Einfache *Login Session* (ohne Datenbankzugriff)

Bekanntlich leidet HTTP unter Amnesie und vergisst alles, was weiter als ein Request entfernt ist.
Eine Möglichkeit dies zu beheben bieten **Sessions**.

$_SESSION

Ein assoziatives Array als superglobale Variable, das die Session-Variablen enthält und dem aktuellen Skript zur Verfügung gestellt wird.

Die Session bietet die Möglichkeit, Informationen über mehrere Requests zu speichern. Es ist somit möglich, Zustände wie eingeloggt, nicht eingeloggt nachzubilden.

$_SESSION["User"] = “tester“;

$_SESSION["Auth"] = 1;

Damit eine PHP-Seite mit Sessions arbeiten kann, muss als erstes immer die Funktion:

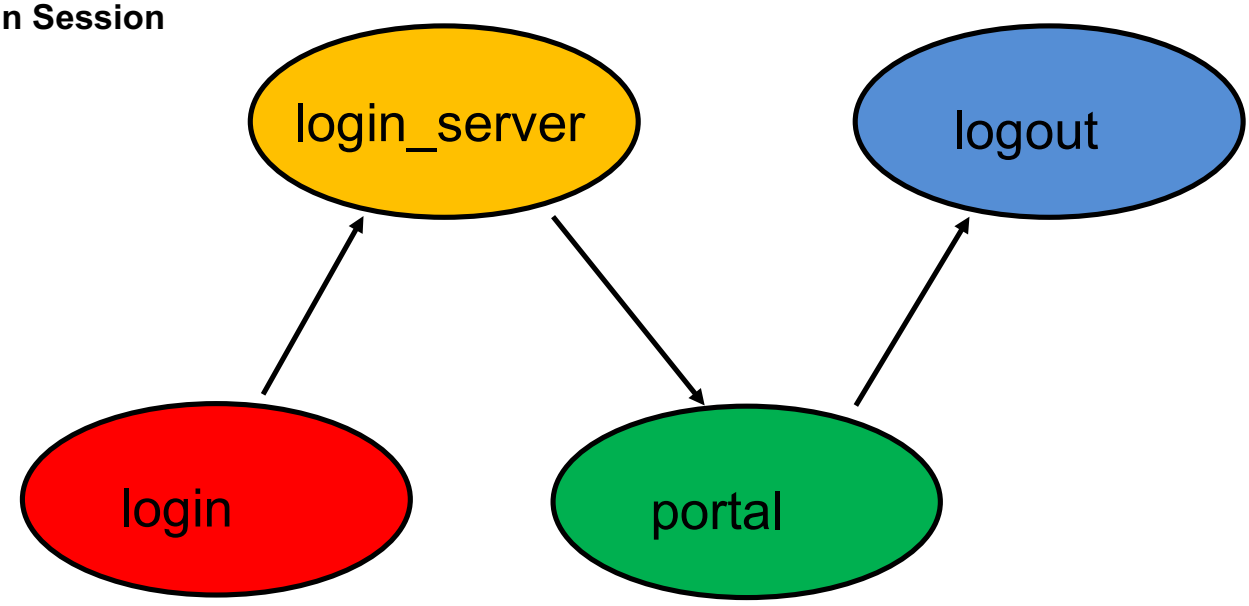
session_start( );

aufgerufen werden.

Die Funktion legt auf dem Server die Session an und gibt eine Session ID an den Client zurück. PHP speichert dann alle Daten, die unter dieser ID anfallen zentral und abrufbereit auf dem Web-Server.

Der Client speichert die Session-ID in der Regel in einem Cookie. Ein Problem besteht allerdings, wenn diese Cookies ausgeschaltet sind, die ID wäre wirkungslos und würde verloren gehen. Deshalb gibt es die Möglichkeit die Session ID in Hidden Input einzutragen oder als Parameter bei einem Link zu übergeben.

Login Session



```php
<?php
// Session starten
session_start();

// Session schon vorhanden?
if(isset($_SESSION['Auth']) && $_SESSION['Auth']==1){
  header('location: portal.php');
}

?>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Anmelden</title>
  </head>
  <body>
    <h1>Anmelden</h1>
    <form action="login_server.php" method="post">
      Benutzer: <input name="frmUser" type="text"><br>
      Kennwort: <input name="frmPass" type="text"><br>
      <input name="Login" type="submit"><br>
    </form>
  </body>
</html>

```

index.php

```php
<?php
$error = ''; // Initialisierung

session_start(); // Session starten

// Session schon vorhanden?
if(isset($_SESSION['Auth']) && $_SESSION['Auth']==1){
  header('location: portal.php');
} else {

  // Benutzereingaben vorhanden?
  if(isset($_POST["frmUser"]) && isset($_POST["frmPass"])) {

    //Benutzereingaben speichern
    $frmUser=$_POST["frmUser"];
    $frmPass=$_POST["frmPass"];

    // Stimmen die Angaben des Benutzers?
    if((strtolower($frmUser)=="tester") && ($frmPass=="geheim")){
      $_SESSION["User"]=$frmUser;
      $_SESSION["Auth"]=1;

      // Weiterleitung auf portal.php
      header('Location: portal.php');

    // Fehlermeldung: Benutzereingaben stimmen nicht überein.
    } else {
      $error .= "Ihre Eingabe war falsch, melden Sie sich bitte auf der
      <a href=\"index.php\">Login-Seite</a> neu an<br / >";
    }
  // Fehlermeldung: wenn keine Eingabe gemacht wurde.
  } else {
    $error .= "Ihre Eingabe war falsch, melden Sie sich bitte auf der
    <a href=\"index.php\">Login-Seite</a> neu an<br / >";
  }
}
?>

```

```php
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Prüfen der Benutzereingaben / Session erstellen </title>
  </head>
  <body>
    <h1>Fehlermeldung</h1>
    <?php
      echo $error; // Ausgabe der Fehlermeldung
    ?>
  </body>
</html>

```

login_server.php

```php
<?php
// Session starten
session_start();

// Session prüfen
if (!isset($_SESSION["Auth"]) || $_SESSION["Auth"]!=1){

  // weiterleiten auf login.php
  header('Location: index.php');
}
?>

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Session gestartet</title>
  </head>
  <body>
    <h1>Hallo <?php echo $_SESSION["User"]?>, Herzlich Willkommen im Portal!</h1>

     <a href = "logout.php">Logout</a><br />

      <?php
        echo "<h3>Der Inhalt deiner Session:</h3>";
        echo "<pre>";
        print_r($_SESSION); // DUMP $_SESSION
        echo "</pre>";
        echo "<h3>Weitere Informationen über die Session: </h3>";
        echo "Session Name: " . session_name() . "<br />"; // Ausgabe Session Name
        echo "Session ID: " . session_id() . "<br />"; // Ausgabe Session ID
        echo "Session Pfad : " . session_save_path() . "<br />"; // Pfad zu Session File
        echo "Session Cache: " . session_cache_expire() . "<br />"; // Session Cache in Minuten
      ?>
  </body>
</html>

```

portal.php

Auftrag:

1. Testen Sie die Applikation.
2. Sollte bereits eine Session vorhanden sein, leiten Sie den User in den Scripts index.php und login_server.php auf die portal.php Seite weiter.
3. Sollte noch KEINE Session vorhanden sein, leiten Sie den User in den Scripts portal.php auf die index.php Seite weiter.
4. Erweitern Sie die Applikation durch eine **logout-Funktion** welche auf der Portal-Seite aufgerufen werden kann.
5. Versuchen Sie in der Logout-Funktion auch das „PHPSESSID“-Cookie zu löschen.
6. Lesen Sie die beiden Blog-Einträge
   http://www.d-mueller.de/blog/php-session-management-erklaert/
   https://d-mueller.de/blog/angriffe-auf-webanwendungen-teil-2-session-highjacking-und-session-fixation/
   und sichern Sie Ihre Applikation gegen Session-Highjacking und Session-Fixation.
7. Notieren Sie Ihre Erkenntnisse in Ihrem Lernjournal.

D. A. Waldvogel / D. Brodbeck, 25.05.2017
M133_Session_V2.docx