5.0 PDO en Sessies

In deze les wordt een demo gegeven van hoe je met PDO en sessie variabelen een login kunt maken.
Je leert gebruik maken van PDO en van PHP sessies.

De wachtwoorden worden in deze les niet-encypted opgeslagen. Op deze manier kan je goed zien wat er gebeurt.

Sessions

Een sessie is wordt gestart met het php commando `session_start()`. Hiermee krijgt jouw browser sessie een stukje geheugen op de web server waar sessie variabelen worden opgeslagen.

Elke keer als je `session_start()` aanroept wordt er eerst gekeken of er al een sessie bestaat, als die bestaat dan worden alle sessie variabelen door de server naar de browser gestuurd. Bestaat de sessie nog niet, dan wordt er een sessie opgestart en wordt er een stukje geheugen op de server voor jou gereserveerd.

Binnen een sessie kan je met `$_SESSION['naam']='xyz'` de sessie variabele naam de waarde 'xyz' geven.

Met `session_destroy()` kun je de sessie beëindigen. Alles sessie variabele worden dan weg gegooid.

Als je een keer bent aangelogd dan zet je in een sessie variabele een indicator dat je bent aangelogd. Je kunt bijvoorbeeld jouw userid in een sessie variabele zetten. Dit userid wordt dan elke keer door de server nar jouw browser gestuurd,

Login

Je maakt dus een form waarin je een gebruikersnaam en wachtwoord aan de gebruiker opvraagt. Met PDO voer je een query uit en controleer je of de combinatie gebruikersnaam en wachtwoord klopt. Als dat zo is dan start je een sessie en zet je (bijvoorbeeld) het userid in een sessie variabele. Je hoeft dan geen wachtwoord te onthouden en telkens opnieuw aanloggen. De sessie variabele is een betrouwbare indicator dat je bent aangelogd.

Database

Maak een database test en draai het volgende SQL script:

```
CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `naam` varchar(100) NOT NULL,
  `wachtwoord` varchar(100) NOT NULL,
  `rol` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `users` (`id`, `naam`, `wachtwoord`, `rol`) VALUES
(1, 'max', 'max', 1),
(2, 'piet', 'piet', 1);

ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `users`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
COMMIT;
```

De database heeft de tabel users waarin het id, naam, wachtwoord en rol (id) van de gebruiker staan.

DB

De verbinding van de database wordt in **db.php** gedefinieerd.

```
<?php

$dsn = "mysql:host=localhost;dbname=test";
$user = "root";
$passwd = "";

try {
  $pdo = new PDO($dsn, $user, $passwd);
} catch (Exception $e) {
  echo 'Caught exception: ',  $e->getMessage(), "\n";
}



?>
```

Landing page

De **index.html** is alleen maar een landing page die een zeer eenvoudig menu laat zien.

```
<html>
<body>

<a href='index.php'>Landing page</a> -
<a href="login.php">Login</a> -
<a href="logout.php">Log out</a> -
<a href='page.php'>Just another page</a>
<hr>

Landing page
```

Login page

Deze pagina laat een laat een form zien waarmee je kunt inloggen. Als je het form post dan wordt er gecontroleerd aan de hand van de database of de opgegeven gebruikersnaam bestaat en of het wachtwoord klopt.

Als je bent aangelogd dan wordt er een sessie variabele gezet. Aan de hand van deze sessie variabele kan je later zien of je bent aangelogd. Zolang de sessie blijft bestaan zal de server elke keer de sessie variabelen meesturen naar jouw browser.

Maak de file **login.php**

```
<!DOCTYPE html>
<html>
<body>

  <a href='index.php'>Landing page</a> -
  <a href="login.php">Login</a> -
  <a href="logout.php">Log out</a> -
  <a href='page.php'>Just another page</a>
<hr>

<?php

if ( isset($_GET['naam']) &&  isset($_GET['wachtwoord']) ) {
```

```
  $naam = $_GET['naam'];
  $ww=$_GET['wachtwoord'];

  include "db.php";
  echo "Trying to login with ".$_GET['naam'];

  $stmt = $pdo->prepare("SELECT * FROM users where naam = :naam and wachtwoord=:wachtwoord");
  $stmt->bindParam(':naam', $naam, PDO::PARAM_STR);
  $stmt->bindParam(':wachtwoord', $ww, PDO::PARAM_STR);
  $stmt->execute();
  $result = $stmt->fetch();

  if ($result ) {

    echo "<pre>";
    var_dump($result);
    echo "</pre>";

    session_start([ 'cookie_lifetime' => 86400,]);
    $_SESSION['id'] = $result['id'];
    $_SESSION['naam'] = $result['naam'];

    echo "<br>Succesfully loggin in!";
    echo "<br><a href='page.php'>Just another page</a>";
    exit;
  }
}

?>

<h2>Login</h2>

<form action="login.php">
  <label for="naam">Naam:</label><br>
  <input type="text" id="naam" name="naam" value=""><br>
  <label for="wachtwoord">Wachtwoord:</label><br>
  <input type="text" id="wachtwoord" name="wachtwoord" value=""><br><br>
  <input type="submit" value="Submit">
</form>
```

```
</body>
</html>
```

Logout

Als je de sessie beëindigt met sessie_destroy() dan is de sessie beëindigd en weet je browser niet meer of je bent aangelogd.

Maak een **logout.php**

```
<!DOCTYPE html>
<html>
<body>

  <a href='index.php'>Landing page</a> -
  <a href="login.php">Login</a> -
  <a href="logout.php">Log out</a> -
  <a href='page.php'>Just another page</a>
<hr>

<?php
session_start();
session_destroy();
?>
Uitgelogd
```

Page

Tenslotte maken we nog een pagina die gewoon laat zien of we aangelogd zijn of niet.

Maak de pagina **page.php**

```
<!DOCTYPE html>
<html>
<body>

  <a href='index.php'>Landing page</a> -
  <a href="login.php">Login</a> -
  <a href="logout.php">Log out</a> -
  <a href='page.php'>Just another page</a>
  <hr>
```

```php
<?php

session_start();

if (isset($_SESSION['id'])) {
  echo  "<br>You are logged in and your user id is    ". $_SESSION['id'];
  echo  "<br>and your naam id is ". $_SESSION['naam'];
} else {
    echo  "<br>You are NOT logged in";
}

?>
```

Revision #4
Created 2020-10-05 11:17:01 UTC by Max
Updated 2020-10-06 08:32:49 UTC by Max