

Entwicklerdokumentation Übungsplattform 2.0

Dokumentation UI

Felix Schmidt

Florian Lücke

Ralf Busch

31. März 2014

1 Templates

Die Seiten der Übungsplattform sind aus (eventuell mehreren) Templates aufgebaut. Ein Template ist ein php-Skript, das abhängig von den durch das Template dargestellten Daten vordefinierte Variablen kennt. Daten die an ein Template übergeben werden sind assoziative Arrays, die Schlüssel des Arrays sind im Template als Variablen verfügbar. Die folgenden listings zeigen ein JSON-codiertes assoziatives Array und ein mögliches Template zur Darstellung der Daten in dem Array.

Listing 1: Listings/template-json.json

```
1 {
2     "userName": "mustermann",
3     "firstName": "Max",
4     "lastName": "Mustermann",
5     "orderIDs":
6     [
7         1,
8         2,
9         3
10    ]
11 }
```

Listing 2: Listings/template-html.html

```
1 <html>
2     <head>
3         <title>Person</title>
4     </head>
5     <body>
6         Bestellungen von <?= $firstName ?> <?= $lastName ?> (<?= $userName ?>):
7         <br />
8         <?php if (count($orderIDs) != 0): ?>
9         <ol>
10             <?php foreach ($orderIDs as $id): ?>
11                 <li> <?= $id ?> </li>
12             <?php endforeach; ?>
13         </ol>
14         <?php else: ?>
15             Keine Bestellungen Vorhanden
16         <?php endif; ?>
17     </body>
18 </html>
```

Mit dem Code in folgendem Listing könnte schon eine Seite dargestellt werden

Listing 3: Listings/template-usage.php

```
1 <?php
2 include_once 'Template.php';
3
4 $data = file_get_contents('Listings/template-json.json');
5 $data = json_decode($data);
6
7 $t = Template::WithTemplateFile('Listings/template-html.html');
8 $t->bind($data);
9
10 print $t;
11
12 ?>
```

2 Die Klasse HTMLWrapper

Damit nicht jedes Template sein eigenes HTML Grundgerüst bieten muss kann die Klasse HTMLWrapper mehrere Elemente die auf einer Seite angezeigt werden sollen in ein HTML Gerüst einbetten. Außerdem bietet sie die Möglichkeit Elemente in Formulare einzubetten.

Es ist möglich Scripte und Stylesheets die eine Seite beeinhalten soll in einer Konfigurationsdatei einzutragen und diese automatisch einbinden lassen. Eine solche Konfigurationsdatei ist in folgendem Listing zu sehen.

Listing 4: Listings/HTMLWrapper-config.json

```
1 {
2     "title": "Übungsplattform",
3     "content": "text/html",
4     "charset": "utf-8",
5     "stylesheets":
6     [
7         "css/main.css"
8     ],
9     "javascripts":
10    [
11        "javascript/main.js"
12    ]
13 }
```

Die Benutzung der Klasse ist wie folgt:

Listing 5: Listings/HTMLWrapper-template1.html

```
1 <h1>Hello</h1>
```

Listing 6: Listings/HTMLWrapper-template2.html

```
1 <input name="greeting" type="text" value="World" />
```

Listing 7: Listings/HTMLWrapper-usage.php

```
1 <?php
2 include_once 'HTMLWrapper.php';
3 include_once 'Template.php';
4
5 // fill template
6 $t1 = Template::WithTemplateFile('Listings/HTMLWrapper-template1.html');
7 $t2 = Template::WithTemplateFile('Listings/HTMLWrapper-template2.html');
8
9 // embed template in default HTML framework
```

```
10 $w = new HTMLWrapper($t1, $t2);
11 $w->set_config_file('Listings/HTMLWrapper-config.json');
12 $w->defineForm(basename(__FILE__), false, $t2);
13 $w->show();
14
15 ?>
```

Dieser Code erzeugt folgende Ausgabe:

Listing 8: Listings/HTMLWrapper-result.html

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4     <meta http-equiv="content-type" content="text/html; charset=utf-8">
5     <link rel="stylesheet" type="text/css" href="css/main.css">
6     <script src="javascript/main.js"></script>
7     <title>Übungsplattform</title>
8 </head>
9 <body>
10     <div id="body-wrapper" class="body-wrapper">
11         <h1>Hello</h1>
12         <div id="content-wrapper" class="content-wrapper">
13             <form action="HTMLWrapper-usage.php" method="POST">
14                 <input name="greeting" type="text" value="World" />
15             </form>
16         </div> <!-- end: content-element -->
17     </div>
18 </body>
19 </html>
```

3 Anlegen einer neuen Seite

Um das Anlegen einer neuen Seite weiter zu vereinfachen fasst die Datei “Boilerplate.php” häufige `include` Anweisungen zusammen. Die Datei führt außerdem eine Sessionprüfung durch und macht die id des Benutzers, die id der zur Seite gehörigen Serie und die id des zur Seite gehörigen Kurses als Variablen verfügbar. Die Namen dieser Variablen lauten `$uid`, `$sid` und `$cid`.

Eine neue Seite wird angelegt, indem wie in ein neues PHP Script angelegt wird. Passwortgeschützte Dateien sollten “Boilerplate.php” importieren. Das Script sollte dann die für die Anzeige benötigten Daten laden und wie in Abschnitt 2 mit einem Template darstellen. Das Erstellen von Templates ist in Abschnitt 1 erklärt.

4 Helpers.php

In der Datei `Helper.php`, die durch `Boilerplate.php` in alle Dateien importiert wird, werden einige häufiger verwendete, von den Dateien geteilte Funktionen, wie etwa das erstellen eines Navigationselementes definiert. Die Datei definiert auch die Funktionen `http_get`, `http_delete`, `http_put_data` und `http_post_data` diese Funktionen sind auch in der PHP Erweiterung “`pecl_http`” definiert und sollten durch diese ersetzt werden, was Aufgrund vorhandener Spezialfunktionalität bisher nicht geschehen ist.