

Tutorial Sessões em PHP

Como usar Sessões em PHP - 02

Objetivos

- Desenvolver um Sistema de Login usando o PHP e PDO

Estrutura dos arquivos do projeto

Este tutorial conterá os seguintes arquivos principais:

- a) connect.php: arquivo com os comandos para conectar-se ao BD
- b) register.php: formulário de registro de usuários
- c) login.php: script que efetua o login do usuário usando um formulário
- d) home.php: script para certificar se o usuário está logado

Passo 1:

Crie uma pasta chamada “pdo_sessao_02” dentro daquela que está no servidor de sua máquina (i.e. “htdocs” ou “public_html”)

Passo 2:

Com o PhpMyAdmin, crie um BD chamado “pdo_sessao_02_bd”.

Passo 3:

Em seguida, execute o seguinte código SQL. Ele serve para criar nossa tabela no BD.

```
CREATE TABLE IF NOT EXISTS `users` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `username` varchar(25) NOT NULL,  
  `password` varchar(60) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `username` (`username`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci AUTO_INCREMENT=1  
;
```

Tutorial Sessões em PHP**Passo 4:**

Agora vamos precisar de um arquivo para conexão com o BD. Para isto, crie um novo arquivo chamado ("connect.php") no diretório ("pdo_sessao_02") e cole (CTRL+V) o seguinte código:

```
<?php

//connect.php

/**
 * This script connects to MySQL using the PDO object.
 * This can be included in web pages where a database connection is needed.
 * Customize these to match your MySQL database connection details.
 * This info should be available from within your hosting panel.
 */

//Our MySQL user account.
define('MYSQL_USER', 'root');

//Our MySQL password.
define('MYSQL_PASSWORD', 'root');

//The server that MySQL is located on.
define('MYSQL_HOST', 'localhost');

//The name of our database.
define('MYSQL_DATABASE', 'pdo_sessao_02_bd');

/**
 * PDO options / configuration details.
 * I'm going to set the error mode to "Exceptions".
 * I'm also going to turn off emulated prepared statements.
 */
$pdoOptions = array(
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_EMULATE_PREPARES => false
);

/**
 * Connect to MySQL and instantiate the PDO object.
 */
$pdo = new PDO(
    "mysql:host=" . MYSQL_HOST . ";dbname=" . MYSQL_DATABASE, //DSN
    MYSQL_USER, //Username
    MYSQL_PASSWORD, //Password
    $pdoOptions //Options
);

//The PDO object can now be used to query MySQL.
```

Tutorial Sessões em PHP

Passo 5:

Agora crie um novo arquivo chamado ("register.php") no diretório ("pdo_sessao_02") e cole (CTRL+V) o seguinte código:

```
<?php

//register.php

/**
 * Start the session.
 */
session_start();

/**
 * Include our MySQL connection.
 */
require 'connect.php';

//If the POST var "register" exists (our submit button), then we can
//assume that the user has submitted the registration form.
if(isset($_POST['register'])){

    //Retrieve the field values from our registration form.
    $username = !empty($_POST['username']) ? trim($_POST['username']) : null;
    $pass = !empty($_POST['password']) ? trim($_POST['password']) : null;

    //TO ADD: Error checking (username characters, password length, etc).
    //Basically, you will need to add your own error checking BEFORE
    //the prepared statement is built and executed.

    //Now, we need to check if the supplied username already exists.

    //Construct the SQL statement and prepare it.
    $sql = "SELECT COUNT(username) AS num FROM users WHERE username = :username";
    $stmt = $pdo->prepare($sql);

    //Bind the provided username to our prepared statement.
    $stmt->bindValue(':username', $username);

    //Execute.
    $stmt->execute();

    //Fetch the row.
    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    //If the provided username already exists - display error.
    //TO ADD - Your own method of handling this error. For example purposes,
    //I'm just going to kill the script completely, as error handling is outside
```

Tutorial Sessões em PHP

```
//the scope of this tutorial.
if($row['num'] > 0){
    die('That username already exists!');
}

//Hash the password as we do NOT want to store our passwords in plain text.
$passwordHash = password_hash($pass, PASSWORD_BCRYPT, array("cost" => 12));

//Prepare our INSERT statement.
//Remember: We are inserting a new row into our users table.
$sql = "INSERT INTO users (username, password) VALUES (:username, :password)";
$stmt = $pdo->prepare($sql);

//Bind our variables.
$stmt->bindValue(':username', $username);
$stmt->bindValue(':password', $passwordHash);

//Execute the statement and insert the new account.
$result = $stmt->execute();

//If the signup process is successful.
if($result){
    //What you do here is up to you!
    echo 'Thank you for registering with our website.';
}

}

?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Register</title>
    </head>
    <body>
        <h1>Register</h1>
        <form action="register.php" method="post">
            <label for="username">Username</label>
            <input type="text" id="username" name="username"><br>
            <label for="password">Password</label>
            <input type="text" id="password" name="password"><br>
            <input type="submit" name="register" value="Register"></button>
        </form>
    </body>
</html>
```

Tutorial Sessões em PHP**Passo 6:**

Agora crie um novo arquivo chamado ("login.php") no diretório ("pdo_sessao_02") e cole (CTRL+V) o seguinte código:

```
<?php

//login.php

/**
 * Start the session.
 */
session_start();

/**
 * Include our MySQL connection.
 */
require 'connect.php';

//If the POST var "login" exists (our submit button), then we can
//assume that the user has submitted the login form.
if(isset($_POST['login'])) {

    //Retrieve the field values from our login form.
    $username = !empty($_POST['username']) ? trim($_POST['username']) : null;
    $passwordAttempt = !empty($_POST['password']) ? trim($_POST['password']) : null;

    //Retrieve the user account information for the given username.
    $sql = "SELECT id, username, password FROM users WHERE username = :username";
    $stmt = $pdo->prepare($sql);

    //Bind value.
    $stmt->bindValue(':username', $username);

    //Execute.
    $stmt->execute();

    //Fetch row.
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    //If $row is FALSE.
    if($user === false) {
        //Could not find a user with that username!
        //PS: You might want to handle this error in a more user-friendly manner!
        die('Incorrect username / password combination!');
    } else {
        //User account found. Check to see if the given password matches the
        //password hash that we stored in our users table.

        //Compare the passwords.
        $validPassword = password_verify($passwordAttempt, $user['password']);
```

Tutorial Sessões em PHP

```
//If $validPassword is TRUE, the login has been successful.
if($validPassword){

    //Provide the user with a login session.
    $_SESSION['user_id'] = $user['id'];
    $_SESSION['logged_in'] = time();

    //Redirect to our protected page, which we called home.php
    header('Location: home.php');
    exit;

} else{
    //If $validPassword was FALSE. Passwords do not match.
    die('Incorrect username / password combination!');
}
}

?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Login</title>
    </head>
    <body>
        <h1>Login</h1>
        <form action="login.php" method="post">
            <label for="username">Username</label>
            <input type="text" id="username" name="username"><br>
            <label for="password">Password</label>
            <input type="text" id="password" name="password"><br>
            <input type="submit" name="login" value="Login">
        </form>
    </body>
</html>
```

Passo 7:

Crie um novo arquivo chamado “home.php” no diretório (“pdo_sessao_02”) e cole (CTRL+V) o código abaixo nele.

```
<?php

//home.php

/**
 * Start the session.
 */
```

Tutorial Sessões em PHP

```
session_start();

/**
 * Check if the user is logged in.
 */
if(!isset($_SESSION['user_id']) || !isset($_SESSION['logged_in'])){
    //User not logged in. Redirect them back to the login.php page.
    header('Location: login.php');
    exit;
}

/**
 * Print out something that only logged in users can see.
 */

echo 'Congratulations! You are logged in!';
```