Veri Tabanından Bağımsız PHP Kodlamak: ADOdb

Sibel Kuzgun Akın
Yakamoz Yazılım

İçerik

- Ben kimim?
- ADOdb nedir?
- Veri tabanından bağımsız PHP kodu yazmak neden önemli?
- MySQL ile kullanım
- Oracle ile kullanım
- Güçlü ve zayıf yanları
- Bir Alternatif: PHP PDO
- Örnek kodlar, sorular, yanıtlar, yorumlar...

Sibel Kuzgun Akın - Eğitim

- 1994 – ODTÜ Bilgisayar Mühendisliği
  - Lisans
- 1998 - ODTÜ Bilgisayar Mühendisliği
  - Yüksek lisans
- 2018'den bu yana Bursa Uludağ Üniversitesi Ekonometri Bölümünde doktora
  - Tez aşamasındayım.

Sibel Kuzgun Akın - Deneyim

- 1994 – 1996: ODTÜ Bilgi İşlem – webmaster
- 1996 – 2000: TOFAŞ – yazılımcı
- 2000 – bugüne: Yakamoz Yazılım – kurucu ve yönetici ortak
- Yarı zamanlı üniversitelerde ders verme
  - Bahçeşehir Üniversitesi
  - Bursa Uludağ Üniversitesi
  - Bursa Teknik Üniversitesi

Yakamoz Yazılım

yakamoz.net
Yazılım ve Bilgisayar Sistemleri

2000 yılında şahıs firması olarak kuruldu.
2004 yılında Hacettepe Teknokent'e geçiş ve Limited Şirkete dönüşüm
2008 yılında Ulutek Teknokent'e geçiş

Ağırlıklı olarak Otomotiv yan sanayi firmalarına yönelik özel yazılım hizmetleri esas faaliyet alanımızı oluşturmaktadır. Özellikle sac kalıpçıları ve otomotiv tedarik zincirinde olan firmalar için sektöre özel veya firmaya özel geliştirdiğimiz uygulamalar bulunmaktadır.
Açık kaynak kodlu ve web tabanlı uygulamalar geliştirmeyi tercih ediyoruz.

Giriş

- 2000'lerin başında ADOdb ile tanıştım.
- PHP – MySQL kullanıyordum.
  - Kurufasulye – Pilav veya Döner – Ayran gibi ayrılmaz ikili miydi???
- Postnuke içerik yönetim sistemi ADOdb kullanıyordu.
- Kod üreticisini yazarken, veri tabanı ile PHP kodlarının bağımsız olması gerektiğini düşündüm.
- Taşınabilirlik (Portability): yazılım kalite göstergesi
- MySQL, Oracle ve MS SQL ile PHP (ADOdb) kullanmayı denedim.

Taşınabilirlik - Portability

- Taşınabilirlik bir yazılım kalite göstergesidir.
- Projenizin veri tabanını değiştirmeseniz bile bir kısım kodu alıp, örneğin MySQL yerine PostgreSQL ile kullanmak isteyebilirsiniz.
- PHP'nin kendi veri tabanı kütüphanelerini kullanırsanız (örnek: MySQLi), veri tabanını değiştirmek istediğinizde komutlar değişiyor. Bu nedenle kodlar taşınabilir olmuyor.

Yazılım Kütüphaneleri

- Uzun soluklu, yıllara yayılan projelerde kütüphane bağımlılıkları dikkatle düşünülmelidir.
- Sizin programınız yaşarken, bağımlı olduğunuz kütüphane ölebilir.
- PHP sürüm yükseltmelerinde zorlayıcı olabilir.
  - Örnek: PHPOffice/PHPExcel → PHPOffice/PHPSpreadSheet
- Kütüphanenin yeni sürümü ile sizin kullandığınız PHP sürümü uyumsuz olabilir.
- Gerçekten gerekmiyorsa kütüphane kullanmakta çekimser davranırım. Ama ADOdb gerçekten gerekliymiş...

MySQL → MySQLi

- MySQL eklentisi:
  - PHP 5.5 ile birlikte artık kullanılmayacağı duyuruldu.
  - PHP 7 ile birlikte emekli oldu.
- MySQLi: Improved MySQL extension
- MySQLi PHP içinden MySQL, Percona Server ve MariaDB kullanmak için dilin kendisine ait standart eklentidir.
- ADOdb kullandığımız için biz bu geçişte yalnızca ayar dosyalarını değiştirdik.
- Alternatif: PHP Data Objects (PDO)

Örnek MySQL Eklentisi Kodu

```
function abone_turu_listesi ($p_degisken , $p_cins) {
      $sql = "SELECT * FROM abone_turu ORDER BY kod";
      $cins_sorgu = mysql_query ($sql);
      printf ("<SELECT NAME=\"%s\">\n" , $p_degisken);
      echo "<option value=\" \">\n";
      while ($cins = mysql_fetch_array($cins_sorgu)) {
        $kod = $cins ["kod"];
        $aciklama = $cins ["aciklama"];
        printf ("<OPTION VALUE=\"%s\" " , $kod);
        if ($kod == $p_cins) printf (" SELECTED>\n");
        else printf (">\n");
        printf ("%s\n" , $aciklama);
      }
      echo "</SELECT> \n";
}
```

Örnek MySQLi Eklentisi Kodu

```
function abone_turu_listesi ($p_degisken , $p_cins) {
     global $connection;
     $sql = "SELECT * FROM abone_turu ORDER BY kod";
     $cins_sorgu = mysqli_query ($connection, $sql);
     printf ("<SELECT NAME=\"%s\">\n" , $p_degisken);
     echo "<option value=\" \">\n";
     while ($cins = mysqli_fetch_array($cins_sorgu)){
          $kod = $cins ["kod"];
          $aciklama = $cins ["aciklama"];
          printf ("<OPTION VALUE=\"%s\" " , $kod);
          if ($kod == $p_cins)
            printf (" SELECTED>\n");
          else
            printf (">\n");
          printf ("%s\n" , $aciklama);
     }
     echo "</SELECT> \n";
}
```

Örnek ADOdb Kodu

```
function abone_turu_listesi ($p_degisken , $p_cins) {
     global $DB; // connection
     printf ("<SELECT NAME=\"%s\">\n" , $p_degisken);
     echo "<option value=\" \">\n";
     $sql = "SELECT * FROM abone_turu ORDER BY kod";
     $cins_sorgu = $DB->Execute ($sql);
     if ($cins_sorgu) {
       while (!$cins_sorgu->EOF ()){
          $kod = $cins_sorgu->fields ["kod"];
          $aciklama = $cins_sorgu->fields["aciklama"];
          printf ("<OPTION VALUE=\"%s\" " , $kod);
          if ($kod == $p_cins)
            printf (" SELECTED>\n");
          else
            printf (">\n");
          printf ("%s\n" , $aciklama);
       }
     }
     echo "</SELECT> \n";
}
```

Örnek ADOdb Bağlantı Kodu

```
$DB = NewADOConnection("mysqli");

$DB->Connect("localhost", "yakamoz",
"xxx", "yakamoz_db");

$DB->SetFetchMode(ADODB_FETCH_ASSOC);
```

Örnek ADodb Sorgu Kodu

```
$sql = "SELECT * FROM personel";

$sorgu = $DB->Execute ($sql);

if ($sorgu) {

   while (!$sorgu->EOF()) {

        printf("%d: %s %s<br />\n",
$sorgu->fields["sicil_no"],
$sorgu->fields["ad"],
$sorgu->fields["soyad"];

        $sorgu->MoveNext();

   }

}
```

Oracle → OCI8

- Oracle 7 ve öncesi için PHP Oracle eklentisi kullanılıyordu.
- Oracle 8 ile birlikte oci8 eklentisi geliştirildi.
- ADOdb kullananlar bu geçişten en az etkilendi.

OCI8 ile Sorgu

```
$conn = OCILogon("scott","tiger", $tnsName);

$stmt = OCIParse($conn,"select * from emp where empno
> :emp order by empno");

$emp = 7900;

OCIBindByName($stmt, ':emp', $emp);

$ok = OCIExecute($stmt);

while (OCIFetchInto($stmt,$arr)) {

  print_r($arr);

  echo "<hr>";

}
```

ADOdb ile Sorgu

```
include "/path/to/adodb.inc.php";

$db = NewADOConnection("oci8");
$db->Connect($tnsName, "scott", "tiger");

$rs = $db->Execute("select * from emp where
empno>:emp order by empno",
                        array('emp' => 7900));
while ($arr = $rs->FetchRow())
{
    print_r($arr);
    echo "<hr>";
}
```

ADOdb'ye Bir Alternatif: PHP PDO: PHP Data Objects

11+ veri tabanını destekliyor:

- CUBRID
- Free TDS / MS SQL Server / Sybase
- Firebird
- IBM DB2
- Informix
- MySQL
- MS SQL Server / SQL Azure
- Oracle
- ODBC
- PostgreSQL
- SQLite

PHP PDO: PHP Data Objects

ADOdb ile PHP PDO aynı işi yaparlar:

(+) PHP 5.1'den itibaren PHP PDO eklentisi dilin bir parçasıdır. ADOdb ise ayrıca kurulması gereken bir kütüphanedir.

(+) PHP PDO da hafif ve tutarlı bir katmandır.

(-) PDO MySQLi'den biraz daha yavaş çalışıyor.

(-) ADOdb'nin desteklediği veri tabanı sayısı daha fazla.

Veri Tabanı Değiştiğinde Pratikte Çıkan Sorunlar

- Bir yazılım projesinin ortasında veri tabanını Oracle'dan MySQL'e çevirdik.
- ADOdb ile PHP kodlarını değiştirmek gerekmedi. Yalnızca ayar dosyasını değiştirmek yeterli oldu.
- Oracle'daki tabloları MySQL'de baştan yarattık.
- Ancak Oracle tablo ve kolon adlarını “HEPSİ BÜYÜK” kullanıyor. MySQL ise “küçük harfleri” tercih ediyor.
- Programda kolon adlarının geçtiği SQL komutlarını elle değiştirmek gerekti.
- Tablo adlarının büyük harfle yazılması sorun olmadı.

Veri Tabanı Değiştiğinde Pratikte Çıkan Sorunlar

- Veri tabanına ait standart olmayan özellikler kullanıldı mı?
- Saklı yordamlar (stored procedures):
  - Hepsi baştan yazılmalıdır.
- Oracle sekansları (sequence) ile MySQL auto_increment kolonları
  - ADOdb: Insert_Id () ile buna erişiliyor.
- Veri bütünlüğü problemleri:
  - MySQL'deki ON UPDATE CASCADE komutu Oracle'da yok.
- Arka planda veriyi diskte saklamak ile ilgili farklılıklar:
  - Depolama motoru, bölümleme (partition), vb.

Sonuç

- Uzun yıllar kullanılacak bir uygulamada, uygulamanın değişen teknolojilere ayak uydurabilmesi için bağımlılıklarını zayıf tutmak önemlidir.
- Programlama dili, veri tabanı, yazılım kütüphaneleri seçimi yıllar içinde sorunlara yol açabilir.
- Veri tabanından – en azından teorik olarak – bağımsız kodlanan bir uygulama, ihtiyaç durumunda veya kütüphaneler değiştiğinde daha kolay uyarlanır.
- Uygulamanın tamamı olmasa bile bir kısım kodları daha kolay taşınabilir.
- ADOdb hafif ve hızlı bir katman olarak bu amaca başarıyla hizmet eden bir yazılım kütüphanesidir.
- Yine de saklı yordamlar, otomatik artan kolonlar, depolama motorları gibi veri tabanına özel SQL kodlarını elden geçirmek gerekebilir.

Kaynakça

- ADOdb: Database Abstraction Layer for PHP [ADOdb]
- ADOdb & Oracle [ADOdb]
  https://adodb.org/dokuwiki/doku.php?id=v5:userguide:oracle_tutorial
- PHP: Intoduction – Manual
  https://www.php.net/manual/en/intro.pdo.php
- What are the differences between ADOdb and PDO in PHP? - Stack Overflow
  https://stackoverflow.com/questions/1943051/what-are-the-differences-between-adodb-and-pdo-in-php