

Android 2 – PHP

written by archi | 17 października 2019

Łączenie z bazą danych MySQL za pomocą PHP

Teraz zaczyna się faktyczne kodowanie po stronie serwera. Utwórz klasę PHP, aby połączyć się z bazą danych MySQL. Głównym celem tej klasy jest otwarcie połączenia z bazą danych i zamknięcie połączenia, gdy nie jest to potrzebne. Utwórz dwa pliki o nazwie db_config.php i db_connect.php

db_config.php – będzie miał zmienne połączenia z bazą danych

db_connect.php – plik klasy do połączenia z bazą danych

Poniżej znajduje się kod dwóch plików php

db_config.php

```
<?php
/*
 * All database connection variables
 */
define('DB_USER', "android"); // db user
define('DB_PASSWORD', ""); // db password (mention your db
password here)
define('DB_DATABASE', "androidhive"); // database name
define('DB_SERVER', "localhost"); // db server
?>
```

db_connect.php

```
<?php
/**
 * A class file to connect to database
 */
```

```
function podlacz() {
require_once 'db_config.php';

$con = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD);
mysqli_set_charset($con,"utf8");
mysqli_select_db($con,DB_DATABASE);

return $con;
}
?>
```

Użycie: Gdy chcesz połączyć się z bazą danych MySQL i wykonać jakiegokolwiek operacje, użyj klasy db_connect.php w ten sposób

```
$db = new DB_CONNECT(); // tworzenie obiektu klasy (otworzy połączenie z
bazą danych)
```

Podstawowe operacje CRUD MySQL przy użyciu PHP (Create, Read, Update, Delete)

W swoim projekcie PHP utwórz nowy plik php o nazwie create_product.php i umieść następujący kod. Ten plik służy głównie do tworzenia nowego produktu w tabeli produktów.

W poniższym kodzie czytam dane produktu przez POST i przechowuję je w tabeli produktów. Na koniec odbieram odpowiednią odpowiedź JSON jako odpowiedź.

create_product.php

```
<?php

$response = array();

// check for required fields
if (isset($_POST['name']) && isset($_POST['price']) &&
isset($_POST['description'])) {
```

```
$name = $_POST['name'];
$price = $_POST['price'];
$description = $_POST['description'];

// include db connect class
require_once 'db_connect.php';

// connecting to db
$con = podlacz();

// mysql inserting a new row
$result = mysqli_query($con,"INSERT INTO products(name, price,
description) VALUES('$name', '$price', '$description')");

// check if row inserted or not
if ($result) {
// successfully inserted into database
$response["success"] = 1;
$response["message"] = "Product successfully created.";

// echoing JSON response
echo json_encode($response);
} else {
// failed to insert row
$response["success"] = 0;
$response["message"] = "Oops! An error occurred.";

// echoing JSON response
echo json_encode($response);
}
} else {
// required field is missing
$response["success"] = 0;
$response["message"] = "Required field(s) is missing";

// echoing JSON response
echo json_encode($response);
}
?>
```

Dla kodu powyżej odpowiedź JSON będzie następująca:

Kiedy nie podano parametrów:

```
{
  „success”: 0,
  „message”: „Required field(s) is missing”
}
```

Kiedy produkt został poprawnie utworzony:

```
{
  „success”: 1,
  „message”: „Product successfully created.”
}
```

Kiedy wystąpił problem z dodawaniem danych:

```
{
  „success”: 0,
  „message”: „Oops! An error occurred.”
}
```

Czytanie wiersza z MySQL (czytanie szczegółów produktu)

Utwórz nowy plik php o nazwie `get_product_details.php` i napisz następujący kod. Ten plik otrzyma szczegółowe informacje o pojedynczym produkcie, przyjmując identyfikator produktu (pid) jako parametr wpisu.

get_product_details.php

```
<?php
```

```
$response = array();

// include db connect class
require_once 'db_connect.php';

// connecting to db
$con = podlacz();

// check for post data
if (isset($_GET["pid"])) {
    $pid = $_GET['pid'];

    // get a product from products table
    $result = mysqli_query($con,"SELECT * FROM products WHERE pid =
    $pid");

    if (!empty($result)) {
        // check for empty result
        if (mysqli_num_rows($result) > 0) {

            $result = mysqli_fetch_array($result);

            $product = array();
            $product["pid"] = $result["pid"];
            $product["name"] = $result["name"];
            $product["price"] = $result["price"];
            $product["description"] = $result["description"];
            $product["created_at"] = $result["created_at"];
            $product["updated_at"] = $result["updated_at"];
            // success
            $response["success"] = 1;

            // user node
            $response["product"] = array();

            array_push($response["product"], $product);

            // echoing JSON response
            echo json_encode($response);
        } else {
```

```
// no product found
$response["success"] = 0;
$response["message"] = "No product found";

// echo no users JSON
echo json_encode($response);
}
} else {
// no product found
$response["success"] = 0;
$response["message"] = "No product found";

// echo no users JSON
echo json_encode($response);
}
} else {
// required field is missing
$response["success"] = 0;
$response["message"] = "Required field(s) is missing";

// echoing JSON response
echo json_encode($response);
}
?>
```

Odpowiedź JSON na podstawie powyższego kodu będzie następująca:

Kiedy poprawnie pobierze informacje o produkcie:

```
{
„success”: 1,
„product”: [
{
„pid”: „1”,
„name”: „iPhone 4S”,
„price”: „300.00”,
„description”: „iPhone 4S white”,
```

```
„created_at”: „2012-04-29 01:41:42”,  
„updated_at”: „0000-00-00 00:00:00”  
}  
]  
}
```

Kiedy nie zostanie znaleziony produkt o podanym pid:

```
{  
„success”: 0,  
„message”: „No product found”  
}
```

Wczytanie wszystkich wierszy z bazy (Odczytanie wszystkich produktów)

Potrzebujemy json, aby wyświetlić listę wszystkich produktów na urządzeniu z Androidem. Utwórz nowy plik php o nazwie `get_all_products.php` i napisz następujący kod.

`get_all_products.php`

```
<?php  
$response = array();  
require_once 'db_connect.php';  
//header('Content-type: application/json; charset=utf-8');  
$con = podlacz();  
  
$result = mysqli_query($con, 'SELECT * FROM products');  
  
// check for empty result  
if (!$result === false) {  
$response["products"] = array();  
  
foreach ($result as $row) {  
// temp user array  
$product = array();
```

```
$product["pid"] = $row["pid"];
$product["name"] = $row["name"];
$product["price"] = $row["price"];
$product["description"] = $row["description"];
$product["created_at"] = $row["created_at"];
$product["updated_at"] = $row["updated_at"];

// push single product into final response array
array_push($response["products"], $product);
}
// success

$response["success"] = 1;

// echoing JSON response
echo json_encode($response);
} else {
// no products found
$response["success"] = 0;
$response["message"] = "No products found";

// echo no users JSON
echo json_encode($response);
}
?>
```

A odpowiedź będzie w JSON następująca:

```
{
  „products”: [
    {
      „pid”: „1”,
      „name”: „iPhone 4S”,
      „price”: „300.00”,
      „created_at”: „2012-04-29 02:04:02”,
      „updated_at”: „0000-00-00 00:00:00”
    },
```



```
{
  „pid”: „2”,
  „name”: „Macbook Pro”,
  „price”: „600.00”,
  „created_at”: „2012-04-29 02:04:51”,
  „updated_at”: „0000-00-00 00:00:00”
},
{
  „pid”: „3”,
  „name”: „Macbook Air”,
  „price”: „800.00”,
  „created_at”: „2012-04-29 02:05:57”,
  „updated_at”: „0000-00-00 00:00:00”
},
{
  „pid”: „4”,
  „name”: „OS X Lion”,
  „price”: „100.00”,
  „created_at”: „2012-04-29 02:07:14”,
  „updated_at”: „0000-00-00 00:00:00”
}
],
„success”: 1
}
```

Kiedy nie znajdzie produktów:

```
{
  „success”: 0,
  „message”: „No products found”
}
```

Aktualizowanie wiersza w MySQL (Aktualizowanie

szczegółów produktu)

Utwórz plik php o nazwie `update_product.php`, aby zaktualizować szczegóły produktu. Każdy produkt jest oznaczony pid.

`update_product.php`

```
<?php

$response = array();

// check for required fields
if (isset($_POST['pid']) && isset($_POST['name']) &&
isset($_POST['price']) && isset($_POST['description'])) {

    $pid = $_POST['pid'];
    $name = $_POST['name'];
    $price = $_POST['price'];
    $description = $_POST['description'];

    // include db connect class
    require_once 'db_connect.php';

    // connecting to db
    $con = podlacz();

    // mysql update row with matched pid
    $result = mysqli_query($con,"UPDATE products SET name = '$name',
price = '$price', description = '$description' WHERE pid = $pid");

    // check if row inserted or not
    if ($result) {
        // successfully updated
        $response["success"] = 1;
        $response["message"] = "Product successfully updated.";
    }

    // echoing JSON response
    echo json_encode($response);
}
```

```
} else {  
}  
} else {  
// required field is missing  
$response["success"] = 0;  
$response["message"] = "Required field(s) is missing";  
  
// echoing JSON response  
echo json_encode($response);  
}  
>
```

W przypadku poprawnego wykonania odpowiedź JSON będzie następująca:

```
{  
  „success”: 1,  
  „message”: „Product successfully updated.”  
}
```

Usuwanie wiersza w MySQL (Usuwanie produktu)

Ostatnią operacją jest usunięcie bazy danych. Utwórz nowy plik php o nazwie `delete_product.php` i wklej następujący kod. Główną funkcją tego pliku jest usunięcie produktu z bazy danych.

delete_product.php

```
<?php  
  
$response = array();  
  
// check for required fields  
if (isset($_POST['pid'])) {  
  $pid = $_POST['pid'];
```

```
// include db connect class
require_once 'db_connect.php';

// connecting to db
$con = podlacz();

// mysql update row with matched pid
$result = mysqli_query($con,"DELETE FROM products WHERE pid =
$pid");

// check if row deleted or not
if (mysqli_affected_rows($con) > 0) {
// successfully updated
$response["success"] = 1;
$response["message"] = "Product successfully deleted";

// echoing JSON response
echo json_encode($response);
} else {
// no product found
$response["success"] = 0;
$response["message"] = "No product found";

// echo no users JSON
echo json_encode($response);
}
} else {
// required field is missing
$response["success"] = 0;
$response["message"] = "Required field(s) is missing";

// echoing JSON response
echo json_encode($response);
}
?>
```

W przypadku powodzenia:

```
{  
  „success”: 1,  
  „message”: „Product successfully deleted”  
}
```

w innym przypadku:

```
{  
  „success”: 0,  
  „message”: „No product found”  
}
```