

## DOCUMENTACION API



## Indice

CONEXION.PHP.....	3
INICIO.PHP.....	3
API.JS .....	4
REGISTRAR USUARIOS:.....	4
EDITAR PERFIL: .....	5
ELIMINAR USUARIO:.....	5
RUTAS.PHP .....	6
RUTAS1.PHP .....	7

## PARA LA API TENGO UN TOTAL DE 5 ARCHIVOS PHP

### CONEXION.PHP

Aquí lo que hago es conectarme a la base de datos, al usuario de esta base y a su contraseña , desde esta conexión se van a conectar las demás apis

También hay una función en la que comprueba que esto existe y esta bien hecho

```
class Conexion extends mysqli {
    private $host = "localhost";
    private $db = "proyecto2trim";
    private $user = "proyecto2trim";
    private $pass = "proyecto2trim";

    public function __construct() {
        try {
            parent::__construct($this->host, $this->user, $this->pass, $this->db);
        } catch (mysqli_sql_exception $e) {
            echo "ERROR: {$e->getMessage()}";
            // header("HTTP/1.1 400 Bad Request");
            exit;
        }
    }
}
```

### INICIO.PHP

Aquí tengo 1 sola función además de la conexión a la base de datos, la función es para iniciar sesión y uso un paquete para crear el JWT que le instala con:

```
composer require firebase/php-jwt
composer require paragonie/sodium_compat
```

Y se añade así:

```
use Firebase\JWT\JWT;
use Firebase\JWT\Key;
```

Lo primero compruebo que el método recogido es post y recojo el JSON recibido con esta línea de código:

```
$json = file_get_contents('php://input');
```

después hago un if para comprobar que esta esta variable \$json , la decodifico y hago una nueva variable que es \$user y de ahí saco el nombre de usuario y la contraseña

Luego hago la consulta :

```
$sql = "SELECT id FROM `users` WHERE username like '{$username}' and pass like '{$pass}'";
```

Por ultimo hago un try para controlar los errores en el que tiene una variable key y payload para el JWT , también tengo el result de la consulta y mando en json el id, el token , el username y un mensaje de 201 Created

```
$key = 'example_key';
$payload = [
    'username' => $username,
    'id' => $sql,
```

```
];
$jwt = JWT::encode($payload, $key, 'HS256');
$result = $db->query($sql);
$alumnos = [
    "id" => $result->fetch_all(MYSQLI_ASSOC),
    "token" => $jwt,
    "username" => $username
];
header("HTTP/1.1 201 Created");
echo json_encode($alumnos);
```

## API.JS

Lo primero me conecto a la base de datos y llamo los paquetes para el JWT y aquí tengo 3 funciones

### REGISTRAR USUARIOS:

Lo primero compruebo que el método recogido es post y recojo el JSON recibido con esta línea de código:

```
$json = file_get_contents('php://input');
```

después hago un if para comprobar que esta variable \$json, la decodifico y hago una nueva variable que es \$user y de ahí saco todos los campos necesarios

Luego hago la consulta :

```
$sql = "INSERT INTO users (fullname, username, email, pass, height, weight, birthday,
activities) VALUES('$fullname', '$username', '$email', '$pass', '$height', '$weight', '$birthday',
'$activities')";
```

Luego hago otra consulta para comprobar que no existe un usuario con el nombre de usuario que se ha introducido

```
$sql2 = "SELECT * FROM users WHERE username = '$username'";
```

Luego en el try hago un if y compruebo que si la query de \$sql es mayor a 0 devuelve un success false y no inserta y si no lo inserta bien y devuelve el id

```
$usuarios = $db->query($sql2)->fetch_all(MYSQLI_ASSOC);
if (count($usuarios) > 0) {
    header("HTTP/1.1 409 Username exist");
    header("Content-Type: application/json");
    echo json_encode([
        'success' => false,
        'msg' => "El nombre de usuario ya existe"
    ]);
} else {
    $db->query($sql);
    header("HTTP/1.1 201 Created");
    header("Content-Type: application/json");
    echo json_encode([
        'success' => true,
        'msg' => "El usuario se ha creado
correctamente"]);
}
```

## EDITAR PERFIL:

Lo primero compruebo que el método recogido es post y recojo el JSON recibido con estas dos líneas de código:

```
$json = file_get_contents('php://input');  
$user = json_decode($json);
```

Luego decodifico el JWT y compruebo que el username del JWT es igual al username enviado

```
$decoded = JWT::decode($user->token, new Key('example_key', 'HS256'));  
if ($user->username == $decoded->username)
```

Luego compruebo que todo a sido enviado y si no da error

```
if (  
    isset($user->fullname)  
        && isset($user->id) && isset($user->height) && isset($user->weight) &&  
    isset($user->activities)  
    ) else {  
    header("HTTP/1.1 400 Bad Request");  
}
```

Dentro de este if ya hago la consulta, recogo los campos y hago el query

```
if (  
    isset($user->fullname)  
        && isset($user->id) && isset($user->height) && isset($user->weight) &&  
    isset($user->activities)  
    ) {  
    $id = $user->id;  
    $fullname = $user->fullname;  
    $height = $user->height;  
    $weight = $user->weight;  
    $activities = $user->activities;  
    $sql = "UPDATE users SET fullname='$fullname',  
height='$height', weight='$weight',  
activities='$activities' WHERE id='$id'";  
    try {  
        $db->query($sql);  
        header("HTTP/1.1 200 OK");  
        echo json_encode($db->insert_id);  
    } catch (mysqli_sql_exception $e) {  
        header("HTTP/1.1 400 Bad Request");  
    }  
    } else {  
        header("HTTP/1.1 400 Bad Request");  
    }  
}
```

## ELIMINAR USUARIO:

Lo primero compruebo que el método recogido es post y recojo el JSON recibido con estas dos líneas de código:

```
$json = file_get_contents('php://input');  
$user = json_decode($json);
```

Luego decodifico el JWT y compruebo que el username del JWT es igual al username enviado

```
$decoded = JWT::decode($user->token, new Key('example_key', 'HS256'));  
if ($user->username == $decoded->username)
```

Si coinciden hago la consulta y elimino por el id

```
$sql = "DELETE FROM users WHERE id = '$user->id'";
```

## RUTAS.PHP

Lo primero me conecto a la base de datos y aquí solo tengo una función para sacar todas las rutas

Si el método es GET hago la sentencia sql:

```
$sql = "SELECT  
route_name,distance,max_height,min_height,start_lat,start_lon,descripcion  
FROM  
rutas WHERE 1 ";
```

Luego en el try hago los if donde filtro por nombre, altura máxima, altura mínima y finalmente codifico el resultado

```
if (isset($_GET['name'])) {  
    $name = $_GET['name'];  
    $sql .= "AND route_name LIKE '%$name%' ";  
}  
  
if (isset($_GET['min_dist'])) {  
    $min_dist = $_GET['min_dist'];  
    $sql .= "AND distance >= $min_dist ";  
}  
  
if (isset($_GET['max_dist'])) {  
    $max_dist = $_GET['max_dist'];  
    $sql .= "AND distance <= $max_dist ";  
}  
  
$result = $db->query($sql)->fetch_all(MYSQLI_ASSOC);  
header("HTTP/1.1 200 OK");  
echo json_encode($result);
```

Después tengo la función de insertar rutas que compruebo que el método recogido es post y recojo el JSON recibido con estas dos líneas de código:

```
$json = file_get_contents('php://input');
```

```
$user = json_decode($json);
```

Después recojo los datos y hago la consulta como en insertar usuarios, de que si existe una ya con ese nombre no inserte

```
$sql = "INSERT INTO `rutas` (`route_name`, `distance`,  
`max_height`, `min_height`, `pos_slope`, `neg_slope`,  
`start_lat`, `start_lon`, `descripcion`, `user`, `points`) VALUES  
('$nombre', '$distancia', '$max_height', '$min_height',  
'$pos_slope', '$neg_slope', '$start_lat', '$start_lon',  
'$descripcion', '$id', '$points');";
```

```
//CONSULTA PARA COMPROBAR QUE EXISTE  
$sql2 = "SELECT * FROM rutas WHERE route_name = '$nombre'";  
  
try {  
    //SI XSITE NO LO INSERTA SI NO, SI  
    $usuarios = $db->query($sql2)->fetch_all(MYSQLI_ASSOC);  
    if (count($usuarios) > 0) {  
        header("HTTP/1.1 409 Username exist");  
        header("Content-Type: application/json");  
        echo json_encode([  
            'success' => false,  
            'msg' => "El nombre de usuario ya existe"  
        ]);  
    } else {  
        $db->query($sql);  
        header("HTTP/1.1 201 Created");  
        header("Content-Type: application/json");  
        echo json_encode([  
            'success' => true,  
            'msg' => "El usuario se ha creado correctamente"  
        ]);  
    }  
}
```

## RUTAS1.PHP

Lo primero me conecto a la base de datos y aquí solo tengo una función para sacar todas las rutas

Si el método es GET hago la sentencia sql:

```
$sql = "SELECT  
route_name,distance,max_height,min_height,start_lat,start_lon,descripcion FROM  
rutas WHERE 1 ";
```

Luego en el try hago los if donde filtro por nombre, altura máxima, altura mínima y la diferencia con la otra es que aquí filtro también por el nombre de usuario para sacar MIS RUTAS finalmente codifico el resultado

```
if (isset($_GET['name'])) {  
    $name = $_GET['name'];  
    $sql .= "AND route_name LIKE '%$name%' ";  
}  
  
if (isset($_GET['min_dist'])) {  
    $min_dist = $_GET['min_dist'];  
    $sql .= "AND distance >= $min_dist ";  
}  
  
if (isset($_GET['max_dist'])) {  
    $max_dist = $_GET['max_dist'];  
    $sql .= "AND distance <= $max_dist ";  
}  
  
$result = $db->query($sql)->fetch_all(MYSQLI_ASSOC);  
header("HTTP/1.1 200 OK");  
echo json_encode($result);
```