

Forum:[Hack Forums](#)

Thread: [Call to undefined function mysql_connect\(\)](#)

blackcoder00 - June 1, 2020, 7:13 am

guys i have some trouble when i want to connect to the server

Fatal error: Uncaught Error: Call to undefined function mysql_connect() in
C:\AppServ\www\todolist\php\isrt-new-data.php:3 Stack trace: #0 {main} thrown in
C:\AppServ\www\todolist\php\isrt-new-data.php on line 3

PHP Code

```
<?php
mysql_connect("localhost", "root", "test");
mysql_select_db("todolist");

$tdlist = $_POST["insert"];
$date = date("Y/m/d")." ".date("h:i:sa");
$req = mysql_query("insert into list values ('$tdlist','.$date')");
if ($req) {
    echo "donne";
}else {
    echo "not done";
}
```

Nachfolger - June 1, 2020, 9:20 am

Dang dude, no wonder websites get exploited 24/7.

Use MySQLi and prepared statements. MySQL is a meme.

blackcoder00 - June 1, 2020, 12:10 pm

(Jun 1, 2020 09:20 AM)Nachfolger Wrote: [\[CITE\]](#) Dang dude, no wonder websites get exploited 24/7.

Use MySQLi and prepared statements. MySQL is a meme.

it's a self project and i didn't move yet to pdo :(that's why i wanna activate the mysql in the php.ini
but i didn't find the correct way

Nachfolger - June 1, 2020, 12:46 pm

(Jun 1, 2020 12:10 PM)blackcoder00 Wrote: [\[CITE\]](#) it's a self project and i didn't move yet to pdo :(that's why i wanna activate the mysql in the php.ini but i didn't find the correct way
Read my post again. PDO is not the same thing as prepared statements.

Use MySQLi.

Exalted - June 1, 2020, 1:05 pm

As Nachfolger stated there's a difference between MySQLi and PHP Data Objects, MySQLi is MySQL with object-orientation and prepared statements. PDO is a general layer that supports multiple types of databases, which should make it easier to switch between database types without

the need of changing code. There are some more differences and specifics; I would suggest Googling.

Anxuein - June 1, 2020, 2:29 pm

(Jun 1, 2020 12:10 PM)blackcoder00 Wrote: [\[CITE\]](#) it's a self project and i didn't move yet to pdo :(that's why i wanna activate the mysql in the php.ini but i didn't find the correct way
Why would you not want to learn best practices?

blackcoder00 - June 1, 2020, 5:09 pm

(Jun 1, 2020 02:29 PM)Anxuein Wrote: [\[CITE\]](#) Why would you not want to learn best practices?
man because i don't have time , i want to create some web application to organize my time for more practice like a todolist i want to create a simple web application for organization my time to a self bootcamp after that i will code some other one good enough with what i learned

Squito - June 2, 2020, 3:10 am

(Jun 1, 2020 05:09 PM)blackcoder00 Wrote: [\[CITE\]](#) man because i don't have time , i want to create some web application to organize my time for more practice like a todolist i want to create a simple web application for organization my time to a self bootcamp after that i will code some other one good enough with what i learned

All you have to do is replace mysql_ with mysqli_, put a variable in front of mysqli_connect and put that variable into mysqli_query as the first parameter. Welcome to 2013.

Donuts [™] - June 2, 2020, 3:37 am

PHP Code

```
<?php
```

```
$tdlist = htmlspecialchars($_POST["insert"]);  
$date = htmlspecialchars(date("Y/m/d")." ".date("h:i:sa"));
```

```
$servername = "localhost";  
$username = "root";  
$password = "test";  
$dbname = "todolist";
```

```
// Create connection  
$conn = mysqli_connect($servername, $username, $password, $dbname);  
// Check connection  
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
}
```

```
$sql = "INSERT INTO list (list, date ) //Please change it to your own "name" (list, date)  
VALUES ('$tdlist , $date)";
```

```
if (mysqli_query($conn, $sql)) {  
    echo "New record created successfully";  
} else {  
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);  
}
```

```
mysqli_close($conn);
```

but like others said be carefull with this, its not safe version how to insert data into DB.

blackcoder00 - June 2, 2020, 3:41 am

(Jun 2, 2020 03:37 AM)Donuts TM Wrote: [\[CITE\]](#) PHP Code
<?php

```
$tdlist = htmlspecialchars($_POST["insert"]);
$date = htmlspecialchars(date("Y/m/d")." ".date("h:i:sa"));

$servername = "localhost";
$username = "root";
$password = "test";
$dbname = "todolist";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO list (list, date ) //Please change it to your own "name" (list, date)
VALUES ('$tdlist' , $date)";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "_error_: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
```

but like others said be carefull with this, its not safe version how to insert data into DB.
thank you man ???

Disease - June 5, 2020, 7:27 pm

(Jun 1, 2020 07:13 AM)blackcoder00 Wrote: [\[CITE\]](#) guys i have some trouble when i want to connect to the server

Fatal _error_: Uncaught _error_: Call to undefined function mysql_connect() in
C:\AppServ\www\todolist\php\isrt-new-data.php:3 Stack trace: #0 {main} thrown in
C:\AppServ\www\todolist\php\isrt-new-data.php on line 3

PHP Code

```
<?php
mysql_connect("localhost", "root", "test");
mysql_select_db("todolist");

$tdlist = $_POST["insert"];
$date = date("Y/m/d")." ".date("h:i:sa");
$req = mysql_query("insert into list values ('$tdlist' , '$date')");
if ($req) {
    echo "donne";
```

```
}else {  
    echo "not done";  
}
```

The version of PHP you're using is recent enough that the original MySQL extension is deprecated and dropped. PHP 5.5 (which is 7 years old, mind you) deprecated it. And PHP 7.0 (5 years old on its own) dropped it from the build entirely. Unfortunately for you, whatever resource you're using to learn from hasn't caught up with the times. I would shift resources, as you're going to end up with other issues I am sure.

Now, [Donuts™](#) has given you some example code. Unfortunately he whipped together a very simple, and not so secure solution for you. Don't fall into the trap of just taking his code and leaving it alone. The best practice here is to use prepared statements (parameterized queries). Any time you are inserting user-fed data into your database you should be using these. Anything less and you're apt to be susceptible to SQL Injection vulnerabilities.

Without getting too far into the weeds, parameterized queries ship your query and your data separate from each other. The database then determines what should happen in the query, and using the data provided as just that- data. If you were to send a data payload that contained an attempted injection, the database would read the query written for intention, and safely work with the fed-data from there. In other words, a query meant to INSERT can't delete... a query meant to SELECT can't INSERT, etc. etc.

Alternatively, when you write raw queries, and concatenate or interpolate data, the resulting string is what the database receives. If it contains a successful SQL Injection attack then the query would be allowed to run, since the database doesn't know any better. Why tempt fate and go above and beyond to sanitize user input, when the database is perfectly capable of protecting itself via parameterized queries? It's actually more difficult and time consuming to not use them, than it is to learn them and use them everywhere user-fed data is sent to the database.

PHP Code

```
<?php  
$tdlist = htmlspecialchars($_POST["insert"]);  
$date = htmlspecialchars(date("Y/m/d")." ".date("h:i:sa"));  
  
$servername = "localhost";  
$username = "root";  
$password = "test";  
$dbname = "todolist";  
  
// Create connection  
$conn = mysqli_connect($servername, $username, $password, $dbname);  
// Check connection  
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
}  
  
// Set up the parameterized query  
if ($sql = mysqli_prepare($conn, "INSERT INTO list(list, date) VALUES(?, ?)")) {  
    // Bind the data  
    mysqli_stmt_bind_param($sql, "ss", $tdlist, $date);  
  
    // Execute the query
```

```
mysqli_stmt_execute($sql);

if (mysqli_stmt_error($sql)) {
    // Errors found
}

mysqli_stmt_close($sql);
}

mysqli_close($conn);
```

You'll want to read up on a few key functions here, including how to use [mysqli_stmt_bind_param\(\)](#), as I make some sweeping assumptions here that may be incorrect. But overall this should give you a quick example of a working, procedural-written, MySQLi parameterized query implementation.