

Ejercicio PHP/Base de datos

Repositorio: <https://github.com/mdloma/apiphp.git>

Descripción general

Crear un servicio REST para la gestión de tareas con PHP y MYSQL.

Requisitos mínimos

- ◆ Debe permitir realizar las operaciones CRUD sobre las tareas (listado, creación, obtención, edición y borrado)
- ◆ Las tareas deben tener al menos los campos: id, título, categoría, descripción, fecha creación y fecha actualización
- ◆ El título es el único campo obligatorio, el id será autogenerado, la categoría por defecto estará establecida a “General” y la descripción a una cadena vacía
- ◆ El servicio debe devolver al menos errores 404 al intentar obtener, editar o borrar una tarea que no exista y errores de validación como mínimo en la creación de una tarea si alguno de los datos no son correctos

MYSQL

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SHEMAS' tree with the 'gestion_tareas' database selected. The 'tareas' table is highlighted under the 'Tables' folder. The main window shows the 'Query 1' editor with the SQL query: `SELECT * FROM gestion_tareas.tareas;`. Below the editor, the 'Result Grid' displays the data for the 'tareas' table. The table has 7 rows and 7 columns: id, titulo, descripcion, fecha_creacion, fecha_actualizacion, and categoria. The bottom right corner shows the 'Output' tab with the message: '1 17:12:08 SELECT * FROM gestion_tareas.tareas LIMIT 0, 1000 6 row(s) returned'.

	id	titulo	descripcion	fecha_creacion	fecha_actualizacion	categoria
1	TAREA 1	Alta de usuarios		2022-10-28 13:00:00	2022-10-28 13:24:01	
2	TAREA 2	Confirmación usuarios		2022-10-28 13:24:01	2022-10-28 13:24:01	
3	TAREA 3	Formulario		2022-10-28 13:00:00	2022-10-28 13:28:29	
4	TAREA 4	Gestión usuarios		2022-10-28 13:31:00	2022-10-28 13:32:38	
5	TAREA 5	Inspección usuarios		2022-10-28 13:34:00	2022-10-28 13:34:56	
7	TAREA 7			2022-10-28 13:37:52	2022-10-28 13:37:52	

Result Grid						
Filter Rows:						
Edit: Export/Import: Wrap Cell Content:						
	id	titulo	categoria	descripcion	fecha_creacion	fecha_actualizacion
▶	1	TAREA 1	General	Alta de usuarios	2022-10-28 13:00:00	2022-10-28 13:24:01
	2	TAREA 2	Admin	Confirmación usuarios	2022-10-28 13:24:01	2022-10-28 13:24:01
	3	TAREA 3	Aux	Formulario	2022-10-28 13:00:00	2022-10-28 13:28:29
	4	TAREA 4	Tecnico	Gestión usuarios	2022-10-28 13:31:00	2022-10-28 13:32:38
	5	TAREA 5	General	Inspección usuarios	2022-10-28 13:34:00	2022-10-28 13:34:56
	NULL	NULL	NULL	NULL	NULL	NULL

CRUD

#	Time	Action	Message	Duration / Fetch
✓ 1	18:22:28	SELECT * FROM gestion_tareas.tareas LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
✓ 2	18:24:04	SELECT * FROM gestion_tareas.tareas LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
✓ 3	19:12:54	DELETE FROM gestion_tareas.tareas WHERE id=7	1 row(s) affected	0.000 sec
✗ 4	19:16:40	UPDATE tareas SET titulo= "Prueba", descripcion= "Prueba", categoria= "Prueba" WHERE id=6	Error Code: 1146. Table 'database.tareas' doesn't exist	0.000 sec
✓ 5	19:17:23	UPDATE gestion_tareas.tareas SET titulo= "Prueba", descripcion= "Prueba", categoria= "Prue...	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.016 sec

POSTMAN

GET http://localhost/apiphp
Overview
New Collection
GET http://localhost/apiphp
+ ...
No Environment

http://localhost/apiphp/gestion_tareas
Save

GET http://localhost/apiphp/gestion_tareas
Send

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body
Cookies
Headers (7)
Test Results
Status: 200 OK Time: 57 ms Size: 1.2 KB Save Response

Pretty
Raw
Preview
Visualize
JSON

```

1  {
2    "id": 1,
3    "titulo": "TAREA 1",
4    "descripcion": "Alta de usuarios",
5    "fecha_creacion": "2022-10-28 13:00:00",
6    "fecha_actualizacion": "2022-10-28 13:24:01",
7    "categoria": ""
8  },
9  {
10   "id": 2,
11   "titulo": "TAREA 2",
12   "descripcion": "Confirmación usuarios",
13   "fecha_creacion": "2022-10-28 13:24:01",
14   "fecha_actualizacion": "2022-10-28 13:24:01",
15   "categoria": ""
16 },
17 }

```

Cookies
Pantalla requests
Postman
Runner
Trash

```
16     "categoria": ""
17   },
18   {
19     "id": 3,
20     "titulo": "TAREA 3",
21     "descripcion": "Formulario",
22     "fecha_creacion": "2022-10-28 13:00:00",
23     "fecha_actualizacion": "2022-10-28 13:28:29",
24     "categoria": ""
25   },
26   {
27     "id": 4,
28     "titulo": "TAREA 4",
29     "descripcion": "Gestión usuarios",
30     "fecha_creacion": "2022-10-28 13:31:00",
31     "fecha_actualizacion": "2022-10-28 13:32:38",
32     "categoria": ""
33   },
34   {
35     "id": 5,
36     "titulo": "TAREA 5",
37     "descripcion": "Inspección usuarios",
38     "fecha_creacion": "2022-10-28 13:34:00",
39     "fecha_actualizacion": "2022-10-28 13:34:56",
40     "categoria": ""
41   },
42   {
43     "id": 7,
44     "titulo": "TAREA 7",
45     "descripcion": "",
46     "fecha_creacion": "2022-10-28 13:37:52",
47     "fecha_actualizacion": "2022-10-28 13:37:52",
48     "categoria": ""
49   }
50 ]
```

DELETE

http://localhost/apiphp/gestion_tareas/users?id=6

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	6			
	Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

404 Not Found

18 ms

300 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "message": "La tarea no existe"
3  }
```

PHP

index.php

```
index.php
1  <?php
2
3  header('Content-Type: application/json; charset=utf-8');
4
5  require_once __DIR__ . '/db.php';
6
7  require_once __DIR__ . '/functions.php';
8
9  $uri = $_SERVER['REQUEST_URI'];
10
11 $httpVerb = $_SERVER['REQUEST_METHOD'];
12
13
14 if ($httpVerb === 'POST') {
15     require_once __DIR__ . '/controllers/post.controller.php';
16 }
17
18
19 if ($httpVerb === 'GET') {
20     require_once __DIR__ . '/controllers/get.controller.php';
21 }
22
23 if ($httpVerb === 'PUT') {
24     require_once __DIR__ . '/controllers/put.controller.php';
25 }
26
27
28 if ($httpVerb === 'DELETE') {
```

functions.php

functions.php

```
<?php

function getRequestBody(){
    $json = file_get_contents('php://input');

    return json_decode($json, true);
}

function insertTarea($conn, $tareaData){
    $insertTarea = [
        ':titulo' => $tareaData['titulo'],
        ':descripcion' => $tareaData['descripcion'],
        ':categoria' => $tareaData['categoria'],
        ':fecha_creacion' => (new DateTime())->format('Y-m-d H:i:s'),
        ':fecha_actualizacion' => (new DateTime())->format('Y-m-d H:i:s')
    ];

    $insertSQL = "INSERT INTO tareas ( titulo, categoria ,descripcion,
    fecha_creacion, fecha_ actualizacion) VALUES (:titulo,:categoria,
    :descripcion, :fecha_creacion,:fecha_actualizacion)";

    $query = $conn->prepare($insertSQL);

    try{

        if($query->execute($insertTarea)) {
            return $conn->lastInsertId();
        }
    }catch(Exception $e){
        return $e->getMessage();
    }
}

function getTarea($conn, $id){
    $tareaSQL = "SELECT * FROM tareas WHERE id=:id";
    $query = $conn->prepare($tareaSQL);

    $query->setFetchMode(PDO::FETCH_ASSOC);

    $query->execute([':id' => $id]);

    $tareas = $query->fetchAll();

    if(count($tareas) === 0){
        return null;
    }

    return $tareas[0];
}
```

```

function getTareaList($conn){
    $usersSQL = "SELECT * FROM tareas ORDER BY titulo ASC";
    $query = $conn->prepare($tareaSQL);

    $query->setFetchMode(PDO::FETCH_ASSOC);

    $query->execute();

    return $query->fetchAll();
}

function editTarea($conn, $tareaData, $id){
    $updateTarea = [
        ':titulo' => $tareaData['titulo'],
        ':categoria' => $tareaData['categoria'],
        ':descripcion' => $tareaData['descripcion'],
        ':id' => $id
    ];

    $updateSQL = "UPDATE tareas SET titulo=:titulo, descripcion=:descripcion,
    categoria=:categoria, WHERE id=:id";

    $query = $conn->prepare($updateSQL);

    try{

        if($query->execute($updateTarea)) {
            return $id;
        }
    }catch(Exception $e){
        return null;
    }
}

function deleteTarea($conn, $id){
    $deleteTarea = [
        ':id' => $id
    ];

    $deleteSQL = "DELETE FROM gestion_tareas.tareas WHERE id=:id";

    $query = $conn->prepare($deleteSQL);

    try{

        if($query->execute($deleteTarea)) {
            return $query->rowCount();

```

```

        }
    }catch(Exception $e){
        return 0;
    }
}

```

Exercices-db.sql

```
Get Started index.php functions.php put.controller.php .htaccess
exercise-db.sql
1
2 DROP TABLE IF EXISTS `tareas`;
3
4 CREATE TABLE `usuarios` (
5     `id` int NOT NULL AUTO_INCREMENT,
6     `titulo` varchar(45) NOT NULL,
7     `categoria` varchar(45) NOT NULL,
8     `descripcion` varchar(250) NOT NULL DEFAULT '',
9     `fecha_creacion` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
10    PRIMARY KEY (`id`),
11
12 ) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8mb4
13 COLLATE=utf8mb4_0900_ai_ci;
14
15
16 LOCK TABLES `tareas` WRITE;
17
18 INSERT INTO `tareas` VALUES (6, 'TAREA 1','General','Alta usuario','2022-10-19
19 11:45:26'),(8,'TAREA 8','General','2022-10-19 11:45:26')(9,'TAREA 9',
20 'Confirmacion usuarios','2022-10-19 12:03:30');
21
22 UNLOCK TABLES;
```

db.php

```
db.php
1 <?php
2
3
4 $servername = "localhost";
5 $username = "root";
6 $password = "";
7 $dbname = "gestion_tareas";
8 $dbport = 3306;
9
10
11 $conn = null;
12
13 try {
14     $conn = new PDO("mysql:host=$servername;dbname=$dbname;port=$dbport",
15         $username, $password);
16     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
17
18 } catch(PDOException $e) {
19
20
21     http_response_code(500);
22     echo '{"message": "Se ha producido un error interno. Ponte en contacto con el
23     administrador si el problema persiste."}';
24     return;
25 }
```

Controllers

Delete.controller.php

```
controllers > delete.controller.php

$uriParts = explode('/', substr($uri, 1));

$affectedRows = deleteTarea($conn, $uriParts[1]);

if($affectedRows === 0){

    http_response_code(404);
    echo '{"message": "La tarea no existe"}';
    return;
}

http_response_code(204);
echo '';
return;
```

Get.controller.php

```
rollers > get.controller.php

<?php

    $tareas = getTareaList($conn);

    http_response_code(200);
    echo json_encode($tareas);
    return;

$uriParts = explode('/', substr($uri, 0));

    $tarea = getTarea($conn, $uriParts[0]);
    print($uriParts[0]);

    if(!$tarea){
        //devolvemos un error
        http_response_code(404);
        echo '{"message": "La tarea no existe"}';
        return;
    }

    http_response_code(200);
    echo json_encode($tarea);
    return;
```


Post.controller.php

controllers > post.controller.php

```
1 <?php
2
3 $tareaData = getRequestBody();
4
5 if(!is_array($tareaData)){
6     http_response_code(400);
7     echo '{"message": "Petición mal formada"}';
8     return;
9 }
10
11 if(!key_exists('titulo',$tareaData)){
12     http_response_code(400);
13     echo '{"message": "Es necesario especificar el titulo de la tarea"}';
14     return;
15 }
16
17 if(!key_exists('categoria',$tareaData)){
18     http_response_code(400);
19     echo '{"message": "Es necesario que especifiques una categoria de la
20     tarea"}';
21     return;
22 }
23
24 if(!key_exists('descripcion',$tareaData)){
25     http_response_code(400);
26     echo '{"message": "Es necesario que especifiques una descripcion de la
27     tarea"}';
28     return;
29 }
```

Put.controller.php

controllers > put.controller.php

```
1 <?php
2
3 $tareaData = getRequestBody();
4
5 if(!is_array($tareaData)){
6
7     http_response_code(400);
8     echo '{"message": "Petición mal formada"}';
9     return;
10 }
11
12 if(!key_exists('titulo',$tareaData)){
13     http_response_code(400);
14     echo '{"message": "Es necesario especificar el titulo de la tarea"}';
15     return;
16 }
17
18 if(!key_exists('categoria',$tareaData)){
19     http_response_code(400);
20     echo '{"message": "Es necesario que especifiques una categoria de la tarea"}';
21     return;
22 }
23
24
25 $uriParts = explode('/',substr($uri,1));
26
27
28 $id = editUser($conn, $tareaData, $uriParts[0]);
```

.htaccess

```
.htaccess
1 RewriteEngine On
2 RewriteCond %{REQUEST_FILENAME} !-f
3 RewriteRule ^(.*) index.php
```