

# Projektowanie Aplikacji Internetowych

Artur Niewiarowski

Wydział Fizyki, Matematyki i Informatyki

Politechnika Krakowska



# PHP, cz. 2



# **Połączenie z bazą danych MySQL/MariaDB**

## Połączenie z bazą danych MySQL/MariaDB

- MySQLi Object-Oriented
- MySQL Procedural
- PDO - PHP Data Objects

## Połączenie z bazą danych MySQL/MariaDB

### • MySQLi Object-Oriented

```
<?php
$servername = "adres_IP";
$username = "login_do_bazy_danych";
$password = "hasło";

$link = new mysqli($servername, $username, $password);

if ($link->connect_error) {
    die("Błąd połączenia: " . $link->connect_error);
}
echo "Połączenie powiodło się!";

$link->close();
?>
```

## Połączenie z bazą danych MySQL/MariaDB

### • MySQL Procedural

```
<?php
$servername = "adres_IP";
$username = "login_do_bazy_danych";
$password = "hasło";

$link = mysqli_connect($servername, $username,
$password);

if (!$link) {
    die("Błąd połączenia: " . mysqli_connect_error());
}
echo "Połączenie powiodło się!";

mysqli_close($link);
?>
```

## Połączenie z bazą danych MySQL/MariaDB

### • PDO - PHP Data Objects

```
<?php
$servername = "adres_IP";
$username = "login_do_bazy_danych";
$password = "hasło";

try {
    $link = new
PDO("mysql:host=$servername;dbname=myDB", $username,
$password);
    echo "Połączenie powiodło się!";
}
catch(PDOException $e)
{
    echo "Błąd połączenia: " . $e->getMessage();
}
$link = null;
?>
```

[http://pl.wikibooks.org/wiki/PHP/Biblioteka\\_PDO](http://pl.wikibooks.org/wiki/PHP/Biblioteka_PDO)

## Wybrane funkcje dla MySQLi

### **mysqli\_connect()**

```
$link = mysqli_connect("localhost", "login", "hasło",  
"baza_danych");  
  
if (!$link) {  
    die("Błąd połączenia: " . mysqli_connect_error());  
}  
echo "Połączenie powiodło się!";  
  
mysqli_close($link);
```



## Wybrane funkcje dla MySQLi

### **mysqli\_close()**

```
mysqli_close($link);  
  
$q=mysqli_query($link, "select * from tester");  
  
while ($tabl = mysqli_fetch_array($q)) {  
    echo $tabl['ID'];  
}
```

## Wybrane funkcje dla MySQLi

### **mysqli\_affected\_rows()**

```
echo mysqli_affected_rows($link);  
echo "<br>";  
mysqli_query($link,"select * from tester;");  
echo mysqli_affected_rows($link);  
echo "<br>";  
mysqli_query($link,"insert into tester values (null,  
'abc');");  
echo mysqli_affected_rows($link);  
echo "<br>";  
mysqli_query($link,"insert into bledna_nazwa (null,  
'abc');");  
echo mysqli_affected_rows($link);  
echo "<br>";
```

0

3

1

-1

## Wybrane funkcje dla MySQLi

### **mysqli\_autocommit()**

```
mysqli_autocommit($link, FALSE);  
  
mysqli_query($link, "delete from tester;");  
  
//mysqli_commit($link);  
  
//mysqli_query($link, "rollback;");  
//mysqli_rollback($link);
```

## Wybrane funkcje dla MySQLi

### **mysqli\_commit()**

```
mysqli_autocommit($link, FALSE);  
  
mysqli_query($link, "delete from tester;");  
  
mysqli_commit($link); //mysqli_query($link, "commit;");
```

## Wybrane funkcje dla MySQLi

```
mysqli_connect_errno()
```

```
mysqli_connect_error()
```

```
$link=mysqli_connect("localhost", "anewiarowski1",  
"haslo", "anewiarowski_baza");  
  
if (! $link) {  
echo "Błąd połączenia: " . mysqli_connect_errno() . ", " .  
mysqli_connect_error();  
}
```

Błąd połączenia: 1045, Access denied for user 'anewiarowski1'@'localhost' (using password: YES)

## Wybrane funkcje dla MySQLi

```
mysqli_error()  
mysqli_errno()
```

```
$q=mysqli_query($link,"insert into tester values  
(123,456,789)");  
  
if (! $q) {  
echo mysqli_error($link) . ", " . mysqli_errno($link);  
}  
  
echo "<br>";  
  
if (! mysqli_query($link,"select * from testerrr")) {  
echo mysqli_error($link) . ", " . mysqli_errno($link);  
}
```

```
Column count doesn't match value count at row 1, 1136  
Table 'aniewiarowski_baza.testerrr' doesn't exist, 1146
```

## Wybrane funkcje dla MySQLi

### `mysqli_fetch_array()`

```
mysql> desc uzytkownicy;
```

Field	Type	Null	Key	Default	Extra
ID_u	int(11)	NO	PRI	NULL	auto_increment
login	varchar(20)	YES		NULL	
haslo	varchar(32)	YES		NULL	

```
<pre>
```

```
<?php
```

```
$q=mysqli_query($link,"select * from uzytkownicy;");
```

```
while ($tabl = mysqli_fetch_array ($q)) {  
echo "{$tabl['login']} {$tabl['imie']} {$tabl['nazwisko']}  
{$tabl['haslo']} \n";  
}
```

```
?>
```

```
</pre>
```

```
alex    202cb962ac59075b964b07152d234b70  
anka    202cb962ac59075b964b07152d234b70  
michal  202cb962ac59075b964b07152d234b70
```

## Wybrane funkcje dla MySQLi

### `mysqli_fetch_array()`

```
<pre>
<?php
$q=mysqli_query($link,"select * from uzytkownicy;");

while ($tabl = mysqli_fetch_array ($q)) {
echo "{$tabl['0']} {$tabl['1']} {$tabl[2]} {$tabl[3]} \n";
}

?>
</pre>
```

```
5 alex 202cb962ac59075b964b07152d234b70
6 anka 202cb962ac59075b964b07152d234b70
7 michal 202cb962ac59075b964b07152d234b70
```



## Wybrane funkcje dla MySQLi

### **mysqli\_fetch\_assoc()**

```
<pre>
<?php
$q=mysqli_query($link,"select * from uzytkownicy;");

while ($tabl = mysqli_fetch_assoc ($q)) {
echo "{$tabl['login']} {$tabl['haslo']} {$tabl[0]} \n";
}

?>
</pre>
```

```
alex 202cb962ac59075b964b07152d234b70
anka 202cb962ac59075b964b07152d234b70
michal 202cb962ac59075b964b07152d234b70
```

## Wybrane funkcje dla MySQLi

### `mysqli_fetch_row()`

```
<pre>
<?php
$q=mysqli_query($link,"select * from uzytkownicy;");

while ($tabl = mysqli_fetch_row ($q)) {
echo "{$tabl['login']} {$tabl['0']} {$tabl[1]} \n";
}

?>
</pre>
```

```
5 alex
6 anka
7 michal
```

## Wybrane funkcje dla MySQLi

### `mysqli_fetch_row()`

```
<pre>
<?php
$q=mysqli_query($link,"select * from uzytkownicy;");

while ($tabl = mysqli_fetch_row ($q)) {
foreach ($tabl as $r=>$k)
echo "$r:$k | ";
echo "\n";
}

?>
</pre>
```

```
0:5 | 1:alex | 2:202cb962ac59075b964b07152d234b70 |
0:6 | 1:anka | 2:202cb962ac59075b964b07152d234b70 |
0:7 | 1:michal | 2:202cb962ac59075b964b07152d234b70 |
```

## Wybrane funkcje dla MySQLi

```
mysqli_fetch_row()
```

```
<pre>
<?php
$q=mysqli_query($link,"select * from uzytkownicy;");
foreach (mysqli_fetch_row ($q) as $t => $k) {
echo "$t: $k \n";
}
?>
</pre>
```

```
0: 5
1: alex
2: 202cb962ac59075b964b07152d234b70
```

## Wybrane funkcje dla MySQLi

### `mysqli_fetch_lengths()`

```
<pre>
<?php
$q=mysqli_query($link,"select * from uzytkownicy;");

$row = mysqli_fetch_row($q);

foreach (mysqli_fetch_lengths($q) as $t => $k) {
echo "$t: $k \n";
}
?>
</pre>
```

```
0: 1
1: 4
2: 32
```

## Wybrane funkcje dla MySQLi

### `mysqli_fetch_object()`

```
<pre>
<?php
$q=mysqli_query($link,"select * from uzytkownicy;");

while ($obj = mysqli_fetch_object($q)) {
echo $obj->login . "\n";
}

?>
</pre>
```

```
alex
anka
michal
```

## Wybrane funkcje dla MySQLi

### **mysqli\_fetch\_object()**

```
<pre>
<?php

class user {

public static function pobierz_uzytkownika($id) {

$link = new mysqli("localhost", "aniewiarowski", "xxxxxx",
"aniewiarowski_baza");
    if ($wynik = $link->query("select * from uzytkownicy where ID_u =
$id ;")) {
        return $wynik->fetch_object('user');
    }
}

}

$u = user::pobierz_uzytkownika('6');
echo $u->login;

?>
</pre>
```

## Wybrane funkcje dla MySQLi

### **mysqli\_fetch\_object()**

```
<pre>
<?php

class user {

public static function pobierz_uzytkownika($id) {

$link = mysqli_connect("localhost", "anewiarowski", "xxxxxxx",
"anewiarowski_baza");
    if ($wynik = mysqli_query($link,"select * from uzytkownicy where
ID_u = $id ;")) {
        return mysqli_fetch_object($wynik,'user');
    }
}

}

echo user::pobierz_uzytkownika('6')->login;

?>
</pre>
```



## Wybrane funkcje dla MySQLi

### `mysqli_field_count()`

```
<pre>
<?php

mysqli_query($link, "select ID_u from uzytkownicy;");
echo mysqli_field_count($link);

?>
</pre>
```

## Wybrane funkcje dla MySQLi

### `mysqli_num_fields()`

```
<pre>
<?php

$q=mysqli_query($link, "select ID_u from uzytkownicy;");
echo mysqli_num_fields($q);

?>
</pre>
```

## Wybrane funkcje dla MySQLi

### `mysqli_free_result()`

```
<?php
$q=mysqli_query($link, "select * from uzytkownicy;");
mysqli_free_result($q);

while ($tabl = mysqli_fetch_array($q)) {
echo $tabl['login'] . "\n";
}

?>
```

**Warning:** mysqli\_fetch\_array() [[function mysqli-fetch-array](#)]: Couldn't fetch mysqli\_result in `/home/imk-prac/aniewiarowski/` on line 13

## Wybrane funkcje dla MySQLi

```
mysqli_get_server_info()
```

```
<?php  
  
echo mysqli_get_server_info($link);  
  
?>
```

```
5.1.73, 50173
```

## Wybrane funkcje dla MySQLi

### `mysqli_insert_id()`

```
<pre>
<?php

mysqli_query($link, "insert into uzytkownicy (ID_u, login, haslo)
values (null, 'jola', md5('123'))");
echo mysqli_insert_id($link)."\n";
mysqli_query($link, "insert into uzytkownicy (login, haslo)
values ('joanna', md5('123'))");
echo mysqli_insert_id($link)."\n";
mysqli_query($link, "insert into uzytkownicy (ID_u, login, haslo)
values (123,'wladyslaw', md5('123'))");
echo mysqli_insert_id($link)."\n";
mysqli_query($link, "insert into uzytkownicy (ID_u, login, haslo)
values (123,'wladyslaw', md5('123'))");
echo mysqli_insert_id($link)."\n";

?>
</pre>
```

8

9

123

123

## Wybrane funkcje dla MySQLi

### `mysqli_num_rows()`

```
<pre>
<?php

$q=mysqli_query($link, "select * from uzytkownicy;");
echo mysqli_num_rows($q). "\n";

while ($tabl=mysqli_fetch_array($q)){
echo $tabl['login'] . "\n";
}

?>
</pre>
```

```
6
alex
anka
michal
jola
joanna
wladyslaw
```

## Wybrane funkcje dla MySQLi

### `mysqli_ping()`

```
<pre>
<?php

if (mysqli_ping($link))
{
    echo "Połączenie jest ok! " . mysqli_ping($link);
}
else
{
    echo "Błąd!!! ". mysqli_error($link);
}

mysqli_close($link);

?>
</pre>
```

Połączenie jest ok! 1

## Wybrane funkcje dla MySQLi

### **mysqli\_real\_escape\_string()**

```
$_POST['login']="andrzej";
$_POST['haslo']="123";
?>

<pre>
<?php

$_POST['login'] = mysqli_real_escape_string($link, $_POST['login']);
$_POST['haslo'] = mysqli_real_escape_string($link, $_POST['haslo']);

mysqli_query($link, "insert into uzytkownicy (login, haslo)
values ('{$_POST['login']}', md5('{$_POST['haslo']}'))");

?>
</pre>
```



## Wybrane funkcje dla MySQLi

### `mysqli_real_escape_string()`

```
$_POST['login']="tomasz";
$_POST['haslo']="123";
?>

<pre>
<?php

//$_POST['login'] = mysqli_real_escape_string($link, $_POST['login']);
//$_POST['haslo'] = mysqli_real_escape_string($link, $_POST['haslo']);

mysqli_query($link, "insert into uzytkownicy (login, haslo)
values ('{$_POST['login']}', md5('{$_POST['haslo']}'))");

echo mysqli_error($link);

?>
</pre>
```

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '123'))' at line 2

## Wybrane funkcje dla MySQLi

### **mysqli\_real\_escape\_string()**

```
$_POST['login']="tosia";
$_POST['haslo']="123";
?>

<pre>
<?php

foreach ($_POST as $k=>$r) {$$_POST[$k] = mysqli_real_escape_string($link,
$r);}
foreach ($_GET as $k=>$r) {$$_GET[$k] = mysqli_real_escape_string($link, $r);}

mysqli_query($link, "insert into uzytkownicy (login, haslo)
values ('{$_POST['login']}', md5('{$_POST['haslo']}'))");

echo mysqli_error($link);

?>
</pre>
```

## Wybrane funkcje dla MySQLi

**mysqli\_select\_db()**

```
<pre>
<?php

mysqli_select_db($link, "abc_baza");

?>
</pre>
```

## Wybrane funkcje dla MySQLi

### mysqli\_query()

```
<form method=post>
<textarea name=sql></textarea> <input type=submit value=ok>
</form>
<?
if (isset($_POST['sql'])) {
if (stripos($_POST['sql'], "drop ") >-1 || stripos($_POST['sql'], "delete ") >-1)
{die("nie mogę wykonać polecenia");}

$pyt = mysqli_query($link, $_POST['sql']);
echo "<style>table, tr, th, td {border: 1px solid
black;border-collapse:collapse}</style>";
echo "<table>";
while ($tabl = mysqli_fetch_assoc($pyt)) {
echo "<tr>";
foreach ($tabl as $kol => $val) {echo "<td>{$val}</td>";}
echo "</tr>";
}
echo "</table>";
} //if
?>
```

drop table uzytkownicy

ok

nie mogę wykonać polecenia

desc uzytkownicy

ok

ID_u	int(11)	NO	PRI	auto_increment
login	varchar(20)	YES		
haslo	varchar(32)	YES		

show tables

ok

adjectiveswithpositions

adjpositions

adjpositiontypes

casedwords

dict

lexdomains

lexlinks

linktypes

morphmaps

morphology

## Kodowanie znaków strona WWW – baza danych

```
<? header('Content-Type: text/html; charset=utf-8'); ?>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<?
class db {
public $link="";
    function __construct($db, $user, $pass) {

$this->link=mysqli_connect("localhost",$user, $pass, $db) or die ("Błąd
połączenia z bazą danych.");

$this->query("SET character_set_results = 'utf8', character_set_client =
'utf8', character_set_connection = 'utf8', character_set_database = 'utf8',
character_set_server = 'utf8'");

    }

function query($q) {
    return mysqli_query($this->link, $q);
}

} //class
?>
```

## Logowanie do systemu

```
<form method=post>
login: <input type=text name=login><br>
hasło: <input type=text name=haslo><br>
<input type=submit value=loguj>
</form>

<?
$link= mysqli_connect("localhost", "XX", "Hasło", "XX_baza") or die ("błąd
połączenia: " . mysqli_connect_error ());

$pyt = mysqli_query($link, "select count(*) cnt from users where
login='{$_POST['login']}' and pass='{$_POST['haslo']}'" );

if (! $pyt) {echo mysqli_error($link); exit;}

$tabl= mysqli_fetch_assoc($pyt);
if ( $tabl['cnt'] ) {echo "jesteś zalogowana(y)";} else {echo "błędny login
lub hasło!";}
?>
```

## Logowanie do systemu – SQL injection

```
<form method=post>
login: <input type=text name=login><br>
hasło: <input type=text name=haslo><br>
<input type=submit value=loguj>
</form>

<?
$link= mysqli_connect("localhost", "XX", "Hasło", "XX_baza") or die ("błąd
połączenia: " . mysqli_connect_error ());

$pyt = mysqli_query($link, "select count(*) cnt from users where
login='{$_POST['login']}' and pass='{$_POST['haslo']}'" );

if (! $pyt) {echo mysqli_error($link); exit;}

$stabl= mysqli_fetch_assoc($pyt);
if ( $stabl['cnt'] ) {echo "jesteś zalogowana(y)";} else {echo "błędny login
lub hasło!";}
?>
```

login:

hasło:

jesteś zalogowana(y)

## Logowanie do systemu – SQL injection – zabezpieczenie logowania

```
<form method=post>
login: <input type=text name=login><br>
hasło: <input type=text name=haslo><br>
<input type=submit value=loguj>
</form>
```

login:

hasło:

**błędny login lub hasło!**

```
<?
$link= mysqli_connect("localhost", "XX", "Hasło", "XX_baza") or die ("błąd
połączenia: " . mysqli_connect_error ());

$_POST['login'] = mysqli_real_escape_string($link, $_POST['login']);
$_POST['haslo'] = mysqli_real_escape_string($link, $_POST['haslo']);

$pyt = mysqli_query($link, "select count(*) cnt from users where
login='{$_POST['login']}' and pass='{$_POST['haslo']}'" );

if (! $pyt) {echo mysqli_error($link); exit;}

$stabl= mysqli_fetch_assoc($pyt);
if ( $stabl['cnt'] ) {echo "jesteś zalogowana(y)";} else {echo "błędny login
lub hasło!";}
?>
```



# Logowanie do systemu - zabezpieczenie logowania

## Funkcja md5 – JavaScript

```
https://web.mck.pk.edu.pl/~aniewiarowski/lab/PAI/md5.js
```

# Logowanie do systemu - zabezpieczenie logowania

## Przykłady braku zabezpieczenia protokołem *https*

Często odwiedzane  Pierwsze kroki  Galeria obiektów Web ... Sugerowane witryny

### System ankiety "Ocena nauczycieli akademickich"

Login:  
artur

Hasło:  
●●●●●●

Zaloguj się

Do logowania należy zastosować login i hasło używane przy logowaniu do serwisu informacyjnego pracowników PK.

# Logowanie do systemu - zabezpieczenie logowania

## Przykłady braku zabezpieczenia protokołem *https*

```
root@localhost:~  
[root@localhost ~]# tcpflow -i eno2 src host 10.0.0.3 and dst port 80 -c  
tcpflow[3347]: listening on eno2  
010.000.000.003.64491-149.156.132.053.00080: POST /pracownik/logowanie.php HTTP/  
1.1  
Host: ankiety.pk.edu.pl  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:41.0) Gecko/20100101 Firefox/  
41.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: pl,en-US;q=0.7,en;q=0.3  
Accept-Encoding: gzip, deflate  
Content-Type: application/x-www-form-urlencoded; charset=UTF-8  
Referer: http://ankiety.pk.edu.pl/pracownik/login.php  
Content-Length: 25  
Cookie: _ga=GA1.3.1245974602.1422113417; PHPSESSID=5n2gdcmbp83jd77humhrb9crj5  
Connection: keep-alive  
Pragma: no-cache  
Cache-Control: no-cache  
  
login=artur&haslo=haselko
```

obiektyw Web ... Sugerowane witryny

### ankietowy cieli akademickich"

Login:

Hasło:

Zaloguj się



Do logowania należy zastosować login i hasło używane przy logowaniu do serwisu informacyjnego pracowników PK

# Logowanie do systemu - zabezpieczenie logowania

## Przykłady braku zabezpieczenia protokołem *https*



The image shows a web login interface. At the top, a red error message box contains the text: "Błąd logowania! Wpisane zostały niepoprawne dane." Below this is a light blue box titled "LOGOWANIE". It contains two input fields: "Login:" with the value "artur" and "Hasło:" which is empty. A "Loguj" button is positioned below the password field. The background features a blue geometric pattern.

Błąd logowania!  
Wpisane zostały niepoprawne dane.

**LOGOWANIE**

Login: artur

Hasło:

Loguj

(C) by Artur Niewiarowski, Międzynarodowe Centrum Kształcenia PK, 2015

# Logowanie do systemu - zabezpieczenie logowania

## Przykłady braku zabezpieczenia protokołem *https*

```
root@localhost:~  
login=artur&haslo=b01c6054969ea48167b28469b7fa2b64  
010.000.000.0 .64497-1 )): POST /~ankieta/index.php HTTP/1.1  
Host: .pl  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:41.0) Gecko/20100101 Firefox/  
41.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: pl,en-US;q=0.7,en;q=0.3  
Accept-Encoding: gzip, deflate  
Referer: http:// .kieta/index.php  
Cookie: _ga=GA1.3.1245974602.1422113417; cookinfo=1  
Connection: keep-alive  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 50  
  
login=artur&haslo=85605eee6500931d7b35c0757cd34eb1  
010.000.000.0( 64499- POST , a/index.php HTTP/1.1  
Host:  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:41.0) Gecko/20100101 Firefox/  
41.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: pl,en-US;q=0.7,en;q=0.3  
Accept-Encoding: gzip, deflate  
Referer: ./index.php  
Cookie: _ga=GA1.3.1245974602.1422113417; cookinfo=1  
Connection: keep-alive  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 50  
  
login=artur&haslo=cba73c96621c46c09a84d548e6e03660
```



# Inne funkcje haszujące w JavaScript

<https://code.google.com/p/crypto-js/>

## Hashers

### The Hasher Algorithms

#### MD5

MD5 is a widely used hash function. It's been used in a variety of security applications and is also commonly used to check the integrity of files. Though, MD5 is not collision resistant, and it isn't suitable for applications like SSL certificates or digital signatures that rely on this property.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/md5.js"></script>
<script>
  var hash = CryptoJS.MD5("Message");
</script>
```

#### SHA-1

The SHA hash functions were designed by the National Security Agency (NSA). SHA-1 is the most established of the existing SHA hash functions, and it's used in a variety of security applications and protocols. Though, SHA-1's collision resistance has been weakening as new attacks are discovered or improved.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/sha1.js"></script>
<script>
  var hash = CryptoJS.SHA1("Message");
</script>
```

# Inne funkcje haszujące w JavaScript

<https://code.google.com/p/crypto-js/>

## SHA-2

SHA-256 is one of the four variants in the SHA-2 set. It isn't as widely used as SHA-1, though it appears to provide much better security.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/sha256.js"></script>
<script>
  var hash = CryptoJS.SHA256("Message");
</script>
```

SHA-512 is largely identical to SHA-256 but operates on 64-bit words rather than 32.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/sha512.js"></script>
<script>
  var hash = CryptoJS.SHA512("Message");
</script>
```

CryptoJS also supports SHA-224 and SHA-384, which are largely identical but truncated versions of SHA-256 and SHA-512.

## SHA-3

SHA-3 is the winner of a five-year competition to select a new cryptographic hash algorithm where SHA-2 was the runner-up.

**NOTE:** I made a mistake when I named this implementation SHA-3. It should be named Keccak because it's an instance of the Keccak algorithm, which NIST selected as the winner of the SHA-3 competition, hashes identical to Keccak.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/sha3.js"></script>
<script>
  var hash = CryptoJS.SHA3("Message");
</script>
```

SHA-3 can be configured to output hash lengths of one of 224, 256, 384, or 512 bits. The default is 512 bits.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/sha3.js"></script>
<script>
  var hash = CryptoJS.SHA3("Message", { outputLength: 512 });
  var hash = CryptoJS.SHA3("Message", { outputLength: 384 });
  var hash = CryptoJS.SHA3("Message", { outputLength: 256 });
  var hash = CryptoJS.SHA3("Message", { outputLength: 224 });
</script>
```

## RIPEMD-160

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/ripemd160.js"></script>
<script>
  var hash = CryptoJS.RIPEMD160("Message");
</script>
```

## The Hasher Input

The hash algorithms accept either strings or instances of `CryptoJS.lib.WordArray`. A `WordArray` object represents an array of 32-bit words. When you pass a string, it's automatically converted to a `WordArray` encoded as UTF-8.

## The Hasher Output

The hash you get back isn't a string yet. It's a `WordArray` object. When you use a `WordArray` object in a string context, it's automatically converted to a hex string.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/sha256.js"></script>
<script>
  var hash = CryptoJS.SHA256("Message");

  alert(typeof hash); // object

  alert(hash); // 2f77668a9dfbf8d5848b9eeb4a7145ca94c6ed9236e4a773f6dcfa5132b2f91
</script>
```

You can convert a `WordArray` object to other formats by explicitly calling the `toString` method and passing an encoder.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/sha256.js"></script>
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/components/enc-base64-min.js"></script>
<script>
  var hash = CryptoJS.SHA256("Message");

  alert(hash.toString(CryptoJS.enc.Base64)); // L3dmip37+NwEi57rSnFFypTG7ZI25Kdz9tyvpRmRL5E=

  alert(hash.toString(CryptoJS.enc.Latin1)); // /wf00000eJqEE0i6a5s0U+//

  alert(hash.toString(CryptoJS.enc.Hex)); // 2f77668a9dfbf8d5848b9eeb4a7145ca94c6ed9236e4a773f6dcfa5132b2f91
</script>
```

# Inne funkcje haszujące w JavaScript

<https://code.google.com/p/crypto-js/>

## Rabbit

Rabbit is a high-performance stream cipher and a finalist in the eSTREAM Portfolio. It is one of the four designs selected after a 3 1/2-year process where 22 designs were evaluated.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/rabbit.js"></script>
<script>
  var encrypted = CryptoJS.Rabbit.encrypt("Message", "Secret Passphrase");

  var decrypted = CryptoJS.Rabbit.decrypt(encrypted, "Secret Passphrase");
</script>
```

## RC4, RC4Drop

RC4 is a widely-used stream cipher. It's used in popular protocols such as SSL and WEP. Although remarkable for its simplicity and speed, the algorithm's history doesn't inspire confidence in its security.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/rc4.js"></script>
<script>
  var encrypted = CryptoJS.RC4.encrypt("Message", "Secret Passphrase");

  var decrypted = CryptoJS.RC4.decrypt(encrypted, "Secret Passphrase");
</script>
```

It was discovered that the first few bytes of keystream are strongly non-random and leak information about the key. We can defend against this attack by discarding the initial portion of the keystream. This modified algorithm is traditionally called RC4-drop.

By default, 192 words (768 bytes) are dropped, but you can configure the algorithm to drop any number of words.

```
<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/rc4.js"></script>
<script>
  var encrypted = CryptoJS.RC4Drop.encrypt("Message", "Secret Passphrase");

  var encrypted = CryptoJS.RC4Drop.encrypt("Message", "Secret Passphrase", { drop: 3072/4 });

  var decrypted = CryptoJS.RC4Drop.decrypt(encrypted, "Secret Passphrase", { drop: 3072/4 });
</script>
```



# Logowanie do systemu

## md5 – zabezpieczenie logowania

```
<script src=https://torus.uck.pk.edu.pl/~aniewiarowski/md5.js></script>

<form method=post>

login: <input type=text name=login><br>
hasło: <input type=text id=haslo><br>
<input type=hidden name=haslo id=haslo_id>
<input type=submit
onClick="document.getElementById('haslo').disabled='disabled';document.getEle
mentById('haslo_id').value=hex_md5(document.getElementById('haslo').value) "
value=loguj>
</form>
```

## Logowanie do systemu

### md5 – dodatkowy md5 – zabezpieczenie logowania

```
<script src=https://torus.uck.pk.edu.pl/~aniewiarowski/md5.js></script>

<form method=post>

login: <input type=text name=login><br>
hasło: <input type=text id=haslo><br>
<input type=hidden name=haslo id=haslo_id>
<input type=hidden name=los id=los value="<?echo rand(-10000,10000); ?>">

<input type=submit
onClick="document.getElementById('haslo').disabled='disabled';document.getEle
mentById('haslo_id').value=hex_md5(document.getElementById('los').value + '
+ hex_md5(document.getElementById('haslo').value))" value=loguj>
</form>
```

## Logowanie do systemu

md5 – dodatkowy md5 – zabezpieczenie logowania

```
$pyt = mysqli_query($link, "select count(*) cnt from users where  
login='{$_POST['login']}' and md5(concat( '{$_POST['los']}' ,  
pass))='{$_POST['haslo']}'" );
```

## Logowanie do systemu

md5 – dodatkowy md5 + sesja – zabezpieczenie logowania

```
<?
session_start();

?>
<script src=https://torus.uck.pk.edu.pl/~aniewiarowski/md5.js></script>

<form method=post>

login: <input type=text name=login><br>
hasło: <input type=text id=hasło><br>
<input type=hidden name=hasło id=hasło_id>

<input type=submit
onClick="document.getElementById('hasło').disabled='disabled';document.getEle
mentById('hasło_id').value=hex_md5(document.getElementById('los').value + '
+ hex_md5(document.getElementById('hasło').value))" value=loguj>
</form>

<input type=hidden name=los id=los value="<?
$los = md5(rand(-10000,10000));
echo $los;
?>">
```

## Logowanie do systemu

md5 – dodatkowy md5 + sesja – zabezpieczenie logowania

```
<?
if (isset($_POST['login'])) {
$link= mysqli_connect("localhost", "XX", "Hasło", "XX_baza") or die ("błąd
połączenia: " . mysqli_connect_error ());

$_POST['login'] = mysqli_real_escape_string($link, $_POST['login']);
$_POST['haslo'] = mysqli_real_escape_string($link, $_POST['haslo']);

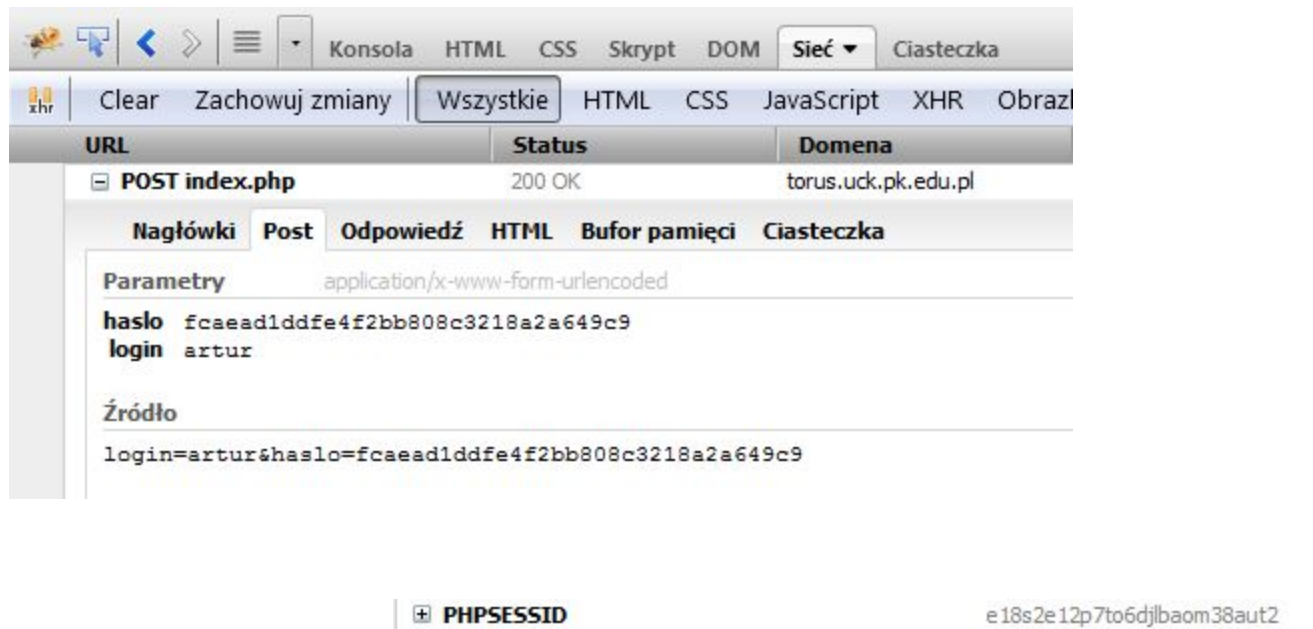
$pyt = mysqli_query($link, "select count(*) cnt from users where
login='{$_POST['login']}' and md5(concat( '{$_SESSION['los']}' ,
pass))='{$_POST['haslo']}'" );

if (! $pyt) {echo mysqli_error($link); exit;}

$tabl= mysqli_fetch_assoc($pyt);
if ( $tabl['cnt'] ) {echo "jesteś zalogowana(y)"; $_SESSION['los'] =
md5(rand(-10000,10000));
} else {
echo "błędny login lub hasło!"; $_SESSION['los'] = $los; }
} else {
$_SESSION['los'] = $los;
}
?>
```

## Logowanie do systemu

md5 – dodatkowy md5 + sesja – zabezpieczenie logowania



The screenshot displays the network tab of a browser's developer tools. The selected request is a POST to `index.php` on the domain `torus.uck.pk.edu.pl`, which returned a `200 OK` status. The request body is encoded as `application/x-www-form-urlencoded` and contains the following data:

Parametry	Value
haslo	fcaead1ddfe4f2bb808c3218a2a649c9
login	artur

The source of the request is `login=artur&haslo=fcaead1ddfe4f2bb808c3218a2a649c9`. Below the network tab, a cookie named `PHPSESSID` is shown with the value `e18s2e12p7to6djlbaom38aut2`.

# Logowanie do systemu

## Logi – błędy logowania użytkowników

```
<?
if (isset($_POST['login'])) {
$link= mysqli_connect("localhost", "XX", "Hasło", "XX_baza") or die ("błąd połączenia: " .
mysqli_connect_error ());

$_POST['login'] = mysqli_real_escape_string($link, $_POST['login']);
$_POST['haslo'] = mysqli_real_escape_string($link, $_POST['haslo']);

$pyt = mysqli_query($link, "select count(*) cnt from users where login='{$_POST['login']}'
and md5(concat( '{$_SESSION['los']}' , pass))='{$_POST['haslo']}'" );

if (! $pyt) {echo mysqli_error($link); exit;}

$tabl= mysqli_fetch_assoc($pyt);
if ( $tabl['cnt'] ) {echo "jesteś zalogowana(y)";} else {echo "błędny login lub hasło!";

mysqli_query($link, "insert into logi (IPhost, login) values ('{$_SERVER['REMOTE_ADDR']}
({$_SERVER['REMOTE_HOST']})', '{$_POST['login']}'");

} //błędne hasło

} //post login

?>
```

## Zabezpieczenie przed cofnięciem się wstecz po wylogowaniu

```
<?
header("Cache-Control: no-store, no-cache, must-revalidate");
header("Cache-Control: post-check=0, pre-check=0, max-age=0", false);
header("Pragma: no-cache");
?>
```



# Połączenie szyfrowane

<https://wiki.centos.org/HowTos/Https>

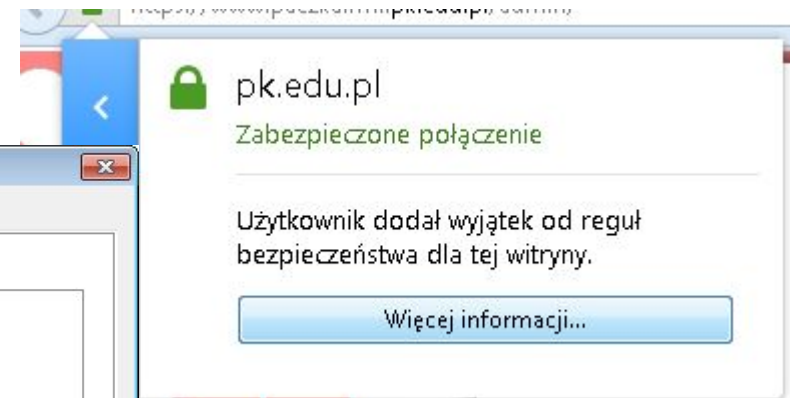
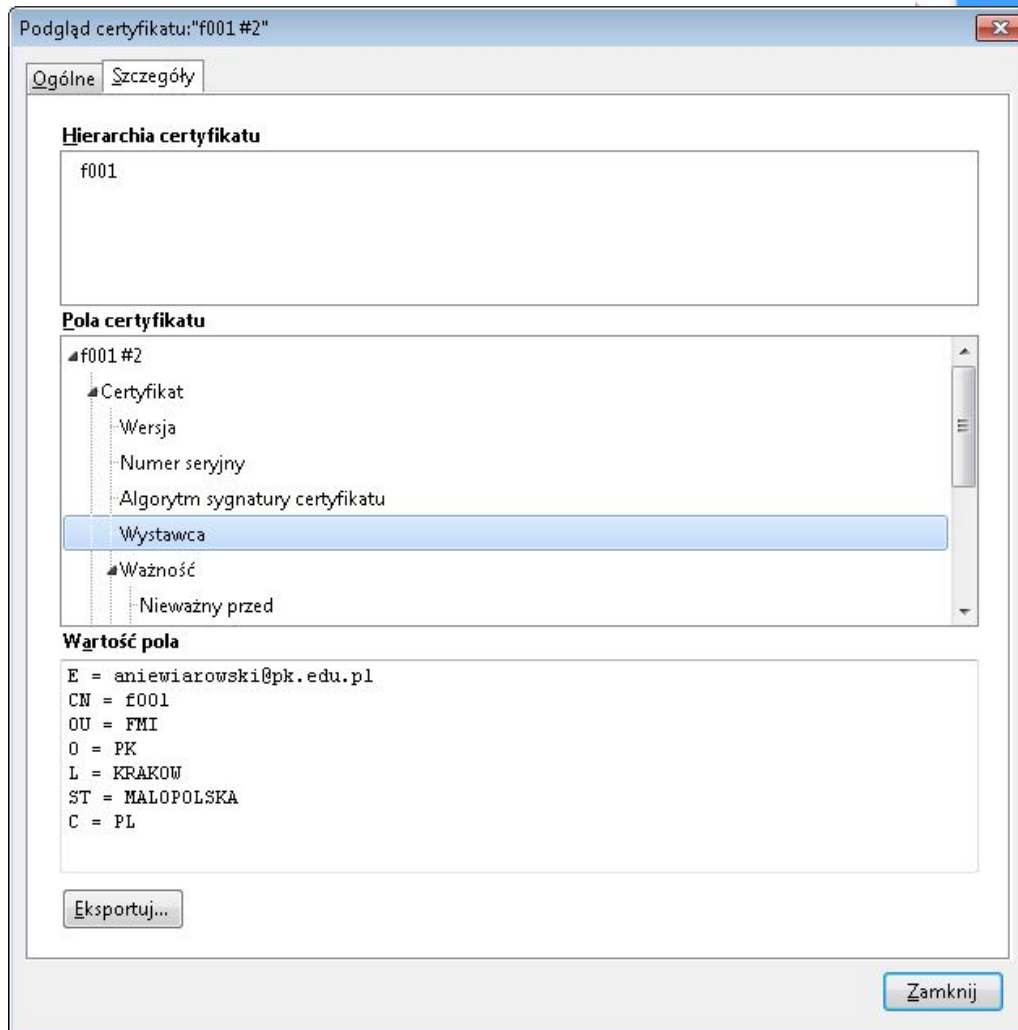
## Setting up an SSL secured Webserver with CentOS

### Contents

1. Getting the required software
2. Generate a self-signed certificate
3. Setting up the virtual hosts
4. Configuring the firewall

This guide will explain how to set up a site over https. The tutorial uses a self signed key so will work provided as is so proceed at your own risk and take backups!

# Połączenie szyfrowane



# **Klasy**

## Klasy – tworzenie klas

```
<?php
class Student
{
public $name;
public $surname;
public function setFullName($name, $surname)
{
$this->name = $name;
$this->surname = $surname;
}
public function getFullName()
{
return $this->name.' '.$this->surname;
}
} //class
```

C. D. N.