



```
1  <!-- Db.php -->
2  <!-- Class DB ... -->
3  class DB{
4      private $db;
5      /**
6       * Connect to the database
7       * @param $dbhost
8       * @param $dbname
9       * @param $dbuser
10      * @param $dbpswd
11      */
12      public function __construct($dbhost, $dbname, $dbuser, $dbpswd){
13          $this->db = new PDO( dsn: 'mysql:host='.$dbhost.';dbname='.$dbname.';charset=utf8', $dbuser, $dbpswd);
14      }
15      /**
16       * Execute an SQL query and return the result (prepared request or not)
17       * @param string $request SQL query
18       * @param array|null $values Optional values
19       * @return PDOStatement
20      */
21      private function exec($request, $values = null){
22          $req = $this->db->prepare($request);
23          $req->execute($values);
24          return $req;
25      }
26      /**
27       * Define the fetchMode
28       * @param int $fetchMode fetchMode
29      */
30      public function setFetchMode($fetchMode){
31          $this->db->setAttribute( attribute: PDO::ATTR_DEFAULT_FETCH_MODE, $fetchMode);
32      }
33      /**
34       * Execute an SQL query and return the status
35       * @param string $request SQL query
36       * @param array|null $values Optional values
37       * @return bool Result of the request
38      */
39      public function execute($request, $values = array()){
40          $results = self::exec($request, $values);
41          return ($results) ? true : false;
42      }
43      /**
44       * Execute an SQL query and return row(s) of the result
45       * @param string $request SQL query
46       * @param array|null $values Optional values
47       * @param bool $all Query with several rows or not
48       * @return array|mixed Return data
49      */
50      public function fetch($request, $values = null, $all = true) {
51          $results = self::exec($request, $values);
52          return ($all) ? $results->fetchAll() : $results->fetch();
53      }
54  }
```

Project ▾

▼ Blog [PHP_BLOG_MVC_SAMP]

▼ _classes

Articles.php

Authors.php

Categories.php

Db_class.php

tables.sql

▼ _config

config.php

db.php

► _functions

► assets

▼ controllers

contact_controller.php

error_controller.php

home_controller.php

test_controller.php

► models

▼ views

► includes

contact_view.php

error_view.php

home_view.php

test_view.php

.htaccess

index.php

► External Libraries

Scratches and Consoles

Db_class.php ×

test_controller.php ×

db.php ×

```
1 <?php
2
3 //Fetch sans paramètre.
4 $articleAssoc = $DBFetchAssoc->fetch( request: 'SELECT * FROM articles');
5 debug($articleAssoc);
6
7
8 //Fetch avec paramètre.
9 $articleAssoc = $DBFetchAssoc->fetch( request: 'SELECT * FROM articles WHERE category_id = ?', array(1));
10 debug($articleAssoc);
11
12
13 //Pour ne pas faire un FetchAll mais un fetch simple simple il faut passer false en troisieme paramètre.
14 $articleAssoc = $DBFetchAssoc->fetch( request: 'SELECT * FROM articles', [], all: false);
15 debug($articleAssoc);
16
17 // Nous renvoie true si la requete est bien executable ou a été executer et false dans le cas contraire.
18 $result = $DB->execute( request: 'SELECT * FROM articles');
19 debug($result);
20
21
```

Project ▾

Blog [PHP_BLOG_MVC_SAMP]

_classes

Articles.php

Authors.php

Categories.php

Db_class.php

tables.sql

_config

config.php

db.php

_functions

assets

controllers

contact_controller.php

error_controller.php

home_controller.php

test_controller.php

models

views

includes

contact_view.php

error_view.php

home_view.php

test_view.php

.htaccess

index.php

External Libraries

Scratches and Consoles

Db_class.php ×

test_controller.php ×

db.php ×

```
1 <?php
2
3
4 require_once 'classes/Db_class.php';
5
6 // $DB simple, sans fetch mode par défaut de paramétré.
7 $DB = new DB( dbhost: DATABASE_HOST, dbname: DATABASE_NAME, dbuser: DATABASE_USER, dbpswd: DATABASE_PASSWORD);
8
9 //DB fetch Associatif.
10 $DBFetchAssoc = new DB( dbhost: DATABASE_HOST, dbname: DATABASE_NAME, dbuser: DATABASE_USER, dbpswd: DATABASE_PASSWORD);
11 $DBFetchAssoc->setFetchMode( fetchMode: PDO::FETCH_ASSOC);
12
13 //DB fetch Obj
14 $DBFetchOBJ = new DB( dbhost: DATABASE_HOST, dbname: DATABASE_NAME, dbuser: DATABASE_USER, dbpswd: DATABASE_PASSWORD);
15 $DBFetchOBJ->setFetchMode( fetchMode: PDO::FETCH_OBJ);
16
17 
```