



Denne guide er oprindeligt udgivet på Eksperten.dk

Loginsystem (med MySQL)

Dette er en guide til, hvordan man kan lave et loginsystem med php og muligvis også med sessioner og MySQL

Skrevet den **02. Feb 2009** af **wicez (nedlagt brugerprofil)** I kategorien **Programmering / PHP** |



Loginsystem med php (og mysql)

Denne guide henvender sig til personer der har en mindre eller større forståelse for PHP og den basale syntaks heri.

Det er en fordel at kende en smule til MySQL hvis man vælger at følge hele denne guide.

Dette er dog ikke en nødvendighed for at forstå det grundlæggende omkring opbygningen af et simpelt loginsystem i PHP.

Vi vil starte med at opbygge formularen i en ny html side. Heri behøver vi ikke noget php:

formular.htm

```
<html>
<head>
<title>Min loginformular</title>
</head>
<body>
<form name="min form" id="min form" method="post" action="check.php">
Brugernavn: <input type="text" name="brugernavn"><br>
Kodeord: <input type="password" name="kodeord">
<input type="submit" name="login" value="Login">
</form>
</body>
</html>
```

Vi vil ikke kigge nærmere på formularen, da vi primært vil koncentrere os om server-side delen.

check.php

Den næste side vi skal have oprettet hedder **check.php**. Dette er den første php side vi skal have kodet lidt på. Så det er bare om at få smøget ærmerne op.

Vi starter med koden:

```
<?php
//Vi starter med at defindere variabler, så det bliver nemmere at tilkomme dem senere.
```

```

$brugernavn = $_POST['brugernavn'];
$kodeord = $_POST['kodeord'];

//Tjekker om både brugernavn og kodeord er indtastet.
if(empty($brugernavn) || empty($kodeord))
{
//Sender brugeren tilbage til formular.htm ved brug af en header
header("location. formular.htm");
exit;
}
//Hvis både brugernavn og kodeord er sat går vi videre.
else
{
//Definerer det rigtige kodeord og det rigtige brugernavn.
//Vi vil senere gå videre ind i dette,
//og se på hvordan vi kan implementere MySQL ind i vores script
$rigtigt_brugernavn = 'Jens';
$rigtig_kode = 'hallo';

//Tjekker om $brugernavn er lig med det rigtige brugernavn, og om $kodeord er lig med det rigtige
kodeord
if($brugernavn == $rigtige_brugernavn && $kodeord == $rigtige_kode)
{
//Brugeren har indtastet det rigtige brugernavn og kodeord.
echo "Tillykke, du er nu logget ind.";
/*
Det er normalt her man inkluderer filer man ikke må se, hvis man ikke er logget ind.
Her er det også muligt at gemme brugerens kodeord i sessioner,
så brugeren også er logget ind når han navigerer rundt på din hjemmeside.
*/
}
else
{
//Skriver til brugeren at han har fået forkert brugernavn
echo "Det indtastede brugernavn og/eller kodeord var forkert";
echo "<br><br>";
//Laver et link som brugeren kan trykke på, for at komme tilbage til formular.htm
echo "Tryk <a href=\"formular.htm\">her</a> for at komme tilbage";
}
}
?>

```

Dette var et meget simpelt eksempel på et loginsystem lavet i php, uden brug af hverken mysql eller sessioner.

Jeg vil hermed afslutte det helt simple, og begynde at forklare om hvordan MySQL virker sammen med php, og hvordan de grundlæggende funktioner virker.

MySQL

For at starte fra en begyndelse, er MySQL en database. Der findes utrolig mange forskellige databaser, og

jeg vil ikke begynde at bedømme hvilke der er bedst - i hvert fald ikke i denne guide. Det grundlæggende i en MySQL database er tabeller. Det første man definerer, er de forskellige kolonners navne. En brugertabel som vi kunne få brug for kunne f.eks. se sådan ud:

id - brugernavn - kodeord

Hvis du bruger phpmyadmin til din database, skal deres attributter se ud som dette:

```
id: int, auto_increasement, primary
brugernavn: varchar(255)
kodeord: varchar(255)
```

Id'et er ikke nødvendigt, det er kun lavet for at kunne identificere hver bruger, idet flere brugere måske har samme brugernavn.

For at forbinde til en MySQL database med php, skal du bruge funktionen **mysql_connect()**. Denne funktion har følgende parametre:

Server, Brugernavn, Kodeord.

Hvis du selv hoster din webserver har du selv defineret brugernavn og kodeord, mens serveren vil være "localhost". Hvis du bruger en anden webserver har du sandsynligvis modtaget en e-mail med disse oplysninger.

En anden funktion du skal bruge er funktionen **mysql_select_db()** der har følgende parametre:

```
Databasenavn
Databaseforbindelse
```

De fleste har kun én database. Navnet på denne database skal du skrive som første parameter, mens der som "databaseforbindelse" skal bruges funktionen **mysql_connect()**.

I dette tilfælde vil det være smart at lave en kode som denne, når du skal forbinde til den database:

```
<?php
$link = mysql_connect("server", "brugernavn", "kodeord");
mysql_select_db("databasenavn", $link);
?>
```

Når denne kode er skrevet, vil du have adgang til databasen fra dit php-script.

Der er én funktion mere vi skal kigge på, før vi kan bygge videre på vores login-script. Denne funktion hedder **mysql_query()** (og **mysql_fetch_assoc()**) denne funktion bruges til at lave et kald til databasen, så det er muligt at trække data ud fra din database. Til dette skal bruges et andet scriptsprog der hedder SQL. Hvis du ikke kan dette "sprog" kan du bare skrive præcis det jeg skriver, selvom det måske er lidt

indlysende hvad jeg skriver, idet det meste er almindelig engelsk.

mysql_fetch_assoc() bruges til at lægge det data du får fra din mysql_query, ind i et array i php. Der findes flere lignede funktioner som mysql_fetch_object og mysql_fetch_row. Disse har jeg dog valgt at udelade fra denne guide.

En anden funktion vi får brug for er funktionen **mysql_num_rows()**. Denne funktion skal også bruge en mysql_query for at fungere. Mysql_num_rows() returnerer antallet af rækker der matcher den valgte query.

Sessioner

Sessioner er det vi skal bruge, hvis vi gerne vil gemme oplysninger fra den ene af vores sider, til andre sider. Disse sessioner er gemt i browseren, og er en slags alternativ til cookies, selvom de langt fra gør det samme.

For at fortælle browseren at "Vi skal til at bruge session" skal vi bruge funktionen session_start(). Denne funktion skal skrives øverst på enhver side, hvorpå du vil bruge sessioner. Der må ikke komme noget output til browseren før denne funktion, ikke engang <html>.

Sessions_variablen bliver gemt i arrayet (listen) \$_SESSION['navnet'] og kan meget nemt tilgås ved rigtig brug af dette array.

Implementering af MySQL og sessioner i check.php

Jeg går her ud fra, at du har en tabel der hedder "brugere" med kolonner som jeg skrev ovenfor

check.php

```
<?php
//Helt i starten starter vi sessioner, og forbinder til vores database
session_start();
$link = mysql_connect("server", "brugernavn", "kodeord");
mysql_select_db("databasenavn", $link);

//Tjekker om både brugernavn og kodeord er indtastet.
if(empty($_POST['brugernavn']) || empty($_POST['kodeord']))
{
    //Sender brugeren tilbage til formular.htm ved brug af en header
    header("location: formular.htm");
    exit;
}
//Hvis både brugernavn og kodeord er sat, går vi videre.
else
{
    /*
    Her bruger vi databasen for første gang, idet vi nu tjekker om der er
    en bruger der har det brugernavn, der er indtastet i formularen.
    Vi bruger or die(mysql_error()) til at få mysql til at udskrive en fejl,
    hvis der er noget galt med databasekaldet.
    */
    $query = mysql_query("SELECT * FROM brugere WHERE brugernavn='". $_POST['brugernavn'] ."'") or
die(mysql_error());
    //Tjekker om der er én rækker i databasen der har det indtastede brugernavn
    //Dette gør dog, at flere brugere ikke må have samme brugernavn!
    if(mysql_num_rows($query) == '1')
    {
```

```

//Laver databasekaldet om til en liset (et array) i php.
$row = mysql_fetch_array($query);
//Tjekker om kodeordet i databasen er lig med det indtastede kodeord
if($row['kodeord'] == $_POST['kodeord'])
{
//Brugeren har indtastet rigtige oplysninger, nu skal han gemmes i sessioner
$_SESSION['brugernavn'] = $_POST['brugernavn'];
//Derudover skal han sendes til en hemlig side, hvorpå der bliver tjekket om han er logget ind
header("location: hemligside.php");
}
else
{
//Brugeren har skrevet et forkert kodeord
//Sender brugeren tilbage til formular.htm ved brug af en header
header("location. formular.htm");
exit;
}
}
//Brugernavnet fandtes ikke i databasen, derfor skal der udskrives en fejl.
else
{
//Brugeren har skrevet et forkert brugernavn
//Sender brugeren tilbage til formular.htm ved brug af en header
header("location. formular.htm");
exit;
}
}
?>

```

For at tjekke om det virkede, kan vi lave hemligside.php sådan her:

Hemligside.php

```

<?
//Starter endnu engang sessionerne øverst på siden, samt forbinder til databasen
session_start();
$link = mysql_connect("server", "brugernavn", "kodeord");
mysql_select_db("databasenavn", $link);

//Tjekker om man er logget ind
if(!empty($_SESSION['brugernavn']))
{
//Laver en query der finder ens brugerid i databasen
$query = mysql_query("SELECT * FROM brugere WHERE brugernavn='". $_SESSION['brugernavn']
.'" or die(mysql_error());
$row = mysql_fetch_assoc($query);
//Man er logget ind, og kan derfor se det hemlige indhold
echo "Dit brugerid er: ". $row['id'];

```

```
}  
else  
{  
  //Brugeren er ikke logget ind  
  echo "Du er ikke logget ind. Du skal ikke prøve at snyde på min hjemmeside!";  
}  
  
?>
```

Alle 3 filer skal ligge i samme mappe på din server for at scriptet skal fungerer.

Jeg håber i kunne bruge denne guide til noget. Den er, som nævnt i starten, ment til nybegyndere indenfor php, der er kendte med den basale syntaks heri.

Hvis i har spørgsmål e.l. kan i sende mig en e-mail på abb@cykeltjek.dk eller prøve at fange mig herinde på exp.dk

Jeg har ændret nogle småting i dag, da jeg har fået at vide at det er en dårlig måde at gøre det på :o) (13/4-2005).

Edit: Jeg havde glemt en `<input type="submit">` i starten af artiklen. Tak venchil

Edit: Jeg har gjort artiklen gratis, selvom jeg nu ikke kan se hvor mange der har læst den :|
Hvis i vil skrive en kommentar vil jeg gerne have at i husker, at artiklen er ment til nybegyndere, og at forskellige regular expressions er udeladte.

- Anders Balmer

Kommentar af mcookie d. 05. Jun 2005 | 1

Der er formodentligt væsentligt smartere måder at gøre det på.....

Der er noget med at man kan lave nogle bandittricks med dine forespørgelser i forhold til at du ikke kontrollere hvad bruger indtaster af brugernavn.....

Kommentar af philip d. 27. May 2005 | 2

kluntet kode, kluntet metode :)

Kommentar af john_stigers (nedlagt brugerprofil) d. 19. Jan 2008 | 3

Kunne ikke bruge det til noget - ikke forklaret særligt godt.

Desuden er det vist en guide og ikke en artikel...

Kommentar af cypermann d. 11. Apr 2005 | 4

En fin artile. Men er det ikke overkill at lave et sql udtræk hver gang en side loader, for at se om brugeren er logget ind. Nu du alligevel har sat en session. God begynder guide, men meget kunne optimeres.

Kommentar af qtax87 (nedlagt brugerprofil) d. 13. Apr 2005 | 5

Jeg havde nu:

```
$query = mysql_query("SELECT * FROM brugere WHERE brugernavn='". $_POST['brugernavn'] .'" AND  
kodeord='". $_POST['kodeord'] .'"") or die(mysql_error());
```

Det kan den lige så godt, og så tjekke om der findes nogen bruger med det indtastede...

Kommentar af wickedd d. 11. Apr 2005 | 6

Tja .. Ikke en særlig god måde at gøre det på

Kommentar af arkanoid d. 28. Jun 2005 | 7

Tja.... en lidt kompliceret/usmart metode... og desuden bør sessioner krypteres

Kommentar af venchil d. 17. Apr 2005 | 8

Det er en utrolig god artikel, som jeg kunne bruge til utrolig meget. Men mangler der ikke en "login"-button, som man kan trykke på i det første stykke kode?

Kommentar af klubba d. 28. Apr 2005 | 9

Super artikel!

Kommentar af hyberpreprocessor d. 11. Apr 2005 | 10

god guide, du burde nok lave noget input validering før du henter info i databasen, bare addslashes() rigeligt, men en simpel reg.exp ville også være godt (hov, havde lige overset sidste linje hehe) :)

Kommentar af pvtsummer d. 07. Apr 2006 | 11

Jeg mener ikke denne er god for begyndere. De skal ikke lære at det er så kluntet at man på alle sider skal have 50 linjers kode, som egentlig ikke har noget med indholdet at gøre. ;) - Men koden er da velskrevet i bund og grund, så du får middel.

Kommentar af frail_vex d. 12. Mar 2006 | 12

Kommentar af lassemelbye d. 07. Jan 2007 | 13

God artikel, som er værd at læse hvis man skal lave et loginsystem. Synes godt du kunne have lavet nogle indrykninger i dine eksempler, men godt at du har fået massere at kommentarer på

Kommentar af phil-profil d. 24. Sep 2006 | 14

sygt god

Kommentar af ohhelpme d. 10. Aug 2008 | 15

hvorfor er der 2 forskellige check.php'er