

Php hata mesajlarının önemi ve anlamı

Merhabalar arkadaşlar,

Bu yazımda size çoğu zaman önemsemediğimiz php mesajları hakkında bilgi vermeye çalışıcam. Php'de bildiğiniz gibi temel olarak 3 mesaj tipi var Notice, Warning, Error. Öncelikle bunları kısaca tanımlayalım.

Php'de kod yazarken yazdığınız kodla ilgili bir problem olduğu durumlarda bu mesaj başlıklarından biri ekranda görünür ve hata hakkında detay iletir.

Notice: Yazdığınız kodda bir aksilik olduğunu ama bu aksiliğin sistemi etkileyecek bir durum olmadığını belirtir.

Warning: Yazdığınız kodda bir hata olduğunu ve şu anda bir sıkıntı yaratmıyor olsa da ileride bir problem yaratabileceğini belirtir.

Error: Bu ise kodun zaten çalışmasına izin vermez ve hatalı olarak belirtir.

Bu genelde php'ye giriş seviyesindeki yazılımcıların dahi bildiği temel bilgilendirme mesajlarıdır bizim burada bugün değineceğimiz konu ise eldeki kaynakların verimli kullanımı. Burada bahsetmeye çalıştığım nedir ?

Php'de kod yazıp müşterinin veya kendi yazılımınızın çalışması sizi tam anlamıyla bir yazılımcı yapmaz. **Yazılımcıdan beklenen:** elindeki minimum kaynağı, maksimum performans ve güvenlikle çalışır hale getirmesidir. Buradan ne anlamamız gerekiyor evet error_reporting v.b. fonksiyonlar yardımıyla hata mesajlarının kullanıcılara gözükmesini engelleyebilirsiniz peki bu sizin sisteminizde bir sıkıntı yaratır mı ? Hatalı ve hatasız kod arasında hız, performans farkı var mı ? Önemli olan nokta bu ve benim de değineğim nokta burası.

Öncelikle çalışma platformumuzu oluşturup, örnek bir hatalı kod yazalım.

```
<?php<br /> $start = microtime(true);<br /> $i=0;<br /> $loop = true;<br /> while($loop){<br /> $id = $_GET["id"];<br /> $id = $i+$id;<br /> $i++;<br /> echo $i."<br>";<br /> if($i==50)<br /> $loop=false;<br /> }<br /> $end = microtime(true);<br /> echo "Süre: ". number_format(1000 * ($end - $start),4)." ms.";<br /> ?>
```

Burada hatalı yazdığımız kod satırı 6. satırda yer alan \$_GET["id"] kısmı. Get sorgusuyla id parametresinde herhangi bir değer gelmediği durumda id değişkenine herhangi bir değer atanamayacağı için ekranda "Notice: Use of undefined constant id" şeklinde bir hata mesajı görücez. Bu

hata mesajını `error_reporting(0);` diyerek gizleyebiliyoruz ama amacım daha önce de belirttiğim gibi bu hata mesajının bize hız olarak kaybı ne oluyor bunu bulmak olduğundan dolayı hata mesajını ilk testimizde kapatmıcaz.

Yukarıdaki kod blogunu kendi bilgisayarımda çalıştırdığımda tam olarak 367.4159 ms'de işlemi tamamlıyor. Şimdi ilgili hatayı düzeltelim ve `get` sorgusunda parametreyi gönderip veya `$id = $_GET["id"];` satırını `$id = 0;` şeklinde deneyelim.

Sonuç: 0.0749 ms yani yaklaşık olarak arada 5200 katlık bir hız farkı var.

Peki hatalı kodu `error_reporting` le kapatıp denersek sonuç ? 0.2611 ms. Hataları kapatmak ve kapatmamak arasında neredeyse 3 katı fark var.

Şimdi hata sayısını 50'den 500'e çekip deneyelim.

Döngüyü 500'e tamamladığımızda;

Hatalı kod: 4,238.5762 ms

Hatasız kod: 1.2579 ms

Hatalar gizlenmiş şekilde (`error_reporting`): 2.3661 ms.

Yani sonuç olarak basit ve önemsiz dediğimiz bir uyarı mesajını kapatmamak bile kimi zaman bize tahmin ettiğimizden fazla maaliyeti oluyor. Bu yüzden benim önerim verilen hata mesajını uyarı veya hata olarak ayırmayıp, tamamen temizlemeniz saniyeler belki küçük projelerde sizi pek etkilemeyecektir fakat projeniz büyüdükçe mili saniyelerin bile sizin için önemli olduğunu; ram, cpu kullanımının ne derece önemli olduğunu göreceksiniz.

Mustafa Erdiñ ZORBA

21 Nis. 2015

info@merdincz.com

www.merdincz.com