



# Pagini WEB cu PHP 4

Mihai Scorțaru, Claudiu Soroiu

În cadrul acestui articol vom continua prezentarea funcțiilor utilizate pentru accesarea bazelor de date MySQL și vom prezenta, de asemenea, modul de utilizare a poștei electronice folosind funcții oferite de limbajul PHP.

## Baze de date MySQL

În numărul anterior am prezentat funcțiile de acces la bazele de date MySQL până la funcția `mysql_info`. În cele ce urmează vom continua prezentarea altor funcții utile în lucrul cu baze de date MySQL.

### Funcția `mysql_affected_rows`

În cazul interogărilor în urma cărora nu se obține nici un rezultat, se poate folosi funcția `mysql_affected_rows` pentru a verifica numărul de înregistrări care au fost actualizate la ultima interogare a bazei de date, pentru celelalte tipuri de interogări putându-se folosi funcția `mysql_num_rows`.

Funcția `mysql_affected_rows` are un singur parametru care reprezintă identificatorul de acces al unei conexiuni spre un server MySQL și returnează un număr întreg, care reprezintă numărul de înregistrări actualizate la ultima interogare a bazei de date.

În continuare prezentăm un exemplu de utilizare a acestei funcții, iar în figura 1 poate fi observat efectul execuției *script*-ului:



Figura 1: Funcția `mysql_affected_rows`

```
<HTML>
<HEAD>
  <TITLE>
    Exemplu mysql_affected_rows
  </TITLE>
</HEAD>
<BODY>
<?PHP
  $conn = mysql_connect("blue", "ginfo",
                        "pginfo");

  if ($conn === FALSE)
```

```
    die("Conectare eșuată.");
  mysql_select_db("ginfo", $conn);
  mysql_query("update utilizatori set nume
    = 'soroiu' where nume = 'Soroiu'", $conn);
  $s = mysql_affected_rows($conn);
  echo "Număr de înregistrări actualizate: $s";
  mysql_close($conn);
?>
</BODY>
</HTML>
```

### Funcția `mysql_fetch_array`

Această funcție transformă o înregistrare dintr-un rezultat primit de la server-ul MySQL într-o listă.

Funcția `mysql_fetch_array` colectează datele referitoare la înregistrarea curentă și le returnează sub forma unei liste ale cărei elemente pot fi accesate folosind numele câmpurilor, numărul lor de ordine sau folosind ambele moduri.

Dacă nu mai sunt înregistrări de prelucrat, funcția returnează valoarea logică **FALSE**.

Această funcție are doi parametri:

- primul parametru este de tipul resursă, a cărui valoare a fost setată folosind una dintre funcțiile `mysql_query` sau `mysql_unbuffered_query`;
- al doilea parametru este opțional, este de tip întreg și reprezintă modul în care se va face transformarea.

Cel de-al doilea parametru poate avea una din valorile următoare:

- `MYSQL_ASSOC` - elementele din lista rezultată pot fi accesate folosind doar numele câmpurilor;
- `MYSQL_NUM` - elementele din lista rezultată pot fi accesate folosind doar numerele de ordine ale câmpurilor;
- `MYSQL_BOTH` - elementele din lista rezultată pot fi accesate folosind ambele moduri prezentate anterior;



### Funcția mysql\_fetch\_assoc

Această funcție transformă o înregistrare dintr-un rezultat primit de la *server*-ul *MySQL* într-o listă și are același efect cu cel al folosirii funcției *mysql\_fetch\_array*, folosind pentru cel de-al doilea parametru valoarea *MYSQL\_ASSOC*.

### Funcția mysql\_fetch\_num

Această funcție transformă o înregistrare dintr-un rezultat primit de la *server*-ul *MySQL* într-o listă și are același efect cu cel al folosirii funcției *mysql\_fetch\_array*, folosind pentru cel de-al doilea parametru valoarea *MYSQL\_NUM*.

### Funcția mysql\_stat

Această funcție returnează un șir de caractere care reprezintă statusul *server*-ului *MySQL* și are un singur parametru care reprezintă identificatorul de acces la o conexiune către un *server MySQL*.

În continuare prezentăm un exemplu de utilizare a acestei funcții, iar în figura 2 poate fi observat efectul execuției acestui *script*:

```
<HTML>
<HEAD>
  <TITLE>
    Exemplu mysql_stat
  </TITLE>
</HEAD>
<BODY>
<PRE>
<?PHP
    $conn = mysql_connect("blue","ginfo",
                        "pginfo");

    $status = explode(' ',mysql_stat($conn));
    print_r($status);
    mysql_close($conn);
?>
</PRE>
</BODY>
</HTML>
```

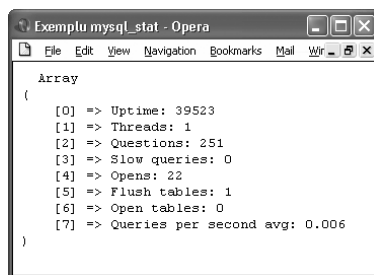


Figura 2: Funcția *mysql\_stat*

Setul de funcții de acces la bazele de date *MySQL* oferit de limbajul *PHP* este vast și în cadrul acestei serii de articole ne-am rezumat la prezentarea celor mai importante astfel de funcții. Cu ajutorul acestor funcții se poate admi-

nistra și utiliza foarte ușor orice bază de date *MySQL* indiferent de complexitatea și dimensiunea lor.

### Poșta electronică

Faptul care a dus la utilizarea pe scară largă a acestui limbaj de *script*-ing se datorează multitudinii de funcții pe care acesta le oferă.

Biblioteca oferită de limbajul *PHP* conține și funcții utile pentru transmiterea și prelucrarea mesajelor de poșta electronică.

Limbajul *PHP* are integrat un client pentru protocolul *SMTP* (*Simple Mail Transfer Protocol*) și un set de funcții necesare pentru prelucrarea mesajelor primite de către un *server de mail* în vederea utilizării datelor conținute.

Cu ajutorul acestor funcții se pot realiza aplicații interesante cum sunt administrarea listelor de discuții, transmiterea unor știri tuturor utilizatorilor înscrși pe o anumită listă, difuzarea de noutăți cu privire la diverse apariții etc.

În continuare vă prezentăm funcția *mail*, cea mai importantă funcție pe care limbajul *PHP* o oferă pentru transmiterea mesajelor de poșta electronică.

### Funcția mail

Această funcție se folosește pentru a transmite un mesaj de poșta electronică la una sau mai multe căsuțe poștale.

Funcția *mail* are cinci parametri de tip șir de caractere, dintre care ultimii doi sunt opționali.

Primul parametru reprezintă o listă a tuturor căsuțelor poștale, separate prin caracterul ",", care reprezintă destinatarul mesajului care se va transmite. O căsuță poștală din lista destinatarilor poate fi specificată utilizând unul dintre următoarele formate "nume\_destinatar <adresă\_email>" (fără ghilimele) sau "adresă\_email" (fără ghilimele), unde *nume\_destinatar* reprezintă numele proprietarului căsuței poștale *adresă\_email* și poate fi și șirul vid, iar *adresă\_email* reprezintă adresa căsuței poștale și are formatul "nume\_utilizator@domeniu" (fără ghilimele), unde *domeniu* reprezintă numele *server*-ului care găzduiește căsuța poștală a utilizatorului *nume\_utilizator*.

În cazul utilizării primului format, pentru a specifica un destinatar pentru mesajul care se va transmite, este obligatorie folosirea caracterelor "<" și ">" înainte, respectiv după câmpul *adresă\_email*, iar dacă valoarea câmpului *nume\_destinatar* conține caracterul spațiu este necesară utilizarea caracterului "\"" înaintea acestuia și după acesta. De exemplu, adresa "Claudiu Soroiu" <csoroiu@yahoo.com>" este o destinație validă pentru un mesaj de poșta electronică.

Al doilea parametru reprezintă subiectul mesajului care se va transmite.

Cel de-al treilea parametru reprezintă conținutul mesajului care va fi transmis.

Pentru a putea trimite mesaje care conțin atașamente sau care au un tip de conținut altul decât text (ex. *HTML*)



este necesar ca valoarea acestui parametru să fie în format *MIME (Multi-purpose Internet Mail Extension)*.

Cel de-al patrulea parametru reprezintă o listă de linii de antet suplimentare, separate prin șirul de caractere "\r\n" (sfârșit de linie), pentru mesajul care va fi transmis.

Ultimul parametru reprezintă o listă de opțiuni suplimentare pentru aplicația care va fi folosită pentru a efectua trimiterea mesajului și este dependent de aceasta.

Funcția returnează valoarea logică **TRUE** dacă s-a reușit trimiterea mesajului și valoarea logică **FALSE** în caz contrar.

În continuare vă prezentăm un exemplu de utilizare a acestei funcții.

```
<?PHP
    if (isset($_POST["trimite"])) {
        $a = mail($_POST["email"],
                $_POST["subject"], $_POST["mesaj"]);
        if ($a)
            include "mesajtrimis.html";
        else
            include "mesajnetrimis.html";
    }
    else
        include "formular.html";
?>
```

Script-ul anterior realizează transmiterea unui *e-mail* pe baza datelor dintr-un formular *HTML*. În momentul încărcării script-ului, acesta afișează pagina "formular.html" care reprezintă un formular similar cu cel din figura 3.

Codul corespunzător formularului din fișierul "formular.html" este prezentat în continuare:

```
<HTML>
<HEAD>
<TITLE>
    Trimitere mesaj
</TITLE>
</HEAD>
<BODY>
<FORM method = "post" action =
    "mailsender.php">

    Subiect:<BR>
<INPUT type = "text" name = "subject">
<P>
    Mesaj:<BR>
<TEXTAREA cols = "40" rows = "10" name =
    "mesaj">

</TEXTAREA>
<P>
    Adresa de mail:<BR>
<INPUT type = "text" name = "email">
<P>
```

```
<INPUT type = "submit" name = "trimite"
    value = "Trimite">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

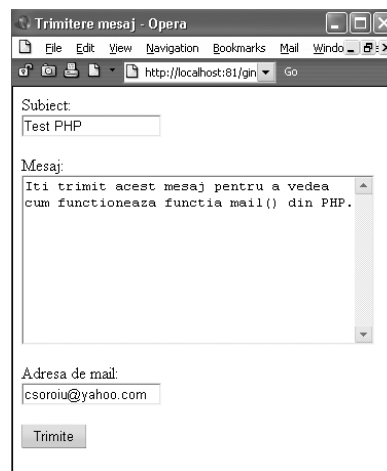


Figura 3: Formular de trimitere a unui mesaj

În momentul apăsării butonului "Trimite", script-ul încearcă trimiterea mesajului cu subiectul "Test PHP" și conținutul specifică în cadrul câmpului mesaj, la adresa csoroiu@yahoo.com.

Se observă că am folosit al doilea format prezentat pentru a specifica destinatarul mesajului.

Dacă se reușește trimiterea mesajului, atunci script-ul afișează pagina "mesajtrimis.html" după cum se poate observa în figura 4, iar în continuare se află codul *HTML* pentru această pagină:

```
<HTML>
<HEAD>
<TITLE>
    Mesaj trimis
</TITLE>
</HEAD>
<BODY>
<FONT color = "navy" size = "+1">
    <B>Mesajul a fost trimis cu succes.</B>
</FONT>
</BODY>
</HTML>
```



Figura 4: Confirmare trimitere mesaj

În cazul în care nu se reușește trimiterea mesajului, atunci script-ul afișează pagina "mesajnetrimis.html"

după cum se poate observa în figura 5, iar în continuare se află codul *HTML* pentru această pagină:

```
<HTML>
<HEAD>
  <TITLE>
    Eroare trimitere mesaj
  </TITLE>
</HEAD>
<BODY>
  <FONT color = "red" size = "+1">
    <B>Mesajul nu a putut fi trimis.</B>
  </FONT>
</BODY>
</HTML>
```



Figura 5: Eroare trimitere mesaj

## PEAR

*PEAR (PHP Extension and Application Repository)* reprezintă un proiect demarat de echipa de programatori *PHP*. Scopul acestui proiect este de a oferi o bibliotecă structurată de clase *open-source* tuturor utilizatorilor limbajului *PHP*.

Această bibliotecă oferă, printre multe altele, un set de clase care pot fi folosite pentru a putea transmite mesaje electronice care să conțină atașamente și al căror conținut poate fi în format *HTML* în loc de text.

Acest set de clase poate fi descărcat de la adresa *web* [http://pear.php.net/Mail\\_Mime](http://pear.php.net/Mail_Mime). Pentru a putea folosi setul de clase *Mail\_Mime* trebuie instalat mai întâi pachetul *PEAR* de bază care poate fi găsit la adresa <http://pear.php.net>.

În continuare vom prezenta clasa *Mail\_Mime*, cea mai importantă clasă necesară pentru a codifica în format *MIME* astfel de mesaje.

### Clasa *Mail\_mime*

Această clasă se folosește pentru a construi mesaje electronice în format *MIME*.

## Puțină configurare PHP...

Pentru a putea utiliza funcțiile oferite de limbajul *PHP* pentru a trimite mesaje de poștă electronică, pe sistemele de operare din familia *Windows* sau *\*NIX* este necesar să se seteze corect interpretorul *PHP* prin intermediul fișierului *php.ini* localizat în directorul în care este instalat sistemul de operare, respectiv în directorul */etc* pe sisteme *\*NIX*.

Funcția *mail* trimite mesajele prin intermediul unui server *SMTP* (Simple Mail Transfer Protocol) care poate fi găzduit de server-ul pe care este instalat interpretorul *PHP* sau de către un alt server accesibil interpretorului.

În fișierul *php.ini* trebuie adăugate variabilele având numele *SMTP*, *smtp\_port*, *sendmail\_from* și *sendmail\_path*, dacă acestea nu există, sau trebuie modificate valorile acestora.

Variabila *SMTP* reprezintă ip-ul sau adresa server-ului de mesaje electronice.

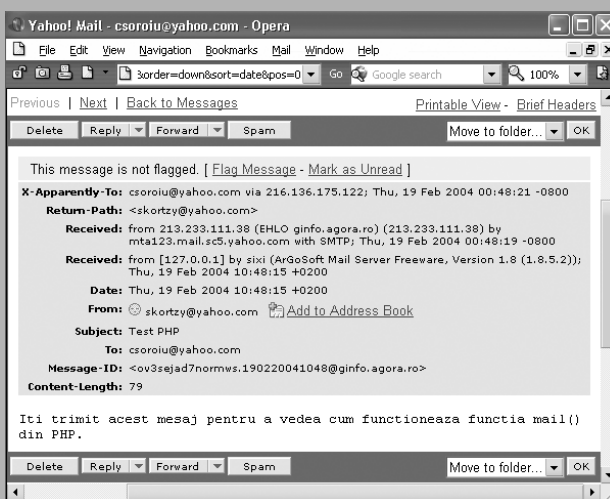
Dacă această variabilă lipsește, atunci, pentru sistemele de operare din familia *Windows* se va încerca folosirea sistemului pe care se află instalat interpretorul pentru a transmite mesaje electronice, iar pentru sistemele de operare din familia *\*NIX* se vor folosi setările implicite ale sistemului.

Variabila *smtp\_port* reprezintă portul pe care răspunde server-ul de mesaje electronice și are valoarea implicită 25, pentru sistemele de operare din familia *Windows* sau setarea im-

plicită a sistemului, pentru sistemele de operare din familia *\*NIX*.

Variabila *sendmail\_from* reprezintă adresa de mail a expeditorului. După cum se observă în imaginea următoare am folosit adresa *skortzy@yahoo.com*. Valoarea acestei variabile este folosită numai pentru sistemele de operare din familia *Windows*, iar pe sistemele de operare din familia *\*NIX* se folosește setarea implicită a sistemului.

Variabila *sendmail\_path* reprezintă calea către aplicația care se ocupă de transmiterea mesajelor pe sistemele de operare din familia *\*NIX*.



Mesajul primit pe adresa *csoroiu@yahoo.com*





## Constructorul clasei

Clasa `Mail_Mime` are șase funcții membru a căror funcționalitate o vom prezenta în continuare și un constructor

Constructorul `Mail_Mime` creează o instanță a clasei și are un singur parametru de tip șir de caractere, care este opțional. Acest parametru reprezintă terminatorul de linie care va fi utilizat de către clasă pentru a încheia fiecare linie din mesaj și are valoarea implicită `"\r\n"`.

## Funcția `Mail_Mime::setTxtBody`

Această funcție setează conținutul text al unui mesaj pentru ca acesta să poată fi vizualizat în cadrul unui client de *mail* de tip text cum este cazul aplicației *mail* prezentă pe sistemele de operare din familia *\*NIX*.

Această funcție are doi parametri.

Primul parametru este de tip șir de caractere și reprezintă conținutul mesajului sau numele fișierului care conține mesajul în format text.

Cel de-al doilea parametru este de tip logic și este opțional. Dacă acest parametru are valoarea **TRUE**, atunci se va considera că primul parametru reprezintă numele unui fișier, iar dacă are valoarea **FALSE**, atunci se va considera că primul parametru reprezintă conținutul mesajului. Valoarea implicită a acestui parametru este **FALSE**.

Funcția `setTxtBody` returnează valoarea logică **TRUE** dacă a reușit să adauge la mesaj conținutul și o instanță a clasei `PEAR_Error` în caz contrar.

## Funcția `Mail_Mime::setHTMLBody`

Această funcție setează conținutul *HTML* al unui mesaj pentru ca acesta să poată fi vizualizat în cadrul unui client de *mail* de tip *HTML* cum este cazul aplicației *Outlook* prezentă pe sistemele de operare din familia *Windows*.

Această funcție are doi parametri.

Primul parametru este de tip șir de caractere și reprezintă conținutul mesajului sau numele fișierului care conține mesajul în format *HTML*.

Cel de-al doilea parametru este de tip logic și este opțional. Dacă acest parametru are valoarea **TRUE**, atunci se va considera că primul parametru reprezintă numele unui fișier, iar dacă are valoarea **FALSE**, atunci se va considera că primul parametru reprezintă conținutul mesajului. Valoarea implicită a acestui parametru este **FALSE**.

Funcția `setHTMLBody` returnează valoarea logică **TRUE** dacă a reușit să adauge la mesaj conținutul și o instanță a clasei `PEAR_Error` în caz contrar.

## Funcția `Mail_Mime::addHTMLImage`

Această funcție se folosește pentru a adăuga o imagine în corpul unui mesaj, în cazul în care conținutul în format *HTML* utilizează imagini integrate.

Această funcție are patru parametri. Primul parametru este de tip șir de caractere și reprezintă conținutul imaginii sau numele fișierului care conține imaginea.

Al doilea parametru este de tip șir de caractere, este opțional și reprezintă tipul conținutului. Acest parametru are

valoarea implicită `"application/octet-stream"`. De exemplu, dacă imaginea este de tip *GIF*, acest parametru poate primi valoarea `"image/gif"`.

Cel de-al treilea parametru este de tip șir de caractere, este opțional și reprezintă numele care va fi dat imaginii. În cazul în care primul parametru conține imaginea, atunci este necesar ca acest parametru să apară, în caz contrar se va folosi numele fișierului transmis ca valoare pentru primul parametru.

Cel de-al patrulea parametru este de tip logic și este opțional. Dacă acest parametru are valoarea **TRUE**, atunci se va considera că primul parametru reprezintă numele unui fișier, iar dacă are valoarea **FALSE**, atunci se va considera că primul parametru reprezintă conținutul unei imaginii. Valoarea implicită a acestui parametru este **TRUE**.

Funcția `addHTMLImage` returnează valoarea logică **TRUE** dacă a reușit să adauge la mesaj conținutul și o instanță a clasei `PEAR_Error` în caz contrar.

## Funcția `Mail_Mime::addAttachement`

Această funcție se folosește pentru a adăuga un fișier la un mesaj și are cinci parametri, dintre care primii patru au aceeași semnificație cu cei de la funcția precedentă.

Ultimul parametru este de tip șir de caractere, este opțional și reprezintă tipul codificării care va fi utilizată. Acest parametru are valoarea implicită `"base64"`. Pentru fișiere atașate care nu sunt binare (ex. fișiere executabile, arhive etc.) se poate folosi valoarea `"quoted-printable"`.

## Funcția `Mail_Mime::headers`

Această funcție se folosește pentru a adăuga linii de antet suplimentare la mesajul constituit de această clasă.

Funcția `headers` are un singur parametru de tip listă, care are ca și chei numele câmpurilor antetelor și ca valori, valorile acestor câmpuri.

Funcția returnează un o referință la o listă de același tip care conține, pe lângă elementele transmise ca parametru, și elementele adăugate de această clasă, pentru ca mesajul să fie recunoscut cum trebuie de către un client de *mail*.

Această funcție nu poate fi apelată după apelul funcției `Mail_Mime::get`.

## Funcția `Mail_Mime::get`

Această funcție se folosește pentru a obține conținutul mesajului electronic în format *MIME*, după ce acesta a fost construit.

Funcția `get` nu are nici un parametru și returnează un șir de caractere care reprezintă conținutul mesajului electronic codificat în format *MIME*.

Pentru a transmite un mesaj folosind funcția *mail* prezentată și această clasă, trebuie prelucrat șirul liniilor de antet returnat de funcția `Mail_Mime::headers` astfel încât să se obțină un șir de caractere, iar conținutul mesajului va fi constituit de șirul de caractere rezultat în urma apelării funcției `Mail_Mime::get`.