PHP Snippets

- PDO
- Binary search
- Counter - cnt.php

PDO

Voorbeeld PDO Simple

```
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "student";

try {
   $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
   echo "Connected successfully";
}
catch(PDOException $e) {
   echo "Connection failed: " . $e->getMessage();
}

$sql="SELECT * FROM student";
$rows = $conn->query($sql)->fetchAll(); // get ALL rows

foreach($rows as $row) {
  // the tabel student has 4 columns
  echo "$row[0], $row[1], $row[2], $row[3]";
  echo "<br>";
}

$sql="SELECT max(id) FROM student";
$rows = $conn->query($sql)->fetch(); // get ONE rows

echo "Last (max) id is: $rows[0];";

$sql="INSERT INTO student (voornaam, achternaam, email) values
('Max','Bisschop','mb@email.com')";
$rows = $conn->query($sql);
```

```
?>
```

Voorbeeld PDO (OOP en Ass. Array)

Simple PDO class for simple SQL queries, typically used when one connection is sufficient.

```
<?php
// Define DB Params
define("DB_HOST", "localhost");
define("DB_USER", "root");
define("DB_PASS", "geheim");
define("DB_NAME", "mydatabase"



class DB{
protected $dbh;
protected $stmt;
    protected $resultSet;

public function __construct(){
$this->dbh = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME, DB_USER, DB_PASS);
        $this->resultSet=[];
}

public function execute($query){
$this->stmt = $this->dbh->prepare($query);
$result = $this->stmt->execute();
        if (! $result) {
die('Oops, Error execute query '.$query.'Result code: '.$result.);
        }
$this->resultSet = $this->stmt->fetchAll(PDO::FETCH_ASSOC);
return count($this->resultSet);
}

public function getRow(){
if (count($this->resultSet) >0 ){
return array_shift($this->resultSet);
} else {
 return 0;
}
```

```
}
?>
```

Example use of this class

```
<?php
// This creates a new object $DB
// When you want to queries to be acitve
// ( f.e. at the same time update row by row where the rows are the result of a query )
// Create a second connection, f.e. $secondCon = new DB

require_once('db.php');
$sqlquery="SELECT name, age FROM USERS";

$DB->execute($sqlquery);

// get rows (not for updates or inserts)
while ($row = $DB->getRow()) {
echo $row['name']." is ".$row['age']." years old";
}
?>
```

Binary search

This script searches hashed passwords in a text file.

Datafile: download SHA1 hashed file, ordered byhash from https://haveibeenpwned.com/Passwords

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Hacked passwords</title>
</head>

<body>
    <form action="index.php" method="get">
      password: <input type="text" name="password">
      <input type="submit" value="Submit">
    </form>

<?php
function hashDiff($hash1, $hash2) {
    $array1 = str_split($hash1);
    $array2 = str_split($hash2);

    for( $i=0; $i<strlen($hash1); $i++ ) {
        if ($array1[$i] <> $array2[$i]) {
            break;
        }
    }

    return($i);
}

function searchPassword($password){
    $filename="pwned-passwords-sha1-ordered-by-hash-v4.big.txt";

    $hashedPassword = strtoupper(sha1($password));
    echo "<br>Searching for: ".$password;
```

```
echo "<br> Hash: ". $hashedPassword;

if (! $largeFileHandle = fopen($filename, "r")) {
    echo "Cannot open large file ".$filename;
    return;
}

$fileSize = filesize($filename);
$pointer = $fileSize/2;
$step = $fileSize/4;
$maxLineLen = 50;

echo "<br> file size: ".number_format($fileSize,0, "." , " " );

while (true) {
    fseek ( $largeFileHandle , $pointer-$maxLineLen , SEEK_SET );
    $line = fgets($largeFileHandle);
    $line = fgets($largeFileHandle);
    list($lineHash, $lineNumber) = explode(":", $line);

    $currPos = ftell($largeFileHandle);
    //echo "<br> file pos: ".number_format($currPos,0, "." , " " );

    if ($lineHash < $hashedPassword) {
        $pointer = $pointer+$step;
    } elseif ($lineHash > $hashedPassword)  {
        $pointer=$pointer-$step;
    } else { // hash found
        return($line);
    }

    $step = (int)($step/2);

    if ( $step < $maxLineLen ) {
        break;
    }
}

if ($lineHash > $hashedPassword) {
    fseek ( $largeFileHandle , $pointer-50 , SEEK_SET );
```

```
        $line = fgets($largeFileHandle);
        $line = fgets($largeFileHandle);
        list($lineHash, $lineNumber) = explode(":",$line);
    }

    if ($lineHash == $hashedPassword) {
        return($line);
    }

    while ($lineHash > $hashedPassword) {
        fseek($largeFileHandle, ftell($largeFileHandle)-2*$maxLineLen);
        $line = fgets($largeFileHandle);
        $line = fgets($largeFileHandle);
        echo "<br>itterating down, ". $line;
        list($lineHash, $lineNumber) = explode(":", $line);
    }

    while ($lineHash < $hashedPassword) {
        $line = fgets($largeFileHandle);
        list($lineHash, $lineNumber) = explode(":", $line);
        echo "<br>itterating up, ". $line;
        if ($lineHash == $hashedPassword) {
            return($line);
        }
    }

    if ($lineHash == $hashedPassword) {
        return($line);
    }

    return(""); //nothing found

}

if ( isset($_GET['password']) ) {
    $password=$_GET['password'];
    $result = searchPassword($password);

    echo "<br>";
```

```
    if ( strlen($result) > 0 ) {
        list($lineHash, $lineNumber) = explode(":", $result);
        echo "<br><b>".$password." FOUND ".number_format( (int)$lineNumber, 0 , "." , " " )."
times</b>";
    } else {
        echo "<br><b>".$password." NOT found</b>";
    }
}

?>

</body>
</html>
```

Counter - cnt.php

The counter exists of two parts; the JavaScript Code that calls a PHP page with parameters to log the access and the PHP page that logs the call.

Database

The database exists of one table.

```
CREATE TABLE `counters` (
  `ID` bigint(20) NOT NULL,
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `cat` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL,
  `page` varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL,
  `ip` varchar(16) COLLATE utf8mb4_unicode_ci DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

The cat is the category and the page is the page name. With the category you can group pages.

The JavaScript Code or iFrame

This example is used a Laravel App (BookStack) and gets the category and page name from the URL.

The JavaScript code only accepts categories Page or Books, those are the BookStack objects from the URL where we are interested in.

The counter is called asynchronously and is placed via the settings in Bookstack in the header of every page.

```
□<script>
□□var HttpClient = function() {
□□    this.get = function(aUrl, aCallback) {
□□        var anHttpRequest = new XMLHttpRequest();
□□        anHttpRequest.onreadystatechange = function() {
□□            if (anHttpRequest.readyState == 4 && anHttpRequest.status == 200) {
□□                aCallback(anHttpRequest.responseText);
□□□□□}
□□        }
□□        anHttpRequest.open( "GET", aUrl, true );
```

```
        anHttpRequest.send( null );
    }
}

var pathArray = window.location.pathname.split('/');
if ( pathArray.length >=2 ) {                 // only process URL paths with minimum of 2
parts
var page = pathArray[pathArray.length-1];  // get past part of (URL)path
var cat = pathArray[pathArray.length-2];   // get second last part of (URL)path
        if ( cat.toUpperCase() == 'PAGE' || cat.toUpperCase() == 'BOOKS' ) {
    var client = new HttpClient();
    var url='www.roc.ovh/cnt.php?id='+cat+':'+page;
    console.log('Getting:'+url);             // Debug info
    client.get(url, function(response) {
        console.log(response);               // Debug info
            document.body.innerHTML += response; // put result (page coutn of today at
bottom of page
    });
        } else {
          console.log('Invalid cat: '+cat);
        }
    }
</script>
```

Alternatively this code can also be called from an HTML iframe

```
<iframe style="border: none;" src="//www.maxdata.ovh/cnt.php?id=php1-allefilmpjes"></iframe>
```

Nate that in the JavaScript version the cat and page are delimited by a semicolon : whereas in the iframe example the cat and page are delimited by a dash -

The PHP code

The code that is called from the JavaScript (above). This version is tuned to the BookStack environment; the dashes are removed from the page and the page name is converted to CamelCase style to make it a bit more compact.

```
<?php

// Define DB Params
define("DB_HOST", "localhost");
```

```
define("DB_USER", "counters");
define("DB_PASS", "");
define("DB_NAME", "counters");

class DB{
        protected $dbh;
        protected $stmt;
        protected $resultSet;

        public function __construct(){
                $this->dbh = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME, DB_USER,
DB_PASS);
                 $this->resultSet=[];
        }

        public function executeSQL($query){
                $this->stmt = $this->dbh->prepare($query);
                $result = $this->stmt->execute();
                if (! $result) {
                        die('<pre>Oops, Error execute query '.$query.'</pre><br><pre>'.'Result
code: '.$result.'</pre>');
                }
                        $this->resultSet = $this->stmt->fetchAll(PDO::FETCH_ASSOC);
                        return count($this->resultSet);
                }

        public function getRow(){
                if (count($this->resultSet) >0 ){
                        return array_shift($this->resultSet);
                } else {
                        return 0;
                }
        }
}


function dashesToCamelCase($string, $capitalizeFirstCharacter = false)
{
    $str = str_replace(' ', '', ucwords(str_replace('-', ' ', $string)));
    if (!$capitalizeFirstCharacter) $str[0] = strtolower($str[0]);
```

```
    return $str;
}

list($cat, $page)=explode(":", $_GET['id']); // get parameters
$cat = dashesToCamelCase($cat, true);
$page = dashesToCamelCase($page, true);
if ( strlen($cat) > 10 ) $cat = substr($cat,0,10);
if ( strlen($page) > 20 ) $page = substr($page,0,20);

$ip=$_SERVER['REMOTE_ADDR'];

if ( $cat != "" AND $page != "" ) {
        $DB = new DB;
        $query = "INSERT INTO counters (cat,page,ip) VALUES ('$cat', '$page', '$ip')";
        $DB->executeSQL($query);
        $query = "SELECT count(*) FROM counters WHERE DATE(date)=CURDATE() AND cat='$cat' AND
page='$page'";
        $DB->executeSQL($query);
        $result=$DB->getRow()['count(*)'];
} else {
        $result="_";
}
echo "<p style=\"font-size:12px;color:#ACACAC;text-align:center;\">- $result views today -
</p>";
?>
```

The somewhat simpler version:

```
<?php

// Define DB Params
define("DB_HOST", "localhost");
define("DB_USER", "counters");
define("DB_PASS", "");
define("DB_NAME", "counters");

class DB{
        protected $dbh;
        protected $stmt;
        protected $resultSet;
```

```
        public function __construct(){
                $this->dbh = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME, DB_USER,
DB_PASS);
                 $this->resultSet=[];
        }

        public function executeSQL($query){
                $this->stmt = $this->dbh->prepare($query);
                $result = $this->stmt->execute();
                if (! $result) {
                        die('<pre>Oops, Error execute query '.$query.'</pre><br><pre>'.'Result
code: '.$result.'</pre>');
                }
                        $this->resultSet = $this->stmt->fetchAll(PDO::FETCH_ASSOC);
                        return count($this->resultSet);
                }

        public function getRow(){
                if (count($this->resultSet) >0 ){
                        return array_shift($this->resultSet);
                } else {
                        return 0;
                }
        }
}

list($cat, $page)=explode("-", $_GET['id']); // get parameters
$ip=$_SERVER['REMOTE_ADDR'];

if ( $cat != "" AND $page != "" ) {
        $DB = new DB;
        $query = "INSERT INTO counters (cat,page,ip) VALUES ('$cat', '$page', '$ip')";
        $DB->executeSQL($query);
        $query = "SELECT count(*) FROM counters WHERE DATE(date)=CURDATE() AND cat='$cat' AND
page='$page'";
        $DB->executeSQL($query);
        $result=$DB->getRow()['count(*)'];
} else {
        $result="_";
```

```
}
echo "<p style=\"font-size:12px;color:#DCDCDC;\">- $result -</p>";
?>
```

This version simply logs the parameters cat and page specified by the GET. Example ?id=cat-page

This version returns a small and almost white page counter.

--