

APLIKACJE INTERNETOWE

PHP



TEMAT 5-04: Tworzenie klasy do obsługi MySQL.

Autor dokumentu: Wojciech Galiński

środa, 7 września 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. Klasa „WG_PDO_Mysql” – to klasa upraszczająca podstawowe zapytania do bazy danych. W założeniach posiada ona następujące funkcje składowe (metody):

- ➔ **konstruktor** – tworzy połączenie z wybraną bazą danych w systemie MySQL;
- ➔ **sql_query** – zwraca liczbę rekordów zmodyfikowanych poprzez modyfikujące zapytanie SQL („INSERT”, „UPDATE”, „DELETE”, itp.);
- ➔ **sql_value** – zwraca pojedynczą wartość pierwszej lub określonej kolumny pierwszego rekordu zapytania SQL;
- ➔ **sql_field** – zwraca zwykłą tablicę zawierającą pierwszą lub określoną kolumnę wyników zapytania SQL;
- ➔ **sql_record** – zwraca tablicę asocjacyjnej zawierającą pierwszy rekord wyników zapytania SQL;
- ➔ **sql_table** – zwraca wyniki zapytania SQL w postaci tabeli danych.

PRZYKŁAD:

```
class WG_PDO_Mysql
{
    private $user, $database, $host, $pdo, $sql;
    public function __construct($usr, $pwd, $db, $host = 'localhost')
    {
        $this->user = $usr;
        $this->database = $db;
        $this->host = $host;
        try
        {
            $this->pdo = new PDO('mysql:host=' . $host . ';dbname=' . $db, $usr, $pwd);
        }
        catch (PDOException $e)
        {
            print "Error!: " . $e->getMessage() . "<br/>"; die();
        }
    }
    public function sql_query($sql)
    {
        $this->sql = $sql;
        $affected_rows = $this->pdo->exec($sql);
        return $affected_rows;
    }
    public function sql_value($sql, $field = null)
    {
        $this->sql = $sql;
        $statement = $this->pdo->query($sql);
        if ($statement===false) return null;
        $row = $statement->fetch();
        if (is_null($field)) return $row[0];
        return (isset($row[$field])? $row[$field]: null);
    }
    public function sql_field($sql, $field = null)
    {
        $this->sql = $sql;
        $statement = $this->pdo->query($sql);
```

```

        if ($statement===false) return null;
        $result = array();
        foreach ($statement as $row)
        {
            // non-existing fields are ignored
            if (!is_null($field) && !isset($row[$field])) continue;
            // adding new first or specific field given by parameter "field"
            else $result[] = (is_null($field)? $row[0]: $row[$field]);
        }
        return $result;
    }
    public function sql_record($sql)
    {
        $this->sql = $sql;
        $statement = $this->pdo->query($sql);
        if ($statement===false) return null;
        $result = $statement->fetch(PDO::FETCH_ASSOC);
        return $result;
    }
    public function sql_table($sql)
    {
        $this->sql = $sql;
        $statement = $this->pdo->query($sql);
        if ($statement===false) return null;
        $statement->setFetchMode(PDO::FETCH_ASSOC);
        $result = array();
        foreach ($statement as $row) $result[] = $row;
        return $result;
    }
};
$pdo = new WG_PDO_Mysql('login', 'password', 'database'); // default host: 'localhost'
echo 'Affected records: ';
var_dump($pdo->sql_query('INSERT INTO uczen(id,imie,nazwisko) '.
                        'VALUES (111, "Bruce", "Lee")')); echo '<br />';
echo 'Affected records: ';
var_dump($pdo->sql_query('DELETE FROM uczen WHERE id=111'));
echo '<h2>sql_value</h2>';
echo '<pre>First field: '. $pdo->sql_value('SELECT * FROM uczen'). '<br />';
echo 'Field "imie": '. $pdo->sql_value('SELECT * FROM uczen', 'imie'). '</pre>';
echo '<h2>sql_field</h2><pre>'; print_r($pdo->sql_field('SELECT * FROM uczen'));
echo '</pre>';
echo '<h2>sql_record</h2><pre>'; print_r($pdo->sql_record('SELECT * FROM uczen'));
echo '</pre>';
echo '<h2>sql_table</h2><pre>'; print_r($pdo->sql_table('SELECT * FROM uczen'));
echo '</pre>';

```

To tylko przykładowa implementacja klasy obsługującej połączenia z MySQL. Można również tworzyć własne klasy albo rozbudowywać powyższą klasę poprzez tworzenie klas pochodnych.

Zadania

1. Zapoznaj się z przedstawionymi przykładami. Wykonaj je i przeanalizuj ich działanie.
2. Wzbogać przedstawioną klasę o wyjątki obsługujące nieprawidłowo sformułowane zapytanie – niech system zapisze treść błędu do zmiennej „last_error”.
3. Napisz klasę pochodną przedstawionej klasy z polskimi nazwami funkcji składowych.
4. Wzbogać przedstawioną klasę bazową albo utworzoną klasę pochodną o podpinanie danych do zapytania SQL.