

Forum:[Hack Forums](#)

Thread: [How to safely let users connect to a mysql](#)

Footer - August 16, 2019, 11:38 am

VB.NET related topic.

I want to create a program that let users to have their own account stored in a mysql database. How to do that in a safe way? I know it can be reversed to know mysql database login details. I should use api?

devvariety - August 16, 2019, 11:49 am

Yes, you should make an API instead of leaving credentials in your application.

Anything you don't want your users seeing should be server sided

Footer - August 16, 2019, 12:19 pm

Tutorial where I can start from?

memento - August 16, 2019, 4:03 pm

(Aug 16, 2019 12:19 PM)Loque Wrote: [\[CITE\]](#) Tutorial where I can start from?

You'll need 3 things: Web API, Standalone VB.NET App and a MySQL database.

You will interact with the database via your Web API and you'll need interact with your Web API through HTTP requests on your VB.NET app.

slober - August 16, 2019, 4:39 pm

(Aug 16, 2019 11:49 AM)devvariety Wrote: [\[CITE\]](#) Yes, you should make an API instead of leaving credentials in your application.

Anything you don't want your users seeing should be server sided

Absolutely right. you shouldn't leave Database credentials in your application.

There are a lot of services exactly for this (ex. rest API's). Not only that but also a few MySQL alternatives [restdb.io](#)

firebase

deployd

cloudboost

You can also write a simple API in PHP.

I've done something like this a few years ago for my Honeypot project PotHead
<https://github.com/Slober3/PotHead/tree/master/Website%2BBackend/api>

PHP Code

```
<?php  
include '../includes/config.php';
```

```

// What can we get from this api?
// Filter on Time,IP,ID,(Input)
// Count records
// http://YOURSITE.COM/api/api.php/{connection|cinput}/{all}/
// http://YOURSITE.COM/api/api.php/{connection|cinput}/{last}/{int} (number max 100)
// http://YOURSITE.COM/api/api.php/{connection|cinput}/{ip}/{ip} (IP must be an ip)
// http://YOURSITE.COM/api/api.php/{connection|cinput}/{id}/{id} (Must be a number)
// http://YOURSITE.COM/api/api.php/{connection|cinput}/{recordCount}
// Retrieve Data from url
$request = explode('/', trim($_SERVER['PATH_INFO'], '/'));
// Retrieve the database,command and the specification
$databaseselect = preg_replace('/[^\a-z0-9_]+/i', '', array_shift($request));
$command = preg_replace('/[^\a-z0-9_]+/i', '', array_shift($request));
$specification = array_shift($request);
// check is database exists
switch ($databaseselect)
{
case 'cinput':
$databaseselect = "loginput";
break;
case 'connection':
$databaseselect = "iplog";
break;
default:
$databaseselect = "iplog";
break;
}
// connect to the database
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// What command was used?
switch ($command)
{
case 'all':
$stmt = $conn->prepare("select * from $databaseselect");
break;
case 'ip':
filter_var($specification, FILTER_VALIDATE_IP) ? $specification = '127.0.0.1':
$stmt = $conn->prepare("select * from $databaseselect where ip = :ip");
$stmt->bindParam(':ip', $specification, PDO::PARAM_STR);
break;
case 'id':
filter_var($specification, FILTER_VALIDATE_INT) ? $specification = intval($specification) :
$specification = 1;
$stmt = $conn->prepare("select * from $databaseselect where id = :id");
$stmt->bindParam(':id', $specification, PDO::PARAM_INT);
break;
case 'last':
filter_var($specification, FILTER_VALIDATE_INT, array(
"options" => array(
"min_range" => 1,
"max_range" => 100
)
)

```

```

)) ? $specification = intval($specification) : $specification = 30;
$stmt = $conn->prepare("SELECT * FROM ( SELECT * FROM $databaseSelect
ORDER BY id DESC LIMIT :intlim ) sub ORDER BY id ASC");
$stmt->bindParam(':intlim', $specification, PDO::PARAM_INT);
break;

```

```

case 'recordCount':
    $stmt = $conn->prepare("select COUNT(*) FROM $databaseSelect");
    break;
default:
    $stmt = $conn->prepare("select * from $databaseSelect");
    break;
}
// Execute SQL statement
$stmt->execute();
$results = $stmt->fetchAll(PDO::FETCH_ASSOC);
// Convert to json and print
$json = json_encode($results);
print htmlspecialchars($json, ENT_NOQUOTES);
// Close connection
$conn = null;
?>

```

Please do not put your credentials to your DB inside your application use an API.

You can read the json in VB.net quite simply by using the newtonsoft library here is a small example:

```

Code Imports System.IO
Imports System.Net
Imports Newtonsoft.Json.Linq

```

```

Public Class Form1

```

```

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

```

```

        Dim inStream As StreamReader

```

```

        Dim webRequest As WebRequest

```

```

        Dim webresponse As WebResponse

```

```

        webRequest = webRequest.Create("https://jsonplaceholder.typicode.com/todos/1")

```

```

        webresponse = webRequest.GetResponse()

```

```

        inStream = New StreamReader(webresponse.GetResponseStream())

```

```

        Dim test = inStream.ReadToEnd.ToString

```

```

        'Read the raw data from url

```

```

        Console.WriteLine("RAW: " + test)

```

```

        'Make it looks nice

```

```

        Dim jsonobject As JObject = JObject.Parse(test)

```

```

        Console.WriteLine(jsonobject.SelectToken("title").ToString)

```

```

    End Sub

```

```

End Class

```

```

'{
' "userId" 1,
' "id": 1,

```

```
' "title": "delectus aut autem",  
' "completed": false  
'}
```

Fhack - *August 17, 2019, 12:02 pm*

Hi, make post https request with username and password as data on php page hosted on your server and echo 1 if it's good and 0 if not for example then read the answer in JSON format if you want more data than 0 or 1 ! The php code should check if credentials are valid with the mysql database ! Also check post parameters before connect and making your request on mysql to avoid SQL injection !