

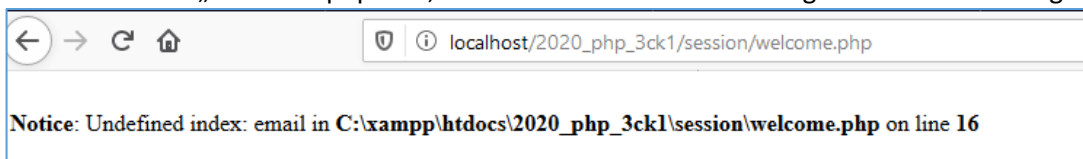
Login mit sessions

Sessions werden genutzt, um die Authentifizierung für Webseiten sicherer zu machen. Das Skript wird ausgeführt, nachdem die Benutzeranmeldeschaltfläche (submit-Button) gesendet wurde. Die Anmeldeseite auf der Sitzung. Wenn der Benutzer die Sitzung schließt, werden die Sitzungsdaten gelöscht.

Sessions dienen zum Schutz von sensiblen Seiten.

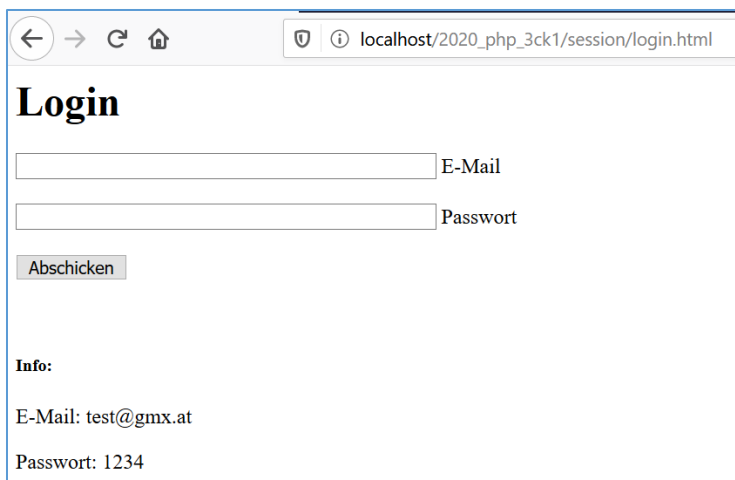
Ist keine Session vorhanden, und das ist sie nur, wenn der LOGIN erfolgreich war, dann kann man auch nicht die sensible Seite durch direkte Eingabe in der URL-Zeile aufrufen. Man kann somit das System nicht austricksen:

- Gibt man direkt „welcome.php“ ein, erhält man eine Fehlermeldung – das ist beabsichtigt.



- Gibt man direkt „inhalt.php“ ein, greift ebenfalls die Absicherung und man wird sogar auf die Login-Seite geleitet.

1)Erstelle bzw. verwende weiterhin die „login.html“.

A screenshot of a web browser window showing a login form. The address bar displays 'localhost/2020_php_3ck1/session/login.html'. The page has a heading 'Login' and two input fields: 'E-Mail' and 'Passwort'. Below the fields is a button labeled 'Abschicken'. At the bottom, there is an 'Info:' section with the text 'E-Mail: test@gmx.at' and 'Passwort: 1234'. The browser's navigation buttons are visible at the top left.

Klickt man auf den Login-Button, wird die „action“ ausgelöst und man auf die Seite „welcome.php“ geleitet.

Code:

```
<html>
  <head></head>
  <body>
    <h1>Login</h1>
    <form action = "welcome.php" method = "post" >
      <input name = "email" type="email" size = "50">
      <label>E-Mail </label>
      <br><br>
      <input name = "passwort" type="password" size = "50">
```

```

        <label>Passwort </label>
        <br><br>
        <input type="submit" name="login" value="Login">
    </form>
    <br>
    <h5>Info:</h5>
    <p>E-Mail: test@gmx.at</p>
    <p>Passwort: 1234</p>
</body>
</html>

```

```

1  <html>
2      <head></head>
3  <body>
4      <h1>Login</h1>
5  <form action = "welcome.php" method = "post" >
6      <input name = "email" type="email" size = "50">
7      <label>E-Mail </label>
8      <br><br>
9      <input name = "passwort" type="password" size = "50">
10     <label>Passwort </label>
11     <br><br>
12     <input type="submit" name="login" value="Login">
13 </form>
14 <br>
15 <h5>Info:</h5>
16 <p>E-Mail: test@gmx.at</p>
17 <p>Passwort: 1234</p>
18 </body>
19 </html>

```

2)welcome.php

Erstelle eine Seite „welcome.php“.

Am Beginn legt man gleich die korrekten Zugangsdaten statisch in zwei Variablen ab. Darunter startet man die „session-Funktion“:

```

1  <?php
2
3  $email="test@gmx.at";
4  $psw=1234;
5
6  session_start();

```

Info:

Das erste IF hat beim ersten Durchlauf noch keine Session gesetzt, da die Session zwar gestartet wird in Zeile 6, aber die Zuordnung erst in der Zeile 17 erfolgt.

- Daher ist die erste IF „false“, es gibt ja noch keine SESSION und somit wird
- „else“ ausgeführt. Hier findet nun die
- Erstellung der SESSION „email“ statt (Zeile 17). Hier wird dann gleich darunter diese
- Seite neu geladen (header('Location: welcome.php');) und somit ist nun die
- Session korrekt vorhanden und es wird
- IF mit true beantwortet und der Inhalt aus den Zeilen 11-13 angezeigt.

- Wäre die Eingabe falsch, wird die Session nicht gesetzt und es kommt die Fehlermeldung.

Somit wird beim ersten Durchlauf der erste IF-Block übersprungen, da noch keine Session zugewiesen ist. Erst durch das Wiederladen (reload) passt es. Also ohne korrektes Login, existiert dieser erste IF-Block somit nicht, weil er ja übersprungen wird.

Die erste IF-Abfrage schaut also, ob die Session gesetzt wurde. Das passiert mit „isset()“.

isset():

Die Funktion isset() überprüft, ob eine Variable oder ein Array existent ist. Diese Funktion liefert true zurück, wenn die Variable oder das Array existent ist, sonst false.

Ist sie „true“, dann heißt das hier, dass der Nutzer eingeloggt ist. Daher wird er begrüßt und zwei Links angezeigt: Die Begrüßung wird mit \$_SESSION['email'] erzeugt, was mit Verkettungspunkten davor und danach erstellt wird.

```

8 //testen ob die session gesetzt wurde
9 ▼ if(isset($_SESSION['email'])){
10
11     echo "<h2>Willkommen mit der E-Mail " . $_SESSION['email'] . "</h2>";
12     echo "<a href='inhalt.php'>Inhalt anzeigen</a><br>";
13     echo "<br><a href='logout.php'><input type='button' value='logout'
14         name='logout'></a>";
15 } else

```

Darunter wird ab Zeile 16 geprüft, ob die E-Mail und das Passwort aus der Eingabe auch mit den hinterlegten Daten übereinstimmen.

```

16 ▼ if ($_POST['email'] == $email && $_POST['password'] == $psw) {
17     $_SESSION['email']=$email;
18     //Weiterleitung zur Seite selbst - reloading
19     header('Location: welcome.php');
20 ▼ } else {
21     echo "<script>alert('Leider ist der Zugang falsch')</script>";
22 }
23 }
24 ?>

```

Ist die Session gesetzt, erfolgt eine Weiterleitung zur Seite selbst, in der man sich gerade befindet (reloading):

```
header('Location: welcome.php');
```

Ansonsten wird eine Fehlermeldung (JavaScript) ausgegeben:

```

20 ▼ } else {
21     echo "<script>alert('Leider ist der Zugang falsch')</script>";
22 }
23 }
24 ?>

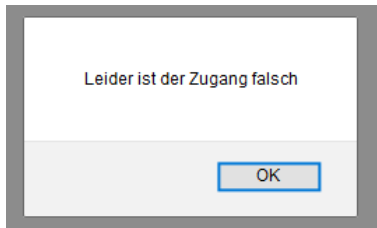
```

Ergebnis:

Korrekte Daten:



Ergebnis bei falsche Daten:



3) Inhaltseite

Erstelle folgende „inhalt.php“.

Dabei gibt es auch einen einfachen Button, der zur Welcome-Seite zurück springen lässt.



```
1 <?php
2 session_start();
3
4 if(isset($_SESSION['email']))
5 {
6     echo "<h2>Willkommen auf der Inhaltseite</h2>";
7     echo "<br><a href='welcome.php'><input type='button' value='Zurück'></a>";
8 } else
9 {
10    header('Location: login.html');
11 }
12 ?>
```

Code:

```
<?php
```

```
session_start();
```

```
if(isset($_SESSION['email'])){
```

```

    echo "<h2>Willkommen auf der Inhaltseite</h2>";
    echo "<br><a href='welcome.php'><input type='button' value='back' name='back'></a>";
} else
{
    header('Location: login.html');
}
?>

```

In Zeile 2 wird die Session gestartet, die ja vorhanden ist und die in Zeile 4 mit der IF-Abfrage sofort benötigt wird.

Ist die Session „email“ vorhanden, wird der geheime Inhalt angezeigt, ansonsten wird man auf die ursprüngliche Login-Seite zurückverwiesen.

4) logout.php

Diese Seite ist nie sichtbar, sondern leitet nur weiter:

```

1  <?php
2  session_start();
3
4  if(isset($_SESSION['email'])){
5      session_destroy();
6      header('Location: login.html');
7  } else
8  {
9      header('Location: login.html');
10     }
11  ?>

```

Code:

```

<?php
session_start();

if(isset($_SESSION['email'])){

    session_destroy();
    header('Location: login.html');
} else
{
    header('Location: login.html');
}
?>

```

Quelle: <https://www.youtube.com/watch?v=O0Ky0tKvsJ8>