

```

<?php

namespace Interfaces;

use \PDO as PDO;

define("DB_HOST", "localhost");
define("DB_USER", "root");
define("DB_PASS", "");
define("DB_NAME", "0917863");

/**
 * Class Database
 * @package Interfaces
 * @author Matthijs Perik
 */
class Database {

    private $host = DB_HOST;
    private $user = DB_USER;
    private $pass = DB_PASS;
    private $db    = DB_NAME;

    private $dbh;
    private $stmt;
    private $error;

    public function __construct() {
        $dsn = 'mysql:host=' . $this->host . ';dbname=' . $this->db . ';charset=utf8';
        $options = array(
            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
        );

        try {
            $this->dbh = new PDO($dsn, $this->user, $this->pass, $options);
        } catch (\PDOException $e) {
            $this->error = $e->getMessage();
        }
    }

    public function __destruct() {
        $this->dbh = null;
    }

    public function prepare($query) {
        $this->stmt = $this->dbh->prepare($query);
    }

    public function bind($param, $value, $option = PDO::PARAM_STR) {

```

```
$this->stmt->bindParam($param, $value, $option);
}

public function execute() {
    return $this->stmt->execute();
}

public function resultSet() {
    $this->execute();
    return $this->stmt->fetchAll(PDO::FETCH_ASSOC);
}

public function single() {
    $this->execute();
    return $this->stmt->fetch(PDO::FETCH_ASSOC);
}

public function fetchColumn() {
    return $this->stmt->fetchColumn();
}

public function rowCount() {
    return $this->stmt->rowCount();
}

public function lastInsertId() {
    return $this->dbh->lastInsertId();
}

public function beginTransaction() {
    return $this->dbh->beginTransaction();
}

public function endTransaction() {
    return $this->dbh->commit();
}

public function cancelTransaction() {
    return $this->dbh->rollBack();
}

public function closeConnection() {
    return $this->dbh = null;
}
}
```