

PHP: Login Tutorial mit Sessions und Registrierung

Ziel ist es nicht eine komplexe Userverwaltung zu erstellen. Ziel ist es, einem PHP Anfänger die wichtigsten Funktionen zu vermitteln!

In diesem Tutorial werdet ihr lernen, wie..

- * man Formulare auswertet
- * man in Dateien schreibt
- * Dateien ausgelesen werden
- * Arrays funktionieren

== Schritt 1: Textdatenbank anlegen ==

Zu aller Anfang erstellen wir eine leere Datei namens "db.txt"

Diese Datei dient uns als Textdatenbank. In ihr werden wir Benutzername und Passwort speichern.

Damit ihr diese Datei aus eurem PHP Skript heraus beschreiben könnt müsst ihr Sie schreibbar machen. Unter UNIX Systemen funktioniert das mit CHMOD. Setzt den Wert mit eurem FTP Programm auf 666.

== Schritt 2: Registrierung ==

Wie ihr in Zeile 4 des folgendes Skriptes seht, werden wir für jeden User eine neue Zeile anlegen (\n).

Benutzername und Passwort werden wir mit einem Tabulator (\t) trennen, damit wir die Daten beim späteren Auslesen unterscheiden können.

In Zeile 5 öffnen wir die Datei zum Schreiben mit der Option "a". Diese Option ("append") benutzen wir um etwas an die Datei anzuhängen.

Es gibt noch andere Optionen, die ihr alle im [phpdoc]fopen[/phpdoc] Manual nachlesen könnt.

MD5 ist eine One-Way-Verschlüsselung. Mehr dazu in Punkt 4.

Quellcode

```
1. <?php
2. // registrierung.php
3. if($_POST['submit'] == 1) {
4. $line = $_POST['user'] ."\t". md5($_POST['password'])."\n";
5. $handle = fopen('db.txt', "a");
6. fwrite($handle, $line)or die('Fehler beim Schreiben');
7. fclose($handle);
8. }
9. }
10. ?>
11. <form method="post" action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>">
12. <input type="hidden" name="submit" value="1" />
13. <table>
14. <tr>
15. <td>Benutzername:</td>
16. <td><input type="text" name="user" /></td>
17. </tr>
18. <tr>
19. <td>Passwort:</td>
20. <td><input type="password" name="password" /></td>
21. </tr>
22. <tr>
23. <td colspan="2">
24. <input type="submit" value="Registrieren" />
25. </td>
26. </tr>
27. </table>
28. </form>
```

Alles anzeigen

== Schritt 3: Login ==

Ist der User erstmal angelegt, wollen wir uns natürlich auch irgendo anmelden.
Das Formular habe ich folgendermaßen erstellt.

Quellcode

```
1. <?php
2. // login.php
3. ?>
4. <form method="post" action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>">
5. <input type="hidden" name="p_submit" value="1" />
6. <table>
7. <tr>
8. <td>Benutzername:</td>
9. <td><input type="text" name="p_user" value="<?php echo htmlspecialchars($_POST['p_user']); ?>" /></td>
10. </tr>
11. <tr>
12. <td>Passwort:</td>
13. <td><input type="password" name="p_password" /></td>
14. </tr>
15. <tr>
16. <td colspan="2">
17. <input type="submit" value="Login" />
18. </td>
19. </tr>
20. </table>
```

21. </form>

Alles anzeigen

== Schritt 4: Dateischutz ==

Nachdem die Basics erstellt wurden sind wir nun soweit PHP Dateien zu schützen.

Zuerst erstellen wir eine eigene Datei namens secure.php. Diese binden wir erst im nächsten Schritt in die zu schützenden Dateien ein.

Das session_start() in Zeile 2 ist Voraussetzung. Das if(\$_POST['submit'] == 1) prüft ob das Login Formular oder die Registrierung abgeschickt wurden.

Mit file() packen wir die Zeilen (\n) in Arrays, diese Arrays trennen wir dann mit explode nach dem "Tabulator" auf.

Dann überprüfen wir ob die in der Session gespeicherten Daten existieren und weisen der Variable \$login den Wert wahr zu, falls dies zutrifft.

Bei keinem Treffer brechen wir alles mit die() ab.

Quellcode

```
1. <?php
2. // secure.php
3. session_start();
5. if($_POST['p_submit'] == 1) {
6. $_SESSION['s_user'] = $_POST['p_user'];
7. $_SESSION['s_password'] = md5($_POST['p_password']);
8. }
9. $login = false;
10. $file = file('db.txt');
12. if($_SESSION['s_user'] != "" && $_SESSION['s_password'] != "")
13. foreach($file as $row) {
14. $row = explode("\t", $row);
16. $user = trim($row[0]);
17. $password = trim($row[1]);
19. if($user == $_SESSION['s_user'] && $password == $_SESSION['s_password']) {
20. $login = true;
21. }
22. }
23. if($login == false) {
26. if($_POST['p_submit'] == 1)
27. echo 'Fehler beim Login. Benutzername und/oder Passwort nicht korrekt.';
28. else
29. echo 'Bitte anmelden';
30. require('login.php');
32. die();
33. }
34. ?>
```

Alles anzeigen

== Schritt 5: Dateien schützen ==

Nun können wir .php Dateien schützen. Dazu brauchen wir nur folgende Zeile zu Beginn des Skripts einfügen

Quellcode

```
1. <?php
2. // admin.php
3. require('secure.php'); ?>
```

8. TO SECURE

== Schritt 6: Ausloggen ==

Wenn sich euer Benutzer abmelden will, dann bietet ihm einfach einen Link zu einer Seite namens logout.php an. Diese muss folgenden Inhalt haben.

Quellcode

1. <?php
2. // logout.php
3. session_start();
4. session_destroy();
5. ?>
6. Erfolgreich vom System abgemeldet