Öğr. Gör. Serkan AKSU
http://www.serkanaksu.net

PHP'de Oturum Yönetimi (Session Management)

Sessionlar php'de oturum yönetimi için kullanılır. Oturumlar ise ziyaretçiyi takip etmek başka sayfalara veri aktarmak için kullanılır. **Session**u en çok kullandığımız yerlerden birisi de üyelik sistemleridir. Ziyaretçi üye girişi yapmışsa ona bir oturum değişkeni atarız ve bu değişkenin varlığına göre üye girişi yapıp yapmadığını kontrol ederiz. Böylece ziyaretçinin istediğimiz alanlara girmesini ya da girmemesini sağlayabiliriz.

PHP, çok gelişmiş bir "**oturum yönetim sistemi**" ile beraber gelir. Bu nedenle çok fazla detayla uğraşmaya gerek kalmaz.

Bilinmesi gereken en önemli konu **"session identifier"** denilen (genel olarak bilinen adıyla **SID**) özel bir değişkenin, her istemciyi ayırabilmek için referans numarası verilmek suretiyle atanmasıdır.

Eğer oturumu başlatılırsa server tarafından otomatik olarak **SID** atanır. Herhangi bir değişken "oturum değişkeni" olarak kayıt edildiği anda server üzerinde cookie benzeri bir dosya açılır. (Bu dosyanın adı genelde **SID**'nin değeri ile aynıdır). İstemci tarafındaki bilgisayar ne zaman bu veriye ulaşmak isterse, yapması gereken tek şey istekle beraber **SID** numarasını da beraber göndermektir.

Session (oturum) ile ilgili verileri server ortamında saklar ve veriler server ortamından okunur. Ziyaretçinin bilgisayarına hiç bir veri gönderilmez ve **tarayıcı kapatıldığında sessionlar otomatik olarak silinirler**. Yani oturum sonlandırılır. Bir oturum işlemi şu aşamalardan oluşur:

- Öncelikle bir oturum başlatırız ve bu oturuma özel bir numara (**Session ID**) atanır.
- Bu **ID** ile aynı adı taşıyan bir dosya sunucuda oluşturulur.
- Oturum boyunca verilere bu dosyadan erişilir.
- Ziyaretçi siteden ayrıldığında yani tarayıcıyı kapattığında dosya silinir ve oturum sonlandırılır.

Öğr. Gör. Serkan AKSU
http://www.serkanaksu.net

Oturum Başlatmak

```
<?php
        session_start();
        ..
?>
```

Bir Oturum başlatmak için **session_start()** fonksiyonu kullanılır. Bu fonksiyon sayfanın en üstünde yer almalıdır. Aksi takdirde hata mesajı döndürür.

Başlatılmış Bir Oturumun ID'sine Erişmek:

Başlatılmış bir oturum ID'sine erişmek için **session_id()** fonksiyonu kullanılır. Bu fonksiyon ile başlatılan oturumun ID'sini elde edilir.

Bir oturum değişkenini şu şekilde oluşturulabilir:

$_SESSION['session_degisken_adi'] = "değer";

session_degisken_adi şeklinde başlattığımız oturumun adını ve değer şeklinde de taşıyacağı değeri belirtiyoruz.

Basit bir oturum örneği:

sayfa1.php

```php
<?php
session_start();
$_SESSION['uyeID'] = 4;
$_SESSION['AD'] = "Serkan Aksu";
?>
<a href="sayfa2.php">2. Sayfaya Gidin</a>
```

sayfa2.php

```php
<?php
session_start();
$sid = session_id();
if(isset($_SESSION['uyeID'])){
    $uyeid = $_SESSION['uyeID'];
    $ad = $_SESSION['AD'];
    echo '<br>Üye ID niz: '.$uyeid;
    echo '<br>Adınız: '.$ad;
    echo '<br>başlatılan oturumun id si :'.$sid;
} else echo "Oturum Başlatılmamış";
?>
<br><a href="sayfa1.php">1. Sayfaya Gidin</a>
  
<a href="sayfa3.php">Oturumu Kapatın</a>
```

sayfa3.php

```php
<?php
session_start();
unset($_SESSION['uyeID']);
unset($_SESSION['AD']);
session_destroy();
?>
<br><a href="sayfa1.php">1. Sayfaya Gidin</a>
  
<a href="sayfa2.php">2. Sayfaya Gidin</a>
```

sayfa1'de **session_start()** fonksiyonu ile oturum başlatılmış ve **$_SESSION** ile **uyeID** ve **AD** session değişkenlerine atama yapılmıştır. **sayfa2**'de atanan bu **session** değişkenleri okunarak işlenmiştir.

sayfa3'de **unset()** fonksiyonu ile session değişkenleri bellekten silinmiş ve açık olan oturum **session_destroy()** fonksiyonu ile sonlandırılmıştır.

Öğr. Gör. Serkan AKSU
http://www.serkanaksu.net

Kullanıcı Kimlik Denetimi Uygulaması

Aşağıdaki örnekten bir veritabanından kullanıcı sorgulaması yapılıyor ve eğer kullanıcı varsa ilgili **UID** adlı **session** değişkenine kullanıcının u**yeID** numarası atanıyor.

Kimlik bilgilerinin sorgulanacağı üyeler tablosu aşağıda verilmiştir.

UID	Ad	Soyad	Ogrenci	Eposta	Parola	UTarihi	Bildir	Yetki
1	SERKAN	AKSU	0	aksuse@gmail.com	srkn	2014-03-25 19:16:43	1	2
2	FARUK	DEMİR	1	farukd@yahoo.com	fd	2014-03-06 17:20:32	1	2
3	BAYRAM	AKGÜ	0	bayrama@gmail.com	ba	2014-03-06 17:32:13	1	2

dbconn.php

```php
<?php
//Oturum işlemlerinin yapılabilmesi için
session_start();
//PHP sayfalarında Türkçe karakterlerin görüntülenmesi için
@header("Content-Type: text/html; charset=utf-8");
//Data Source Name bilgileri.
$dsn = 'mysql:host=localhost;dbname=mysite_db';
//Kullanıcı bilgileri.
$dbuser = 'root';
$dbpass = 'mysql';
//Bağlantı kuruluyor.
try {
    $pdo = new PDO($dsn, $dbuser, $dbpass);
     $pdo->exec("SET NAMES 'utf8'; SET CHARSET 'utf8'");
     //echo "Bağlantı kuruldu";
} catch (PDOException $e) {
     //eğer bağlantıda bir sorun olursa hata mesajı yazılacak
    echo 'Bağlantı hatası: ' . $e->getMessage();
    die();
}
```

login_form.php

```
<?php
include "dbconn.php";
//Eğer oturum açılmışsa kullanıcının UID bilgisi yazılacak
if(isset($_SESSION['uid']))
{
      $uid = $_SESSION['uid'];
      echo "Oturum Açıldı UID:".$uid;
}
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Oturum Açma</title>
<link href="style.css" rel="stylesheet">
</head>
<body>
<div id="container">
<h2>Oturum Açma</h2>
<form action="login.php" method="post">
      <table>
      <tr>
      <td>Eposta</td><td>
            <input type="text" name="txtEposta" value="" />
      </td></tr>
      <tr>
      <td>Parola</td><td>
            <input type="password" name="txtParola" value=""/>
      </td></tr>
      <tr>
      <td></td><td>
            <input type="submit" value="Oturumu Aç" /><br />
            <A href="login.php">Oturumu Kapat</A>
      </td></tr>
      </table>
</form>
</div>
</body>
</html>
<?php
$pdo = NULL;
?>
```

login.php

```
<meta charset="utf-8">
<?php
include "dbconn.php";
if(isset($_POST["txtEposta"]) && isset($_POST["txtParola"]))
{     //Eğer formdan Eposta ve PArola verileri gelmişse oturum açılıyor.
      $ep = $_POST["txtEposta"];
      $par = $_POST["txtParola"];

      $sql = "SELECT * FROM uyeler WHERE Eposta LIKE :eposta AND Parola
LIKE :parola";
      //Çalıştırılmak üzere bir SQL deyimini hazırlanıyor.
      $query = $pdo->prepare($sql);
      //bindParam ile sorgular temizlenerek
      //veritabanına gönderiliyor.
      $query->bindParam(':eposta', $ep, PDO::PARAM_STR);
      $query->bindParam(':parola', $par, PDO::PARAM_STR);
      $query->execute();
      //Sorgu çalıştırılıyor
      if ($query->rowCount() > 0){
            $row = $query->fetch(PDO::FETCH_ASSOC);
            $_SESSION['uid'] = $row['UID'];
      }

}else{
      //Eğer formdan Eposta ve PArola verileri gelmemişse
      //oturum kapatılıyor.
      unset($_SESSION['uid']);
      session_destroy();
}
$pdo = NULL;
echo "<A href='javascript:history.back()'>Geri</A>";
?>
```

prepare() Metodu

prepare() Çalıştırılmak üzere bir SQL deyimini hazırlar. Bu metod **indparam()**, **execute()**, **bindColumn()**, **bindValue()** metotları ile beraber çalışır. Dışarıdan SQL sorgularına dahil edilecek veriler için iki tür tanım yapmayı sağlar.Bunlardan birisi soru işaretidir (?). Diğeri ise önünde iki nokta üst üste olan herhangi bir **:isimdir**.

bindParam() Metodu

bindParam(':parola', $parola, PDO::PARAM_STR) metodu, **prepare($sql)** metodu ile hazırlanan **SQL** sorgusunda **?** işareti veya bir **:isim** ile belirtilen parametrelerin hazırlanıp değişkenler için tanımlanmasını sağlar.

Güvenliğin önemli olduğu yerlerde formadan gelen veriler, **bindParam()** ile temizlenip **SQL Injection** gibi saldırıların önüne geçilmesi sağlanmış olur.

SQL sorguları her çalıştırmada yeniden yorumlanır. Sql sorgusunun hatalara karşı yorumlanması veritabanı sunucusu için doğaldır. SQL sorgularına dışarıdan gelen bilgiler aynen çalışır. Önlem alınmadığı takdirde dışarıdan gelen zararlı bilgilerde çalıştırılabilmektedir. Buna SQL **Injection** denmektedir. Zararı en aza indirmek için PDO quote() metodu özel karakterlerin temzilenmesini sağlar.

execute() Metodu

prepare() metodu ile hazırlanmış bir SQL deyimini çalıştırır.Aynı zamanda aldığı ikinci bir parametre ile prepare() metodu ile hazırlanan SQL sorgusunda ? işareti veya bir :isim ile belirtilen parametreleri değişkenlere bağlar.