



YWEE

Personal Tutoring Service
Gruppe 2

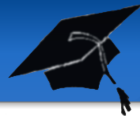
Weiß	Daniel Tatzel	Strobl	Schwindl
Laufenböck	Birnthaler	Götz	Schröter

02. Juli 2014



Agenda

- 1 Einleitung
- 2 SRA
- 3 SAD
- 4 Implementierung



Agenda

1 Einleitung

- Zuständigkeiten
- Implementierungen

2 SRA

3 SAD

4 Implementierung



Die Zuständigkeiten im Projekt:

- Stellvertretender Projektleiter
- Leitung des Teams Backend bestehend aus
 - Florian Laufenböck – Adminverwaltung und Profil
 - Tobias Schwindl – Kreditkarte, News, Suche
- Verwaltung des Github Repository
- Code Review



Die Implementierungen:

- Datenbank
- phpmyadmin
- Inhalte
- Besucherzähler
- Login
- Registrierung
- Gästebuch
- Privater Ordner
- Sprachauswahl
- 404 Error
- URL Rewriting
- ...



Agenda

- 1 Einleitung
- 2 SRA
- 3 SAD
- 4 Implementierung



Implementierung eines „Privaten Ordners“ der:

- Passwortgeschützt ist
- Unabhängig von der Datenbank ist
- Datei-Uploads erlaubt

Aus den Anforderungen entstand der Punkt:

F27: Privater Ordner

Der Private Ordner für Datei-Uploads ist über gesonderte Zugangsdaten erreichbar, die unabhängig von der Datenbank sind.



Agenda

- 1 Einleitung
- 2 SRA
- 3 SAD**
- 4 Implementierung



Um einen besseren Überblick zu Erhalten wurden die folgenden Fragen gestellt:

- 1 Welche Grundanforderungen sind nötig?
- 2 Welcher Webserver steht zur Verfügung?
- 3 Welche Möglichkeiten gibt es zur Umsetzung?
- 4 Welche Sprachen werden benötigt?



Grundanforderungen

Um den „Privaten Ordner“ effektiv nutzen zu können sind folgende Grundmerkmale nötig:

- Man muss das Passwort ändern können
- Man muss Dateien auswählen, hochladen und löschen können
- Die Dateien müssen optisch angemessen aufgelistet werden



1 Webserver

- Auf dem Zielsystem ist ein **Apache** Webserver in der Version 2 vorinstalliert.

2 Bietet der Webserver Lösungen für das Problem?

- **htaccess**: Unterstützt die Zugriffskontrolle per Passwortschutz



Zur Umsetzung standen zwei Möglichkeiten zur Verfügung:

- 1 Verzeichnisschutz mittels htaccess Dateien
- 2 Eigene Implementierung

Da eine geeignete Implementierung vorhanden ist und der Aufwand für eine eigene zu hoch ist, wurde sich für die Umsetzung mittels htaccess Dateien entschieden.



- Passwort ändern
 - HTML5 für die Formulare
 - PHP zum aktualisieren des Passworts
- Dateien auswählen, hochladen und löschen
 - HTML5 zum Auswählen
 - PHP zum Hochladen und zum Löschen



- Darstellung der Dateien
 - HTML5 für die Definition der Tabelle
 - PHP zum Ermitteln der Dateien
 - JQuery zur Generierung der Tabelle
- Sonstiges
 - Javascript für das Design



Agenda



1 Einleitung

2 SRA

3 SAD

4 Implementierung

- Code

- Demonstration



Passwort ändern

```
1 <?php
2 /* Passwortaenderung fuer den Privaten Ordner -- Daniel Tatzel */
3
4 /* create_htpasswd: Generiert eine .htpasswd Datei mit Benutzernamen und Passwort
   fuer den Login zum Privaten Ordner */
5 $DIR = $_SERVER["DOCUMENT_ROOT"] . "/test_02/private/";
6 if ( file_exists($DIR . '.htpasswd') )
7     unlink($DIR.'.htpasswd');
8
9 $passwd = crypt($_POST['passwd']);
10 $htpasswd = 'admin:' . $passwd . "\n";
11
12 $stream = "# Autor: Daniel Tatzel\n\n".$htpasswd;
13
14 $handle = fopen($DIR . '.htpasswd', "a");
15 fputs($handle, $stream);
16 fclose($handle);
17
18 header("Location: http://www.ebenezer-kunatse.net/private/");
19 ?>
```




Löschen einer Datei

```
1 <?php
2 // Autor von delete.php: Daniel Tatzel
3 // Loescht eine Datei
4
5 $path = realpath('files/' . $_POST['file']);
6
7 // Wenn vorhanden, dann l&oeschen
8 if( is_file($path) )
9     unlink( $path );
10 ?>
```



Verzeichnis Inhalt holen

```
1 <?php
2 // Autor von ListDir.php: Daniel Tatzel
3 // Liest alle Dateien aus, berechnet die Groesse in KB und erzeugt ein JSON
  objekt
4
5 $Dir = "../files/"; // Zielordner
6
7 // Kann dann mit JSON in Javascript umgewandelt werden zur Verarbeitung (
  Array )
8 $FileListing = array_diff(scandir($Dir), array('..', '.'));
9
10 // Wird nur ausgefuehrt falls Daten vorhanden sind
11 if ( count( $FileListing ) > 0 )
12 {
13     $file_arr = array();
14
15     foreach ( $FileListing as $File )
16     {
17         $size = number_format( filesize($Dir.$File)/1024, 2, ",", "." );
18
19         // Array fuer JSON Objekt
20         $file_arr[] = array("name" => $File, "size" => $size);
21     }
22
23     echo json_encode( $file_arr ); // JSON Ausgabe
24 }
25 ?>
```



Datei hochladen

```
20 <?php
21 // Autor von upload.php: Daniel Tatzel
22 // Laedt eine ausgewaehlte Datei auf den Server hoch
23
24 $temp = explode(".", $_FILES["file"]["name"]);
25 $extension = end($temp);
26
27 // Gibt eine Meldung aus, falls ein Fehler aufgetreten ist
28 if ($_FILES["file"]["error"] > 0)
29 {
30     echo "Fehler Code: " . $_FILES["file"]["error"] . "<br>";
31 }
32 else
33 {
34     // Prueft ob Datei schon vorhanden ist und gibt eine Meldung aus,
35     // ansonsten wird die Datei hochgeladen
36     if (file_exists($_SERVER["DOCUMENT_ROOT"] . "/test_02/private/files/" .
37         $_FILES["file"]["name"])) {
38         echo $_FILES["file"]["name"] . " bereits vorhanden. ";
39     }
40     else {
41         move_uploaded_file($_FILES["file"]["tmp_name"],
42             $_SERVER["DOCUMENT_ROOT"] . "/test_02/private/files/" . $_FILES["file"]
43             ["name"]);
44     }
45 }
```



Ausblenden von Bereichen

```
1 <script language="javascript">
2   function show(selector)
3   {
4       var sel = selector;
5
6       if ( sel == 0 )
7           sel = "passwd";
8       else
9           sel = "file"
10
11       if(document.getElementById( sel ).style.display == 'block')
12       {
13           document.getElementById( sel ).style.display = 'none';
14       }
15       else
16       {
17           document.getElementById( sel ).style.display = 'block';
18       }
19   }
20 </script>
```



Erstellen der Tabelle

```
1 $(document).ready(function() {
2     var mTable = $('#filesTable').dataTable({
3         "oLanguage": {
4             "sSearch": "Dateien filtern:",
5             "oPaginate": {
6                 "sNext": "Naechste Seite",
7                 "sPrevious": "Vorherige Seite"
8             },
9             "sInfo": "Es wurden _TOTAL_ Dateien gefunden. Aktuelle Anzeige (
10                 _START_ bis _END_)",
11             "sLengthMenu": "Zeige _MENU_ Dateien"
12         }
13     });
14     mTable.fnClearTable();
15     jQuery.each(files, function() {
16         mTable.fnAddData(['<a href="files/' + this.name + '" > ' + this.name + '
17             </a>', this.size, '<form method="POST" action=""><input type="
18             hidden" name="file" value="' + this.name + '"><input type="submit"
19             name="delete" value="Löschen"></form>']);
20     });
21     mTable.draw();
22 });
```



Demonstration



Es folgt eine Demonstration ...