $settingsFromFile);
95     return $settings;
96 }
97
98 public function setConfigGlobalSettings($settings): bool
99 {
100     // Get the global config file details
101     $fullPath = $this->getConfigGlobalFileDetails()['fullPath'];
102     if ($fConfigSettings = fopen($fullPath, 'w')) {
103         // If the settings we've received aren't in serialized format yet, do that now
104         // As $settings could be a serialized string or array
105         if (is_array($settings)) {
106             unset($settings['versionNo']);
107             unset($settings['docRoot']);
108         }
109         return $this->serializedFileData("set", $fullPath, $settings);
110     } else {
111         return false;
112     }
113 }
114
115 public function updateConfigGlobalSettings($array): bool
116 {
117     // Update global config settings file
118     $settingsFromFile = $this->getConfigGlobalSettings();
119     $settings = array_merge($settingsFromFile, $array);
120     return $this->setConfigGlobalSettings($settings);
121 }
122
123 // =====
124 // USERS CONFIG
125 // =====
126
127 public function getConfigUsersTemplate($asArray)

```

```

128     {
129         // Return the serialized users config template
130         $fileName = 'template-config-users.php';
131         $fullPath = dirname(__FILE__) . '/../lib/' . $fileName;
132         if (function_exists('opcache_invalidate')) {
133             opcache_invalidate($fullPath, true);
134         }
135         $settings = file_get_contents($fullPath);
136         if ($asArray) {
137             $settings = $this->serializedFileData("get", $fullPath);
138         }
139         return $settings;
140     }
141
142     public function getConfigUsersFileDetails($fileName)
143     {
144         // Return details about the users config file
145         $fullPath = dirname(__FILE__) . '/../data/' . $fileName;
146         $exists = file_exists($fullPath);
147         $readable = is_readable($fullPath);
148         $writable = is_writable($fullPath);
149         $filemtime = filemtime($fullPath);
150         return [
151             "fileName" => $fileName,
152             "fullPath" => $fullPath,
153             "exists" => $exists,
154             "readable" => $readable,
155             "writable" => $writable,
156             "filemtime" => $filemtime,
157         ];
158     }
159
160     public function getConfigUsersSettings($fileName)
161     {
162         // Get users config file details
163         $fullPath = $this->getConfigUsersFileDetails($fileName)['fullPath'];
164         $settingsFromFile = $this->serializedFileData("get", $fullPath);
165         // Now return
166         return $settingsFromFile;
167     }
168
169     public function setConfigUsersSettings($fileName, $settings): bool
170     {
171         // Get the users config file details
172         $fullPath = $this->getConfigUsersFileDetails($fileName)['fullPath'];
173         if ($fConfigSettings = fopen($fullPath, 'w')) {
174             return $this->serializedFileData("set", $fullPath, $settings);
175         } else {
176             return false;

```

```

177     }
178 }
179
180 public function updateConfigUsersSettings($fileName, $array): bool
181 {
182     // Update users config settings file
183     $settingsFromFile = $this->getConfigUsersSettings($fileName);
184     $settings = array_merge($settingsFromFile, $array);
185     return $this->setConfigUsersSettings($fileName, $settings);
186 }
187
188 public function updateConfigUsersCreateDate($fileName)
189 {
190     global $ICEcoderUserSettings;
191
192     // Get users config file details
193     $filemtime = $this->getConfigUsersFileDetails($fileName)['filemtime'];
194     // Make it a number (avoids null, undefined etc)
195     $filemtime = intval($filemtime);
196     // Set it to the epoch time now if we don't have a real value
197     if (0 === $filemtime) {
198         $filemtime = time();
199     }
200     // Update users config settings file
201     $ICEcoderSettingsFromFile = $this->getConfigUsersSettings($fileName);
202     $ICEcoderSettingsFromFile['configCreateDate'] = $filemtime;
203     $this->setConfigUsersSettings($fileName, $ICEcoderSettingsFromFile);
204     // Set the new value in array
205     $ICEcoderUserSettings['configCreateDate'] = $filemtime;
206 }
207
208 public function createIPSettingsFileIfNotExist()
209 {
210     global $username, $settingsFile, $settingsFileAddr;
211
212     // Create a duplicate version for the IP address of the domain if it doesn't exist yet
213     $serverAddr = $_SERVER['SERVER_ADDR'] ?? "1";
214     if ($serverAddr == "1" || $serverAddr == "::1") {
215         $serverAddr = "127.0.0.1";
216     }
217     $settingsFileAddr = 'config-' . $username . str_replace(".", "_", $serverAddr) . '.php';
218     if (false === file_exists(dirname(__FILE__) . '/../data/' . $settingsFileAddr)) {
219         if (false === copy(dirname(__FILE__) . '/../data/' . $settingsFile, dirname(__FILE__)
220             . $reqsFailures = ["phpCreateSettingsFileAddr"];
221             include dirname(__FILE__) . '/../lib/requirements.php";
222         }
223     }
224 }
225

```

```
226     public function serializedFileData($do, $fullPath, $output=null)
227     {
228         if ("get" === $do) {
229             if (function_exists('opcache_invalidate')) {
230                 opcache_invalidate($fullPath, true);
231             }
232             $data = file_get_contents($fullPath);
233             $data = str_replace("<\".\"?php\n/*\n\n", "", $data);
234             $data = str_replace("\n\n*/\n?\".\">", "", $data);
235             $data = unserialize($data);
236             return $data;
237         }
238         if ("set" === $do) {
239             if (true === is_array($output)) {
240                 $output = serialize($output);
241             }
242             return false !== file_put_contents($fullPath, "<\".\"?php\n/*\n\n" . $output . "\n\n*/\n\n");
243         }
244     }
245 }
```