[http://www.startutorial.com/img/logo.png](http://www.startutorial.com/)

* [Home](http://www.startutorial.com/)
* [Tutorials](http://www.startutorial.com/articles/all)
* [PHP Books](http://www.startutorial.com/homes/books)
* [About](http://www.startutorial.com/homes/about)

Top of Form





Bottom of Form

# PHP CRUD Tutorial (part 1)

[**25 Comments**](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1#disqus_thread)

This is the part 1 of 3 in PHP CRUD tutorial.

* PHP CRUD tutorial (part 1)
* [PHP CRUD tutorial (part 2)](http://www.startutorial.com/articles/view/php-crud-tutorial-part-2)
* [PHP CRUD tutorial (part 3)](http://www.startutorial.com/articles/view/php-crud-tutorial-part-3/)

Creating CRUD grid is a very common task in web development (CRUD stands for Create/Read/Update/Delete). If you are a senior web developer, you must have created plenty of CRUD grids already. They maybe exist in a content management system, an inventory management system, or accounting software. If you just started web development, you are certainly going to experience lots of CRUD grids' creation work in your later career.

The main purpose of a CRUD grid is that enables users create/read/update/delete data. Normally data is stored in MySQL Database.PHP will be the server-side language that manipulates MySQL Database tables to give front-end users power to perform CRUD actions.

In this tutorial series, we will go through steps of a creating PHP CRUD grid. We want to demonstrate how PHP as a server side language, communicates with backend MySQL, and meanwhile renders front-end HTML. We hope you can learn something from this tutorial.

## Table Of Content

1. [Creating a sample Database table](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1#sec1)
2. [Connecting to Database](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1#sec2)
3. [Create a Twitter Bootstrap powered grid](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1#sec3)
4. [To be continued](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1#sec4)

###### 1. Creating a sample Database table

In this tutorial, we will work on a simple Database table as below. After this tutorial, you should use the idea here to create CRUD grid on your own Database tables.

Import the Database table below to your own MySQL Database.

[?](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | CREATE TABLE  `customers` (  `id` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY ,  `name` VARCHAR( 100 ) NOT NULL ,  `email` VARCHAR( 100 ) NOT NULL ,  `mobile` VARCHAR( 100 ) NOT NULL  ) ENGINE = INNODB; |

As you can see, this is a very simple table for tracking customers' information (name, email address and mobile number). And we prefer an auto incremental primary key (id).

###### 2. Connecting to Database

Create a PHP file "database.php"; this file contains a PHP class named "Database". Throughout this application, Database handles all the stuff related to database connections, such as connecting and disconnecting.

[?](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | <?php  class Database  {      private static $dbName = 'crud\_tutorial' ;      private static $dbHost = 'localhost' ;      private static $dbUsername = 'root';      private static $dbUserPassword = 'root';        private static $cont  = null;        public function \_\_construct() {          die('Init function is not allowed');      }        public static function connect()      {         // One connection through whole application         if ( null == self::$cont )         {          try          {            self::$cont =  new PDO( "mysql:host=".self::$dbHost.";"."dbname=".self::$dbName, self::$dbUsername, self::$dbUserPassword);          }          catch(PDOException $e)          {            die($e->getMessage());          }         }         return self::$cont;      }        public static function disconnect()      {          self::$cont = null;      }  }  ?> |

As you can see, we are using PDO for database access. There are a lot of benefits of using PDO. One of the most significant benefits is that it provides a uniform method of access to multiple databases.

To use this class, you will need to supply correct values for $dbName, $dbHost, $dbUsername, $dbUserPassword.

* **$dbName**: Database name which you use to store 'customers' table.
* **$dbHost**: Database host, this is normally "localhost".
* **$dbUsername**: Database username.
* **$dbUserPassword**: Database user's password.

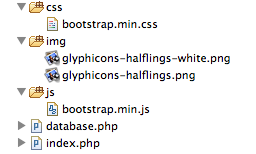
Let us take a look at three functions of this class:

* **\_\_construct()**: This is the constructor of class Database. Since it is a static class, initialization of this class is not allowed. To prevent misuse of the class, we use a "die" function to remind users.
* **connect**: This is the main function of this class. It uses singleton pattern to make sure only one PDO connection exist across the whole application. Since it is a static method. We use **Database::connect()** to create a connection.
* **disconnect**: Disconnect from database. It simply sets connection to NULL. We need to call this function to close connection.

###### 3. Create a Twitter Bootstrap powered grid

Here comes the grid without CRUD capabilities. Because CRUD operation can only be performed when there is a grid. We firstly need to build a grid. From there we can subsequently add "Create" page, "Read" page, "Update" page and finally "Delete" page.

Head to [Bootstrap official page](http://twitter.github.io/bootstrap/), and download a copy. In this tutorial, we are using version 2.3.2. After that, create a PHP file "index.php".

Current file structure should look like below if you have followed the steps correctly:   


Now open file "index.php". And copy the content below:

[?](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="utf-8">      <link   href="css/bootstrap.min.css" rel="stylesheet">      <script src="js/bootstrap.min.js"></script>  </head>    <body>      <div class="container">              <div class="row">                  <h3>PHP CRUD Grid</h3>              </div>              <div class="row">                  <table class="table table-striped table-bordered">                    <thead>                      <tr>                        <th>Name</th>                        <th>Email Address</th>                        <th>Mobile Number</th>                      </tr>                    </thead>                    <tbody>                    <?php                     include 'database.php';                     $pdo = Database::connect();                     $sql = 'SELECT \* FROM customers ORDER BY id DESC';                     foreach ($pdo->query($sql) as $row) {                              echo '<tr>';                              echo '<td>'. $row['name'] . '</td>';                              echo '<td>'. $row['email'] . '</td>';                              echo '<td>'. $row['mobile'] . '</td>';                              echo '</tr>';                     }                     Database::disconnect();                    ?>                    </tbody>              </table>          </div>      </div> <!-- /container -->    </body>  </html> |

Let us dig into the codes.

* <head> part of this file is straightforward, we include Bootstrap's CSS and JavaScript files.
* Line 15 to 38 is the main part of the file. It is where we retrieve data from database and show it on the grid. Let dig deep into each lines carefully.
* We firstly create a table with headers corresponding to database table "customers"'s fields. Which includes "Name", "Email Address", "Mobile Number":

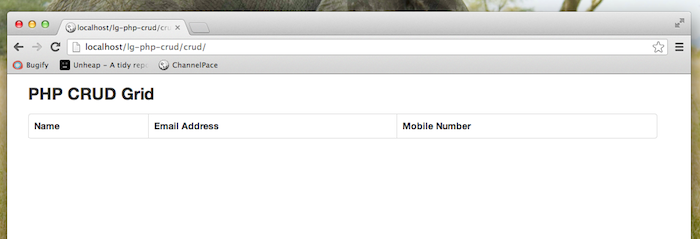
[?](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <thead>             <tr>               <th>Name</th>               <th>Email Address</th>               <th>Mobile Number</th>             </tr>  </thead> |

* Then we include "database.php", create a PDO connection to database, and use a general "SELECT" statement to retrieve data. Lastly, we loop through each row to print content. Do not forget to close the connection as we mentioned at the beginning.

[?](http://www.startutorial.com/articles/view/php-crud-tutorial-part-1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <?php             include 'database.php';             $pdo = Database::connect();             $sql = 'SELECT \* FROM customers ORDER BY id DESC';             foreach ($pdo->query($sql) as $row) {                      echo '<tr>';                      echo '<td>'. $row['name'] . '</td>';                      echo '<td>'. $row['email'] . '</td>';                      echo '<td>'. $row['mobile'] . '</td>';                      echo '</tr>';             }             Database::disconnect();  ?> |

If you have followed correctly, you should have an empty grid as below if you navigate to "index.php" from browser:  


The grid is empty because there is no actual data inside "customers" table, to test if it is actually working, you can manually insert data into "customers" table. It should show them on the grid.

###### 4. To be continued

As this post is getting too long, let us move on to next post. In [PHP CRUD tutorial part 2](http://www.startutorial.com/articles/view/php-crud-tutorial-part-2/), we demonstrate how to create the "Create" and "Read" pages of PHP CRUD grid.

Hopefully this simple tutorial helped you with your development.  
If you like our post, please follow us on [Twitter](http://twitter.com/startutorial) and help spread the word. We need your support to continue.   
If you have questions or find our mistakes in above tutorial, do leave a comment below to let us know.

### **Free Course:** "Guides to be an  advanced **PHP developer** in 1 Week"

This amazing course will provide you a list of organized articles to prepare yourself to be an advanced PHP developer over the next 7 days.

#### Fill out the form below to start your FREE Course

Top of Form

Email



100% Privacy. We will never spam you!

Bottom of Form

© Star Tutorial 2015