

# APLIKACJE INTERNETOWE

## PHP



TEMAT 5-02: PHP i MySQL.

Autor dokumentu: Wojciech Galiński

wtorek, 29 listopada 2016 r.

351203 Technik informatyk

ŹRÓDŁA WIEDZY:

<http://www.php.net/manual/pl>, <http://webmaster.helion.pl/index.php/kurs-php>, <http://pl.wikipedia.org/>,  
<http://pl.wikibooks.org/wiki/PHP>, <http://phpkurs.pl/>, <http://kursphp.com/>.

## Zagadnienia obowiązkowe

1. **Moduł „mysql” (interfejs tradycyjny)** – to przestarzały interfejs (podatny na ataki hakerów) umożliwiający komunikację z bazą danych MySQL za pomocą funkcji.

**Moduł „mysql” został uznany za przestarzały już od PHP 5.5, a od PHP 7.0 nie jest on już dostępny.**

Zatem od PHP 7.0 poniższe funkcje NIE SĄ już dostępne:

*mysql\_connect, mysql\_select\_db, mysql\_query, mysql\_fetch\_array, mysql\_free\_result, mysql\_close, mysql\_error, mysql\_insert\_id, mysql\_affected\_rows, mysql\_num\_fields, mysql\_num\_rows, mysql\_info.*

Pełna lista funkcji znajduje się pod adresem: <http://php.net/manual/en/ref.mysql.php>.

2. **Moduł „MySQLi”** – to proceduralny lub obiektowy interfejs obsługi bazy danych MySQL dostępny w PHP 5. Jest szybszy, niż PDO, ale komunikuje się jedynie z MySQL.

3. **Funkcje proceduralnego interfejsu do obsługi bazy danych za pomocą MySQLi** – pełna lista funkcji znajduje się w podręczniku pod adresem: <http://php.net/manual/en/mysqli.summary.php>.

**Interfejs proceduralny w module „mysqli” został uznany za przestarzały już od PHP 5.3, a w PHP 5.4 niektóre funkcje z tego modułu zostały usunięte.**

Oto wybrane z nich:

Proceduralny interfejs MySQLi	Opis
<b>mysqli_connect</b>	Łączy się z bazą danych i wybiera bieżącą bazę danych.
<b>mysqli_connect_errno</b>	Zwraca numer ostatniego błędu wykrytego przez „mysqli”.
<b>mysqli_connect_error</b>	Zwraca treść ostatniego błędu wykrytego przez „mysqli”.
<b>mysqli_query</b>	Wysyła zapytanie SQL do bazy danych.
<b>mysqli_fetch_assoc</b>	Odczytuje wyniki zapytania rekord po rekordzie.
<b>mysqli_free_result</b>	Zwalnia pamięć dla wyników zwróconych przez funkcję „mysqli_query”.
<b>mysqli_close</b>	Zamyka połączenie z bazą danych MySQL.

PRZYKŁAD: (prosty skrypt z użyciem interfejsu proceduralnego)

```
$polaczenie = @mysqli_connect('nazwa albo adres serwera', 'login', 'hasło', 'baza danych');  
if (mysqli_connect_errno() != 0) die('Błąd połączenia z MySQL: ' . mysqli_connect_error() . '');
```

```
$sql = 'SELECT * FROM tabela';
```

```
if (!($wynik = mysqli_query($polaczenie, $sql)))  
    die('<p>BŁĄD: problem z zapytaniem "' . $sql . '".</p>');  
else  
{  
    echo '<table>';  
    $pierwszy_rekord = true;  
    while ($rekord = mysqli_fetch_assoc($wynik))  
    {  
        if ($pierwszy_rekord)  
        {  
            $pierwszy_rekord = false;  
            echo '<tr>';  
            foreach ($rekord as $k=>$w) echo '<th>'. $k. '</th>';  
            echo "</tr>\n";  
        }  
        echo '<tr>';  
        foreach ($rekord as $w) echo '<td>'. $w. '</td>';  
        echo "</tr>\n";  
    }  
    mysqli_free_result($wynik);  
    echo '</table>';  
}  
mysqli_close($polaczenie);
```

#### 4. Funkcje obiektowego interfejsu do obsługi bazy danych za pomocą MySQLi – pełna lista funkcji znajduje się w podręczniku pod adresem: [php.net/.../mysqli.summary.php](http://php.net/.../mysqli.summary.php).

Oto wybrane funkcje interfejsu MySQLi podzielone na następujące grupy:

##### ➔ funkcje dotyczące połączenia i treści zapytania:

Obiektowy interfejs MySQLi	Opis
<b>new mysqli(...)</b>	Łączy się z bazą danych i wybiera bieżącą bazę danych.
<b>mysqli::connect_errno</b>	Zwraca numer ostatniego błędu wykrytego przez „mysqli”.
<b>mysqli::connect_error</b>	Zwraca treść ostatniego błędu wykrytego przez „mysqli”.
<b>mysqli::query</b>	Wysyła zapytanie SQL do bazy danych.
<b>mysqli::fetch_assoc</b>	Odczytuje wyniki zapytania rekord po rekordzie.
<b>mysqli::free_result</b>	Zwalnia pamięć dla wyników zwróconych przez funkcję „mysqli_query”.
<b>mysqli::close</b>	Zamyka połączenie z bazą danych MySQL.

PRZYKŁAD: (prosty skrypt z użyciem interfejsu obiektowego)

```
$mysqli = @ new mysqli('nazwa albo adres serwera', 'login', 'hasło', 'baza danych');
if ($mysqli->connect_errno) die('Błąd połączenia z MySQL: '. $mysqli->connect_error. '');
```

  

```
$sql = 'SELECT * FROM tabela';

if (!($wynik = $mysqli->query($sql))) die('<p>Błąd w zapytaniu "'. $sql. ' "</p>');
else
{
    echo '<table>';
    $pierwszy_rekord = true;
    while ($rekord = $wynik->fetch_assoc())
    {
        if ($pierwszy_rekord)
        {
            $pierwszy_rekord = false;
            echo '<tr>';    foreach ($rekord as $k=>$w) echo '<th>'. $k. '</th>';    echo '</tr>';
        }
        echo '<tr>';    foreach ($rekord as $w) echo '<td>'. $w. '</td>';    echo '</tr>';
    }
    $wynik->free_result();
    echo '</table>';
}
$mysqli->close();
```

##### ➔ funkcje dotyczące parametrów zapytania:

Obiektowy interfejs MySQLi	Opis
<b>mysqli::prepare</b>	Przygotowuje wzorec zapytania SQL, który należy wypełnić wartościami parametrów.
<b>mysqli_stmt::bind_param</b>	Przekazuje kolejny parametr wysłanego wcześniej wzorca zapytania SQL.
<b>mysqli_stmt::execute</b>	Łączy wzorec z parametrami i zostaje wykonane zapytanie SQL.
<b>mysqli_stmt::bind_result</b>	Ustawia zmienne, do których odbieramy wyniki zapytania.
<b>mysqli_stmt::fetch</b>	Pobiera kolejne rekordy tabeli, będącej wynikiem zapytania typu „SELECT”.
<b>mysqli_stmt::affected_rows</b>	Pobiera liczbę zmodyfikowanych / dodanych / usuniętych rekordów w zapytaniu typu „UPDATE” / „INSERT” / „DELETE”.
<b>mysqli_stmt::close</b>	Zwalnia zasoby związane z wynikami zapytania SQL.

Inne klasy dotyczące MySQLi to: mysqli\_result, mysqli\_driver, mysqli\_warning, mysqli\_exception.

PRZYKŁAD 2 (zastosowanie interfejsu obiektowego):

```
$mysqli = @new mysqli('serwer', 'login', 'hasło', 'baza danych');
if ($mysqli->connect_errno) die('Błąd połączenia z MySQL: '. $mysqli->connect_error. '');
```

  

```
$wiek = (isset($_GET['wiek']) && is_numeric($_GET['wiek']))? $_GET['wiek'] + 0: 0;
$wzrost = (isset($_GET['wzrost']) && is_numeric($_GET['wzrost']))? $_GET['wzrost'] + 0: 0;
$sql = 'SELECT imie, nazwisko, wiek, wzrost FROM uczen WHERE wiek >= ? AND wzrost >= ?';

if (!$wiek || !$wzrost) echo '<p>Zobacz <a href="?wiek=18&wzrost=1.6">przykład</a>.</p>';
elseif (!($fraz = $mysqli->prepare($sql))) echo 'Błąd w zapytaniu "'. $sql. ' "';
else // i - integer, d - double, s - string, b - blob
{
    $fraz->bind_param('id', $wiek, $wzrost);
    $fraz->execute();
    $fraz->bind_result($imie, $nazwisko, $wiek, $wzrost);
    $fraz->fetch();
    $fraz->close();
    printf("<p>Imię: <b>%s</b>, nazwisko: <b>%s</b>, wiek <b>%s</b>, wzrost: <b>%s</b></p>\n",
        $imie, $nazwisko, $wiek, $wzrost);
}
$mysqli->close();
```

## Zadania

1. Korzystając z powyższych hiperłączy zapoznaj się z przedstawionymi funkcjami dotyczącymi obsługi baz danych.
2. Zapoznaj się z przedstawionymi przykładami. Wykonaj je i przeanalizuj ich działanie.