

Forum:[Hack Forums](#)

Thread: [php/mysql function to insert value not working](#)

blackcoder00 - August 15, 2020, 4:38 am

hey guys im running away from repeating the same code many time XDD i just rewrite some code into a function but i had some problem that function just work 1 time not many time as i like it to do so what i had miss guys ? in my code ? why did the function worked just for one time !! and i have called it 11 time to work (pss : i have checked the input name and it's set correctly)

PHP Code

<?php

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// variable
$date = date('Y-m-d');

$name = htmlspecialchars($_POST["name"]);    $name1 = htmlspecialchars($_POST["name1"]);
$prix = htmlspecialchars($_POST["prixing"]); $prix1 = htmlspecialchars($_POST["prixing1"]);
$prixn = htmlspecialchars($_POST["quantite"]); $prixn1 = htmlspecialchars($_POST["quantite1"]);
$uniteing = $_POST['unite'];                $uniteing1 = $_POST['unite1'];

$name2 = htmlspecialchars($_POST["name2"]);    $name3 = htmlspecialchars($_POST["name3"]);
$prix2 = htmlspecialchars($_POST["prixing2"]); $prix3 = htmlspecialchars($_POST["prixing3"]);
$prixn2 = htmlspecialchars($_POST["quantite2"]); $prixn3 = htmlspecialchars($_POST["quantite3"]);
$uniteing2= $_POST['unite2'];                $uniteing3 = $_POST['unite3'];

$name4 = htmlspecialchars($_POST["name4"]);    $name5 = htmlspecialchars($_POST["name5"]);
$prix4 = htmlspecialchars($_POST["prixing4"]); $prix5 = htmlspecialchars($_POST["prixing5"]);
$prixn4 = htmlspecialchars($_POST["quantite4"]); $prixn5 = htmlspecialchars($_POST["quantite5"]);
$uniteing4 = $_POST['unite4'];                $uniteing5 = $_POST['unite5'];
```

```

$name6 = htmlspecialchars($_POST["name6"]);      $name7 = htmlspecialchars($_POST[
"name7"]);
$prix6 = htmlspecialchars($_POST["prixing6"]);    $prix7 = htmlspecialchars($_POST["prixing7"]);
$prixn6 = htmlspecialchars($_POST["quantite6"]);  $prixn7 = htmlspecialchars($_POST[
"quantite7"]);
$uniteing6 = $_POST['unite6'];                    $uniteing7 = $_POST['unite7'];

$name8 = htmlspecialchars($_POST["name8"]);      $name9 = htmlspecialchars($_POST[
"name9"]);
$prix8 = htmlspecialchars($_POST["prixing8"]);    $prix9 = htmlspecialchars($_POST["prixing9"]);
$prixn8 = htmlspecialchars($_POST["quantite8"]);  $prixn9 = htmlspecialchars($_POST[
"quantite9"]);
$uniteing8 = $_POST['unite8'];                    $uniteing9 = $_POST['unite9'];

$name10 = htmlspecialchars($_POST["name10"]);
$prix10 = htmlspecialchars($_POST["prixing10"]);
$prixn10 = htmlspecialchars($_POST["quantite10"]);
$uniteing10 = $_POST['unite10'];

//end variable 2

function insert($namex, $prixx,$prixnx, $datex, $uniteingx,$conn){

$sql = "INSERT INTO ingredient VALUES ('$namex','$prixx','$prixnx','$datex','$uniteingx')";
$res = mysqli_query($conn, $sql);

if ($res) {
echo "New record created successfully";
mysqli_error($conn);
}

else {
echo "_error_ : " . $sql . "<br>" . mysqli_error($conn);

}

}

insert($name, $prix,$prixn, $date, $uniteing,$conn);
insert($name1, $prix1,$prixn1, $date1, $uniteing1,$conn);
insert($name2, $prix2,$prixn2, $date2, $uniteing2,$conn);
insert($name3, $prix3,$prixn3, $date3, $uniteing3,$conn);
insert($name4, $prix4,$prixn4, $date4, $uniteing4,$conn);
insert($name5, $prix5,$prixn5, $date5, $uniteing5,$conn);
insert($name6, $prix6,$prixn6, $date6, $uniteing6,$conn);
insert($name7, $prix7,$prixn7, $date7, $uniteing7,$conn);
insert($name8, $prix8,$prixn8, $date8, $uniteing8,$conn);
insert($name9, $prix9,$prixn9, $date9, $uniteing9,$conn);

```

```
insert($name10, $prix10,$prixn10, $date10, $uniteing10,$conn);
```

```
header('Location: ../index.html');  
?>
```

Nachfolger - August 15, 2020, 6:56 am

Granted, I did only a light skim, you should also echo out the STMT errors in addition to connection errors (Only during debugging, of course). Beyond that, it looks like you know exactly how many records will be created each time the form is submitted, why aren't you just creating an array and iterating through that? Using a function in this case is trivial--At best.

Eg.

PHP Code

```
<?php  
$rows = Array();  
$rows[] = Array("name" => htmlspecialchars($_POST["name"]), "prixing" => htmlspecialchars(  
$_POST["prixing"]), "quantite" => htmlspecialchars($_POST["quantite"]), "unite" => $_POST['unite']);  
$rows[] = Array("name" => htmlspecialchars($_POST["name1"]), "prixing" => htmlspecialchars(  
$_POST["prixing1"]), "quantite" => htmlspecialchars($_POST["quantite1"]), "unite" => $_POST[  
'unite1']);  
// ... continue with each form  
  
if ($stmt = $con->prepare(  
"INSERT INTO ingredient (Name, Prixx, Prixn, Datex, Uniteing) VALUES (?, ?, ?, ?, ?)" ) {  
    $stmt->bind_param("sssss", $Name, $Prixx, $Prixn, $Datex, $Uniteing);  
    // TBD: Fix statement variable types (Eg. "s", "i", "d");  
    foreach ($rows as $row) {  
        $Name = $row["name"];  
        // TBD, finish variable assignment  
        $stmt->execute();  
    }  
    $stmt->close();  
    $con->close();  
}
```

Beyond the scope of what you asked for here,

- You MUST always use die() directly after calling header(). Browsers can theoretically ignore the header request, and the page will keep executing. In this case, there's nothing beyond your header() call, but you should be in the constant mindset of adding die() after a header() call.
- You should specify column names in the insert statement (As I demonstrated, by guessing column names)

Using die():

PHP Code

```
<?php  
header('Location: ../index.html');  
die();  
Or
```

PHP Code

```
<?php  
die(header('Location: ../index.html'));
```

blackcoder00 - August 15, 2020, 7:28 am

(Aug 15, 2020 06:56 AM)Nachfolger Wrote: [\[CITE\]](#) Granted, I did only a light skim, you should also echo out the STMT errors in addition to connection errors (Only during debugging, of course). Beyond that, it looks like you know exactly how many records will be created each time the form is submitted, why aren't you just creating an array and iterating through that? Using a function in this case is trivial--At best.

Eg.

PHP Code

```
<?php
$rows = Array();
$rows[] = Array("name" => htmlspecialchars($_POST["name"]), "prixing" => htmlspecialchars(
$_POST["prixing"]), "quantite" => htmlspecialchars($_POST["quantite"]), "unite" => $_POST['unite']);
$rows[] = Array("name" => htmlspecialchars($_POST["name1"]), "prixing" => htmlspecialchars(
$_POST["prixing1"]), "quantite" => htmlspecialchars($_POST["quantite1"]), "unite" => $_POST[
'unite1']);
// ... continue with each form

if ($stmt = $con->prepare(
"INSERT INTO ingredient (Name, Prixx, Prixn, Datex, Uniteing) VALUES (?, ?, ?, ?, ?)") {
    $stmt->bind_param("sssss", $Name, $Prixx, $Prixn, $Datex, $Uniteing);
// TBD: Fix statement variable types (Eg. "s", "i", "d");
    foreach ($rows as $row) {
        $Name = $row["name"];
        // TBD, finish variable assignment
        $stmt->execute();
    }
    $stmt->close();
    $con->close();
}
```

Beyond the scope of what you asked for here,

- You MUST always use die() directly after calling header(). Browsers can theoretically ignore the header request, and the page will keep executing. In this case, there's nothing beyond your header() call, but you should be in the constant mindset of adding die() after a header() call.
- You should specify column names in the insert statement (As I demonstrated, by guessing column names)

Using die():

PHP Code

```
<?php
header('Location: ../index.html');
die();
```

Or

PHP Code

```
<?php
die(header('Location: ../index.html'));
```

thank you soo muchhh <3

(Aug 15, 2020 06:56 AM)Nachfolger Wrote: [\[CITE\]](#) Granted, I did only a light skim, you should also echo out the STMT errors in addition to connection errors (Only during debugging, of course). Beyond that, it looks like you know exactly how many records will be created each time the form is

submitted, why aren't you just creating an array and iterating through that? Using a function in this case is trivial--At best.

Eg.

PHP Code

```
<?php
```

```
$rows = Array();
$rows[] = Array("name" => htmlspecialchars($_POST["name"]), "prixing" => htmlspecialchars(
$_POST["prixing"]), "quantite" => htmlspecialchars($_POST["quantite"]), "unite" => $_POST['unite']);
$rows[] = Array("name" => htmlspecialchars($_POST["name1"]), "prixing" => htmlspecialchars(
$_POST["prixing1"]), "quantite" => htmlspecialchars($_POST["quantite1"]), "unite" => $_POST[
'unite1']);
// ... continue with each form
```

```
if ($stmt = $con->prepare(
"INSERT INTO ingredient (Name, Prixx, Prixn, Datex, Uniteing) VALUES (?, ?, ?, ?, ?)" ) {
    $stmt->bind_param("sssss", $Name, $Prixx, $Prixn, $Datex, $Uniteing);
    // TBD: Fix statement variable types (Eg. "s", "i", "d");
    foreach ($rows as $row) {
        $Name = $row["name"];
        // TBD, finish variable assignment
        $stmt->execute();
    }
    $stmt->close();
    $con->close();
}
```

Beyond the scope of what you asked for here,

- You MUST always use die() directly after calling header(). Browsers can theoretically ignore the header request, and the page will keep executing. In this case, there's nothing beyond your header() call, but you should be in the constant mindset of adding die() after a header() call.
- You should specify column names in the insert statement (As I demonstrated, by guessing column names)

Using die():

PHP Code

```
<?php
```

```
header('Location: ../index.html');
die();
```

Or

PHP Code

```
<?php
```

```
die(header('Location: ../index.html'));
```

i got this problem but i checked the code there is no ; missing

Parse error: syntax error, unexpected ';' in C:\Program

Files\Amp\www\test\test\test1\php\insert-multi-ing.php on line 24

(Aug 15, 2020 06:56 AM)Nachfolger Wrote: [\[CITE\]](#) Granted, I did only a light skim, you should also echo out the STMT errors in addition to connection errors (Only during debugging, of course). Beyond that, it looks like you know exactly how many records will be created each time the form is submitted, why aren't you just creating an array and iterating through that? Using a function in this case is trivial--At best.

Eg.

PHP Code

```
<?php
$rows = Array();
$rows[] = Array("name" => htmlspecialchars($_POST["name"]), "prixing" => htmlspecialchars(
$_POST["prixing"]), "quantite" => htmlspecialchars($_POST["quantite"]), "unite" => $_POST['unite']);
$rows[] = Array("name" => htmlspecialchars($_POST["name1"]), "prixing" => htmlspecialchars(
$_POST["prixing1"]), "quantite" => htmlspecialchars($_POST["quantite1"]), "unite" => $_POST[
'unite1']);
// ... continue with each form

if ($stmt = $con->prepare(
"INSERT INTO ingredient (Name, Prixx, Prixn, Datex, Uniteing) VALUES (?, ?, ?, ?, ?)") {
    $stmt->bind_param("sssss", $Name, $Prixx, $Prixn, $Datex, $Uniteing);
// TBD: Fix statement variable types (Eg. "s", "i", "d");
    foreach ($rows as $row) {
        $Name = $row["name"];
        // TBD, finish variable assignment
        $stmt->execute();
    }
    $stmt->close();
    $con->close();
}
```

Beyond the scope of what you asked for here,

- You MUST always use die() directly after calling header(). Browsers can theoretically ignore the header request, and the page will keep executing. In this case, there's nothing beyond your header() call, but you should be in the constant mindset of adding die() after a header() call.
- You should specify column names in the insert statement (As I demonstrated, by guessing column names)

Using die():

PHP Code

```
<?php
header('Location: ../index.html');
die();
Or
PHP Code
<?php
die(header('Location: ../index.html'));
```

there is a missing) in the if statement but i try it the code it didn't insert any values

PHP Code

```
<?php
if (!$con) {
    die("Connection failed: " . mysqli_connect_error());
}
// variable
$rows = Array();
$rows[] = Array("name" => htmlspecialchars($_POST["name"]), "prixing" => htmlspecialchars(
$_POST["prixing"]), "quantite" => htmlspecialchars($_POST["quantite"]), "unite" => $_POST['unite']);
```

```

$rows[] = Array("name" => htmlspecialchars($_POST["name1"]), "prixing" => htmlspecialchars(
$_POST["prixing1"]), "quantite" => htmlspecialchars($_POST["quantite1"]), "unite" => $_POST[
'unite1']);
// ... continue with each form

if ($stmt = $con->prepare("INSERT INTO ingredient VALUES (?, ?, ?, ?, ?)")) {
    $stmt->bind_param("sssds", $Name, $Prixx, $Prixn, $Datex, $Uniteing);

    foreach ($rows as $row) {
        $Name = $row["name"];
        $Prixn = $row["prix"];
        $Prixn = $row["quantite"];
        $Datex = $row["date"];
        $Uniteing = $row["uniteingre"];
        $stmt->execute();
    }
    $stmt->close();
    $con->close();
}

```

Charger - August 15, 2020, 7:58 am

(Aug 15, 2020 07:28 AM)blackcoder00 Wrote: [CITE](#) thank you soo muchhh <3

i got this problem but i checked the code there is no ; missing
 Parse _error_: syntax _error_, unexpected ';' in C:\Program
 Files\Amp\www\test\test\test1\php\insert-multi-ing.php on line 24

there is a missing) in the if statement but i try it the code it didn't insert any values

PHP Code

```

<?php
if (!$con) {
    die("Connection failed: " . mysqli_connect_error());
}
// variable
$rows = Array();
$rows[] = Array("name" => htmlspecialchars($_POST["name"]), "prixing" => htmlspecialchars(
$_POST["prixing"]), "quantite" => htmlspecialchars($_POST["quantite"]), "unite" => $_POST['unite']);
$rows[] = Array("name" => htmlspecialchars($_POST["name1"]), "prixing" => htmlspecialchars(
$_POST["prixing1"]), "quantite" => htmlspecialchars($_POST["quantite1"]), "unite" => $_POST[
'unite1']);
// ... continue with each form

if ($stmt = $con->prepare("INSERT INTO ingredient VALUES (?, ?, ?, ?, ?)")) {
    $stmt->bind_param("sssds", $Name, $Prixx, $Prixn, $Datex, $Uniteing);

    foreach ($rows as $row) {
        $Name = $row["name"];
        $Prixn = $row["prix"];
        $Prixn = $row["quantite"];
    }
}

```

```

        $Datex = $row["date"];
        $Uniteing = $row["uniteingre"];
        $stmt->execute();
    }
    $stmt->close();
    $con->close();
}

```

Do this before execute

```
$stmt->debugDumpParams();
```

Also, I'm pretty sure you have to have your bind values set before you bind them. I'm like 99% sure on that

[Nachfolger](#)

Also just use Medoo

blackcoder00 - August 15, 2020, 8:05 am

(Aug 15, 2020 07:58 AM)Charger Wrote: [\[CITE\]](#) Do this before execute

```
$stmt->debugDumpParams();
```

Also, I'm pretty sure you have to have your bind values set before you bind them. I'm like 99% sure on that

[Nachfolger](#)

Also just use Medooi got this error i have set values with many input

Fatal error: Uncaught Error: Call to undefined method mysqli_stmt::debugDumpParams() in C:\Program Files\Ampps\www\test\test1\php\insert-multi-ing.php:32 Stack trace: #0 {main} thrown in C:\Program Files\Ampps\www\test\test1\php\insert-multi-ing.php on line 32

Nachfolger - August 15, 2020, 9:10 am

(Aug 15, 2020 07:28 AM)blackcoder00 Wrote: [\[CITE\]](#) thank you soo muchhhh <3

i got this problem but i checked the code there is no ; missing

Parse _error_: syntax _error_, unexpected ';' in C:\Program Files\Ampps\www\test\test1\php\insert-multi-ing.php on line 24

there is a missing) in the if statement but i try it the code it didn't insert any values

PHP Code

```

<?php
if (!$con) {
    die("Connection failed: " . mysqli_connect_error());
}
// variable
$rows = Array();
$rows[] = Array("name" => htmlspecialchars($_POST["name"]), "prixing" => htmlspecialchars(
$_POST["prixing"]), "quantite" => htmlspecialchars($_POST["quantite"]), "unite" => $_POST["unite"]);
$rows[] = Array("name" => htmlspecialchars($_POST["name1"]), "prixing" => htmlspecialchars(
$_POST["prixing1"]), "quantite" => htmlspecialchars($_POST["quantite1"]), "unite" => $_POST[
'unite1']);

```

// ... continue with each form

```
if ($stmt = $con->prepare("INSERT INTO ingredient VALUES (?, ?, ?, ?, ?)")) {
    $stmt->bind_param("ssds", $Name, $Prixx, $Prixn, $Datex, $Uniteing);

    foreach ($rows as $row) {
        $Name = $row["name"];
        $Prixn = $row["prix"];
        $Prixn = $row["quantite"];
        $Datex = $row["date"];
        $Uniteing = $row["uniteingre"];
        $stmt->execute();
    }
    $stmt->close();
    $con->close();
}
```

Sorry about the errors, I wrote this in thread and didn't check for syntax issues.

I'm unsure what language you're using, but if "\$Datex" is a date, don't use the string "ssds". The d does not mean date, it means decimal. Additionally, why did you remove the column names? I put them in there for a reason.

1. Post a picture of the table design/structure (Preferably from PHPMysqlAdmin, but it doesn't matter)
2. Find any place I put "\$con" and replace it with "\$conn", that was my mistake.
3. Change "\$stmt->execute" to the following:

PHP Code

```
<?php
```

```
if (!$stmt->execute()) { die($stmt->error ." ". $conn->error); }
```

(Aug 15, 2020 07:58 AM)Charger Wrote: [CITE](#) Also, I'm pretty sure you have to have your bind values set before you bind them. I'm like 99% sure on that

[Nachfolger](#) I'm not sure if you're talking about this:

PHP Code

```
<?php
```

```
$stmt->bind_param("sssss", $Name, $Prixx, $Prixn, $Datex, $Uniteing);
```

```
// TBD: Fix statement variable types (Eg. "s", "i", "d");
```

```
foreach ($rows as $row) {
    $Name = $row["name"];
    // TBD, finish variable assignment
    $stmt->execute();
}
```

But if you are, you do not need to define the variables before calling bind_param.

As for the microframework suggestion, it would certainly prove to be more useful, but goodluck getting him to switch. I'm fairly certain this is the guy who took 4 years to switch to MySQLi--And apparently he still refuses to utilize prepared statements (Given OP).

blackcoder00 - August 15, 2020, 9:25 am

(Aug 15, 2020 09:10 AM)Nachfolger Wrote: [CITE](#) Sorry about the errors, I wrote this in thread and didn't check for syntax issues.

I'm unsure what language you're using, but if "\$Datex" is a date, don't use the string "ssds". The d

does not mean date, it means decimal. Additionally, why did you remove the column names? I put them in there for a reason.

1. Post a picture of the table design/structure (Preferrably from PHPMysqlAdmin, but it doesn't matter)
2. Find any place I put "\$con" and replace it with "\$conn", that was my mistake.
3. Change "\$stmt->execute" to the following:

PHP Code

```
<?php
if (!$stmt->execute()) { die($stmt->_error_ . " ". $conn->_error_); }
```

I'm not sure if you're talking about this:

PHP Code

```
<?php
$stmt->bind_param("sssss", $Name, $Prixx, $Prixn, $Datex, $Uniteing);
// TBD: Fix statement variable types (Eg. "s", "i", "d");
foreach ($rows as $row) {
    $Name = $row["name"];
    // TBD, finish variable assignment
    $stmt->execute();
}
```

But if you are, you do not need to define the variables before calling bind_param.

As for the microframework suggestion, it would certainly prove to be more useful, but goodluck getting him to switch. I'm fairly certain this is the guy who took 4 years to switch to MySQLi--And apparently he still refuses to utilize prepared statements (Given OP).

i try the following code and it's work fine thank you <3333

PHP Code

```
<?php
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$stmt = $db->prepare("
    INSERT
    INTO ingredient
    VALUES (:name, :prix, :quantite, NOW(), :unite)
");

$stmt->bindParam('name', $name);
$stmt->bindParam('prix', $prix);
$stmt->bindParam('quantite', $quantite);
$stmt->bindParam('unite', $unite);

foreach ($data as $item) {

    // Adds some data validation to make sure you won't save million empty rows,
    // also add custom validation for other fields (if any)

    $name = checkValue($item['name']);
    $prix = checkValue($item['prix']);

    $quantite = floatval($item['quantite']);
    $unite = checkValue($item['unite']);
```

```
        if (!is_null($name) && !is_null($prix) && !is_null($unite) && $quantite > 0) {  
            $stmt->execute();  
        }  
    }  
}
```

Squito - *August 16, 2020, 3:43 am*

Also instead of applying htmlspecialchars to every element of \$_POST manually:

PHP Code

```
<?php
```

```
$myarray=array_map('htmlspecialchars', $_POST);
```