Denne guide er oprindeligt udgivet på Eksperten.dk

COMPUTERWORLD

Sikrere håndtering af MySQL-fejl

Ofte udskrives MySQL-fejlmeddelelser helt ukritisk. Dette er en ren lækkerbidsken for hackere eller andre med destruktive hensigter. Denne guides formål er at anvise en mere hensigtsmæssig fremgangsmåde til at håndtere databasefejl.

Skrevet den **31. Mar 2012** af **olebole** I kategorien **Programmering / PHP** | ★★★☆☆

Historik:
29-02-2012: Tilføjet funktionen displayError
31-03-2012: Rettet mysql_errno til mysql_errno()

En almindelig, men skidt tilgang
Ved brug af MySQL ses alt for ofte udskrivning af evt. fejl på formen:

```
$res = mysql_query( ... ) or die (mysql_error());
```

Eller endnu værre:

```
$sql = ' ... ';
$res = mysql_query($sql) or die (mysql_error() . '<br>SQL: ' . $sql);
```

Ved brug af MySQLI med Prepared Statements ser det ofte således ud:

```
if ($stmt = $mysqli->prepare(' ... ')) {
    // Kode ...
} else {
    echo 'Der opstod en fejl: ' . $mysqli->error;
    exit();
}
```

Disse og lignende fremgangsmåder ses såmænd også i tutorials og eksempler på php.net, men de udgør ikke desto mindre en væsentlig sikkerhedsrisiko!

Der skal ikke herske tvivl om, at de informationer, udvikleren herved kan få om problemer i koden, gør fejlfinding og dermed udviklingen af applikationen væsentligt hurtigere.

Problemet er dog, at når/hvis der opstår en fejl, efter at applikationen er gået i luften, vil alle og enhver få afsløret indgående viden om databasens og applikationens opbygning. Viden, som ondsindede personer kan bruge til destruktive handlinger. Alt efter, hvor meget der i øvrigt er gjort ved sikkerheden, kan en ondsindet person endda inducere en fejl, hvorved vedkommende kan få denne eftertragtede viden serveret på et sølvfad.

"*Jamen, det sletter jeg naturligvis bare, inden applikationen går i luften.*"

Nej, det gør du så ikke! Ifølge Mr. Murphy vil du med garanti glemme at fjerne det ét eller andet sted - og overvejende sandsynligt på det mest uhensigtsmæssige sted - og på det værst tænkelige tidspunkt.

En bedre tilgang
I stedet kan du definere en konstant, som angiver, om du er i debug mode, eller i production mode - om du udvikler, eller om applikationen er gået i luften. Meget sandsynligt eksisterer der allerede en konfigurationsfil i din applikation. I den kan du skrive:

```
define('IN_DEBUG_MODE', true); // Sæt til false i production mode
```

Denne fil inkluderes øbverst i dokumentet alle steder, hvor der udskrives fejl. Forudsat, at konfigurationsfilen ligger i mappen 'inc', som ligger i roden:

```
<?php
require_once($_SERVER['DOCUMENT_ROOT'] . 'inc/config.php');
```

Nu kan fejlmeddelelser skrives på formen:

```
$sql = ' ... ';
$res = mysql_query($sql);
if (mysql_errno() > 0) {
    echo 'Der opsod en fejl.';
    if (IN_DEBUG_MODE) echo '<br>' . mysql_error() . '<br>SQL: ' . $sql;
    exit();
}
```

Eller:

```
if ($stmt = $mysqli->prepare(' ... ')) {
    // Kode ...
} else {
    echo 'Der opstod en fejl.';
    if (IN_DEBUG_MODE) echo '<br>' . $mysqli->error;
    exit();
}
```

Nu skal du 'kun' huske at sætte IN_DEBUG_MODE til false, når du lægger projektet på serveren og åbner for offentlig adgang. Så vil brugeren kun få oplyst, at der er indtrådt en fejl, men ikke noget om tabellens struktur eller variabler i SQL'en.

En bruger har i en tråd gjort opmærksom på, at han er ked af alle de nødvendige if-strukturer, dette medfører. Løsningen kunne være at oprette en funktion **displayError** til at håndtere fejlene:

```
function displayError($userError, $devError) {
    echo $userError;
    if (IN_DEBUG_MODE) echo '<br>' . $devError;
    exit();
}

if ($stmt = $mysqli->prepare(' ... ')) {
    // Kode ...
} else {
    displayError('Der opstod en fejl.', $mysqli->error);
}
```

NB: Husk under alle omstændigheder på, at al udvikling *altid* skal foregå i et lukket, lokalt miljø - eller på en server uden offentlig adgang. Almindelige brugere må *aldrig* have adgang til applikationen eller dele af den, mens der udvikles!

Denne fremgangsmåde er ikke kun aktuel i forbindelse med databasefejl. Den kan naturligvis også anvendes, hvor man bruger manuel fejlhåndtering i PHP.

"*Hvorfor en konstant? Hvorfor ikke bare en almindelig variabel?*"

Fordi en konstant er global og kan læses inde fra en funktion uden at være erklæret som global i denne.

Husk: Ingen fremgangsmåde er skudsikker - heller ikke denne! På den anden side skal du her kun huske at ændre ét sted, for at det slår igennem i hele din applikation.

Tip: Hvis din editor understøtter en eller anden form for snippets (evt. med keyboard shortcuts), kan det betale sig at oprette en snippet med fejludskrivningen. Det gør det lynhurtigt at få skrevet fejlkoden - og mindsker lysten til 'lige at lave et hurtigt hack', hvis du i situationen ikke kan huske syntaksen.

Kommentar af jakobdo d. 07. Mar 2012 | 1

Smart lille trick.

Kommentar af squazz d. 30. Mar 2012 | 2

```
mysql_errno
```

skal vel i stedet være

```
mysql_errno()
```

Jeg havde lidt problemer medd at få dette til at virke, og da jeg slog mysql_errno op fandt jeg ud af at der skulle et () på for at det virker ;)

Lille fix til dem der endnu ikke bruger MySQLi ;)

Kommentar af olebole d. 31. Mar 2012 | 3

@**sqauzz**: Tak, det er rettet *o)

Kommentar af tobrukDk d. 12. May 2012 | 4

Tror du ikke skal have skrevet i title "mysql - fejl & mysqli - Fejl"??

Kommentar af softspot d. 13. May 2012 | 5

Kunne man ikke, i stedet for en define, lave et tjek på om host-adressen er localhost, således fejl kun udskives i det lokale miljø?

Jeg er ikke PHP udvikler, men det ville vel lukke for den sidste fejlmulighed i dette setup... eller hvad?

Kommentar af jokkejensen d. 14. May 2012 | 6

man kunne måske have et ønske om at debugge på en anden server end localhost, men godt man ikke skal døje med sådanne i asp.net :)

@Ole >> få nu skiftet fra IE6 så du får stavekontrol på !! fin artikel :)

Kommentar af bredbaandmobilt d. 12. Jan 2013 | 7

Hvor er kildehenvisningerne ole

Kommentar af olebole d. 13. Jan 2013 | 8

Hvis der er kildehenvisninger, du mener, jeg har glemt, er du mere end velkommen til at gøre mig opmærksom på, hvilke. Hvis jeg finder dem relevante, kommer de naturligvis øjeblikkeligt med.

Hvilke kildehenvisninger er det specifikt, du tænker på? Hvad mangler du uddybning af?