

# How to connect flutter app to mysql web server and phpmyadmin



Santos Enoque [Follow](#)  
Feb 24, 2019 · 5 min read

Hey guys on this article I am going to show you the step by step process that will enable you to connect your flutter app to a web server!



Flutter



Connect **FLUTTER** app  
with, **PHP** and  
**MYSQL** server

Connect flutter app to mySql server and phpmyadmin



## Configure your web server

Let's agree that the first step, to connect to a web server is to have a web server :D, well if you have your own server already, thats good, but if you don't, heres how you can have a free web server!

### 1. create a free account on 00webhosting.com

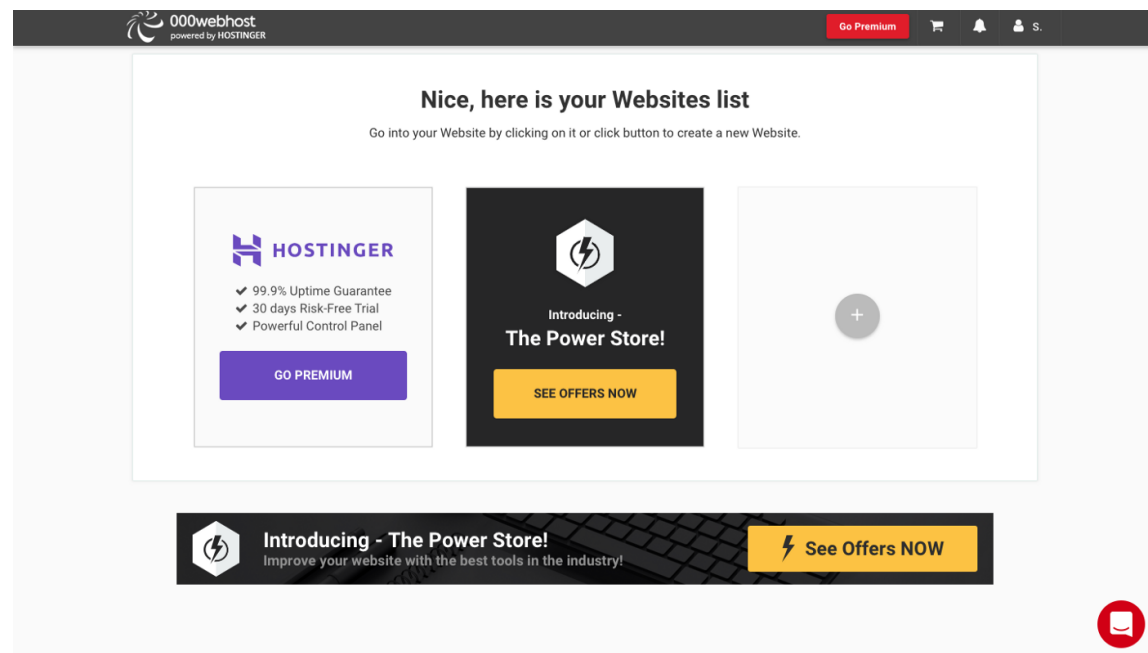
**Free Web Hosting with PHP, MySQL and cPanel, No Ads**

Free Web Hosting with PHP, MySQL, free Website Builder, cPanel and no ads.  
Almost unlimited free website hosting and...

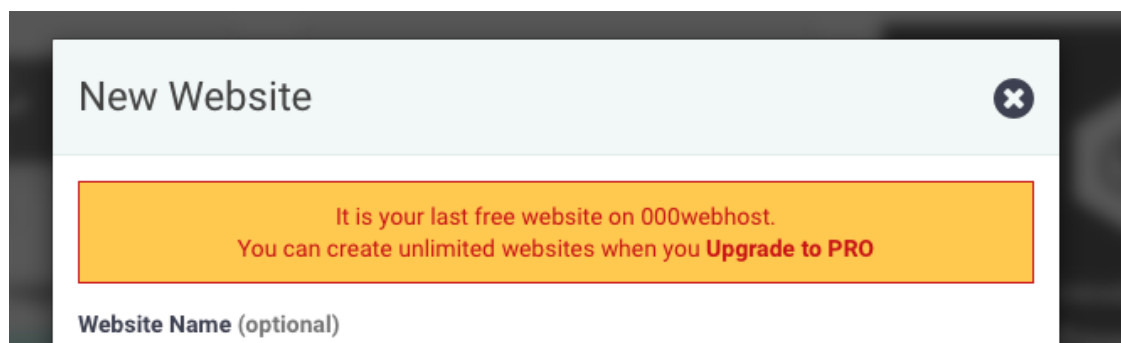
in.000webhost.com

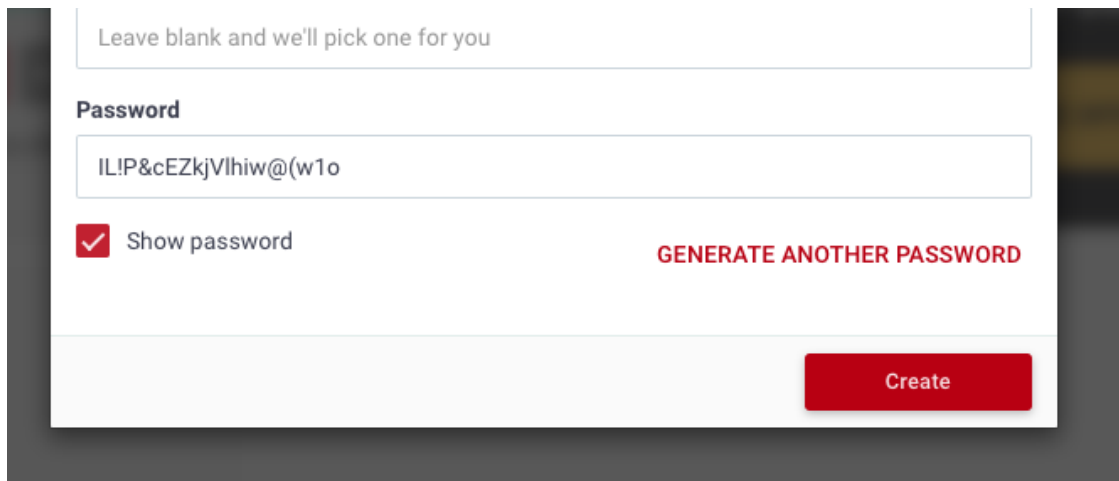
### 2. configure the webserver

After you created your free account, you will br presented with a scree like this.



press the + icon to add your web server, then configure your server





Leave blank and we'll pick one for you

**Password**

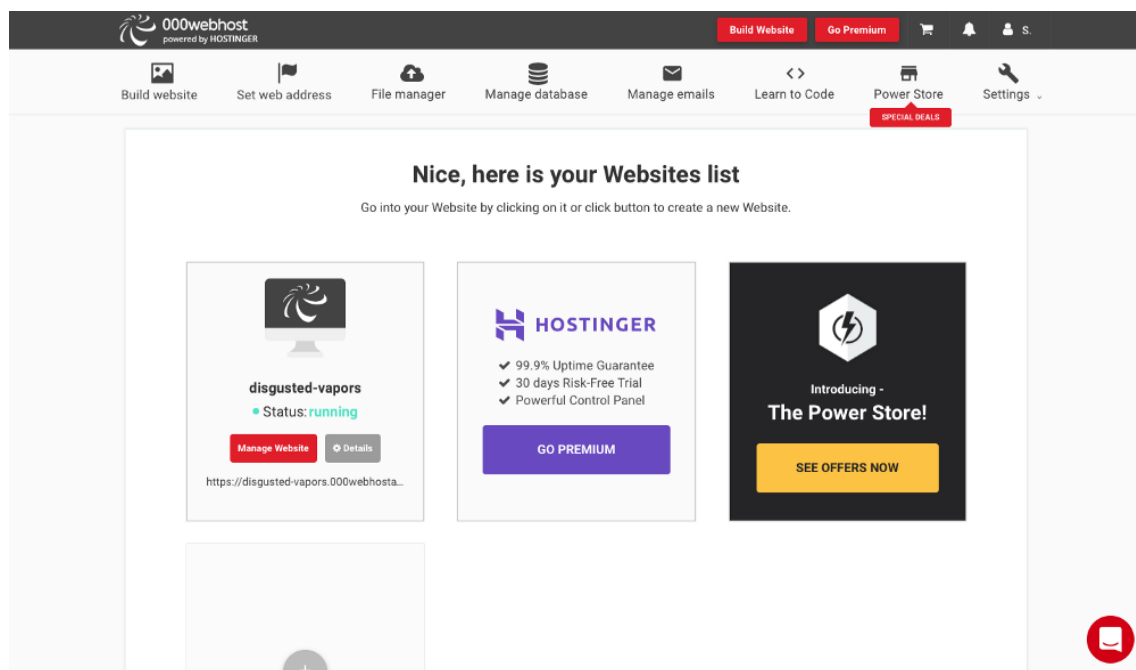
IL!P&cEZkjVlhiw@(w1o

☒ Show password

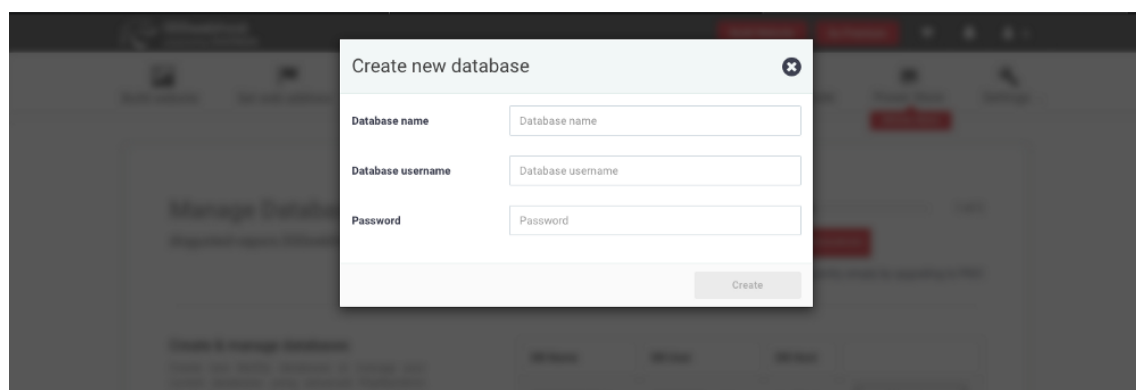
**GENERATE ANOTHER PASSWORD**

**Create**

you can provide a domain name if you wish to, but you can just accept the defaults as well as just click create



This is the screen you get after you created your web server, now go ahead and click on manage database icon at the top of the screen



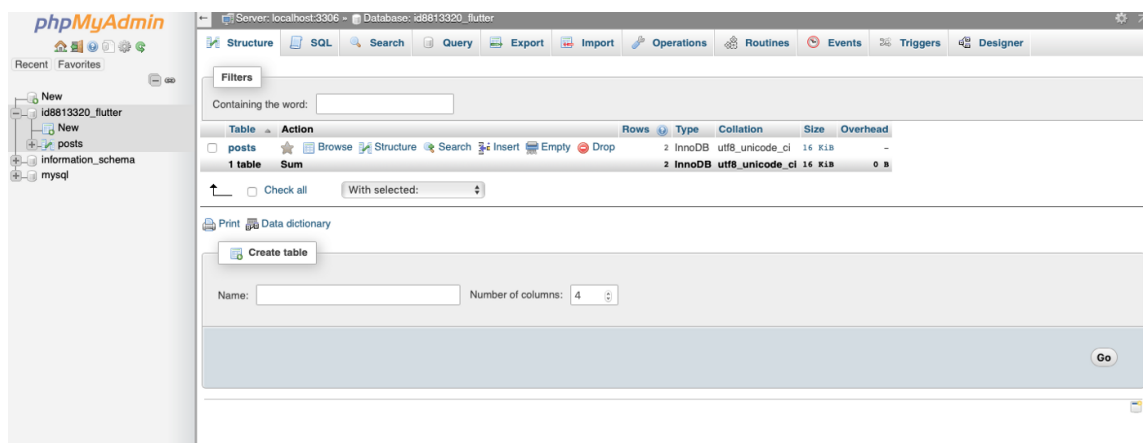


now you will have to create a database name, a username (for the database) and finally a password, make user you save this information because we are going to need it afterward

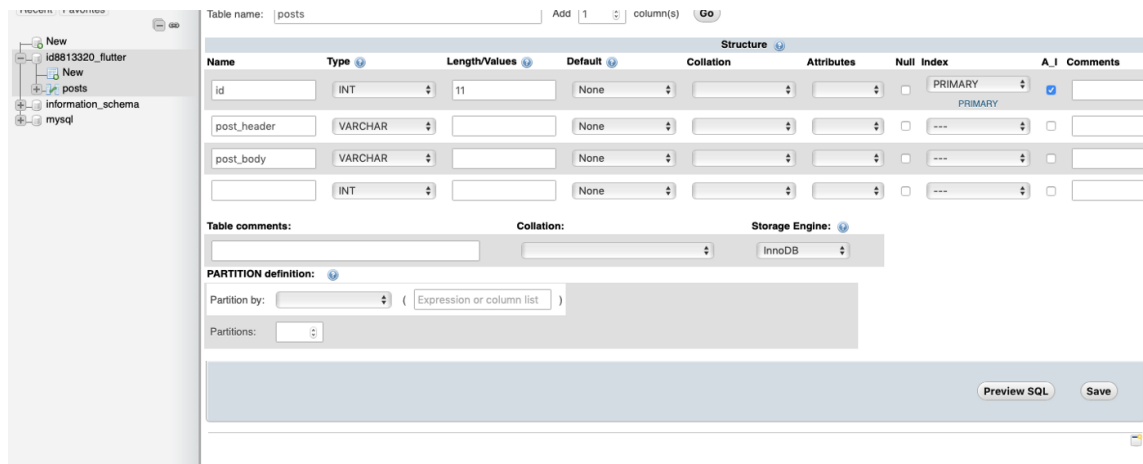
DB Name	DB User	DB Host	
id8813320_flutter	id8813320_santos	localhost	<div>Manage ▾</div> <div>           PhpMyAdmin            Change Password            Delete DB         </div> <div>           • DB Size: 0 MB            • DB Tables: 0         </div>

Now, after your database is created, go ahead and hit **Manage -> phpmyadmin**

### 3. now lets create a table on your database



Select the database you just created and then hit **new** to create a new table on your database



user the above attributes on the table that you are going to create, and finally insert some dummy values into the table, those values are the ones that we will retrieve from our flutter app.

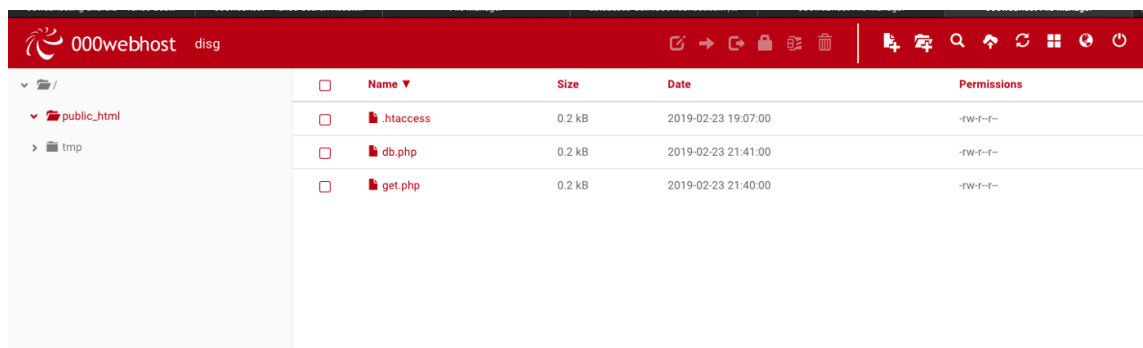
After, you inserted the dummy data this is how the table will look like

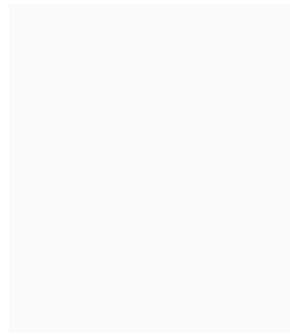


## Now let's connect our app to our server

To do that we will need to create a file that is going to handle the connection with the server.

From the 00webhost home page select **file manager**, and the click **upload file now**, from there you will be redirected to a file manager page, on that page you will create two files, one called *db.php* and the other called *get.php*, and your final output will look like this.





now select the file **db.php**, and edit its contents, the code will look like this:

```
<?php

$dns = 'mysql:host=localhost;dbname=enter-database-name';
$user = 'enter-username';
$password = 'enter_the_password_of_your_database';

try{
    $db = new PDO ($dns, $user, $pass);
} catch( PDOException $e){
    $error = $e->getMessage();
    echo $error;
}
```

*Make sure to enter the correct credentials, or else you will get an error, from the home page, go to the **manage databases** window to get the correct credentials like database name and user name! **\$db = new PDO (\$dns, \$user, \$pass);** will create a PDO object and will establish the db connection, in case there is an error it will be thrown to the catch block*

now select the **get.php** file, and edit its contents, the code is:

```
<?php
require_once('db.php');
$query = 'SELECT * FROM posts';
$stmt = $db->prepare($query);
$stmt->execute();
$row = $stmt->fetch(PDO::FETCH_ASSOC);
echo json_encode($row);
```

*here we first require the db.php file, to establish connection with our database, then we create a variable **\$query** that is going to query our database table, then we use the PDO command **\$stmt = \$db->prepare(\$query)** to make sure that our query is error free, and finally we use **\$stmt->execute();** to execute the query, then **\$row = \$stmt->fetch(PDO::FETCH\_ASSOC);** will assign to the variable row the content of the table in the form of an associative array, **echo json\_encode(\$row);** will display the response in the json format.*

## Now lets create our flutter app

we will create a simple flutter app with the following code

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Server',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.pink,
      ),
      home: MyHomePage(title: 'Flutter Server App'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);
  final String title;

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
        centerTitle: true,
      ),
    );
  }
}
```

The above code will give you the basic app structure

Lets create the method to get the contents of our table on the server

```
Future getData() async{
  var url = 'https://disgusted-vapors.000webhostapp.com/get.php';
  http.Response response = await http.get(url);
  var data = jsonDecode(response.body);
}
```

```
print(data.toString());
}
```

*This code will get the content from our table and print it on the terminal, note that the method must be of type future and it must be asynchronous,*

*the url, that you have to provide on your code is your url/get.php*

now we will call the method from the initial state

```
@override
void initState() {
  getData();
}
```

the complete code will look like this

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Server',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.pink,
      ),
      home: MyHomePage(title: 'Flutter Server App'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);
  final String title;

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
        centerTitle: true,
```



```
    ),  
  );  
}  
  
Future getData() async{  
  var url = 'https://disgusted-vapors.000webhostapp.com/get.php';  
  http.Response response = await http.get(url);  
  var data = jsonDecode(response.body);  
  print(data.toString());  
}  
  
@override  
void initState() {  
  getData();  
}  
}
```

Thank you for reading!

my git hub: <https://github.com/Santos-Enoque>

linked in: <https://www.linkedin.com/in/santos-enoque/>

my youtube channel: <https://www.youtube.com/c/santosenoque>

[Flutter](#)   [Flutter Php](#)   [Flutter Article](#)   [Flutter Sql](#)

[About](#)   [Help](#)   [Legal](#)

Get the Medium app

