

User Guide API (Create Account)

What does it do?

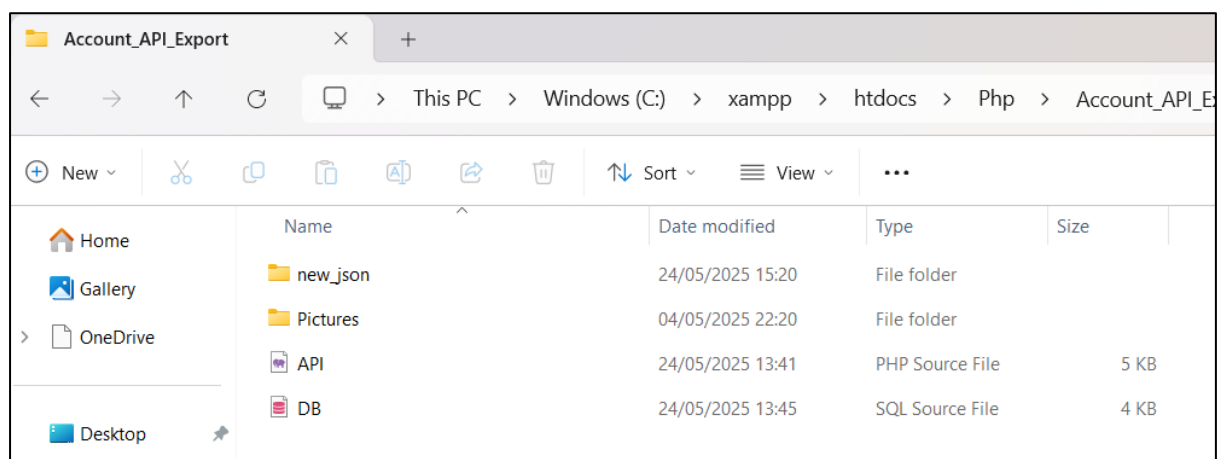
This API is designed to create user accounts. Through a front-end interface (e.g., HTML), it collects the necessary information via a form to create an account. The data is then stored in the “gamming_01” database, where it can be managed using the [POST, GET, UPDATE, and DELETE] methods.

Additionally, this API generates a log each time an account is created. The log is saved in a folder in JSON format, making it easier to manage or manipulate the files if needed.

How do I implement it?

Step 0: Unzip the Folder You Downloaded from GitHub

The **Account API Export** folder contains all the necessary files to use the API in any other project, requiring only minimal modifications to ensure compatibility.

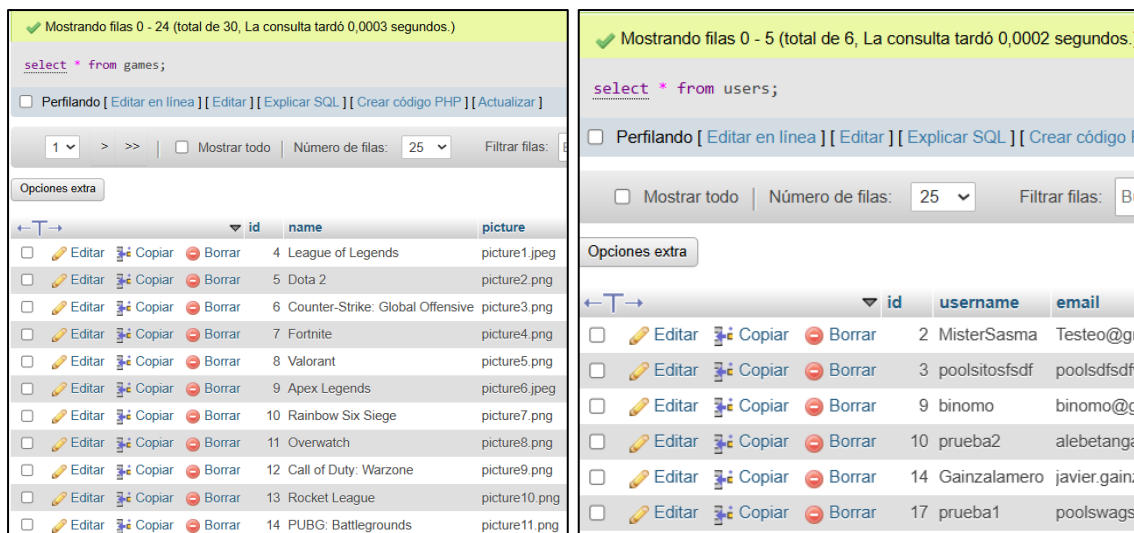


Step 1: Create the Database

To get started, the DB.sql file contains the steps to create the gamming_01 database. This file is an exported SQL version of my own project.

By importing this file, e.g., in phpMyAdmin (XAMPP), the database will be created automatically, including the necessary tables and the initial data entries for the games table.

Additionally, within the folder there is a subfolder that contains the game images. These images are named exactly as the values in the pictures attribute of the games table, making it easier to manage and display them on the main page.



Step 2: Modify the File to Include the Database

To link the database to the API file, you need to insert the database connection details into the script. Make sure to update the following variables in the PHP API file:

\$host	Localhost / 127.0.0.1
\$dbname	gamming_01
\$user	pooladmin
\$pass	123456

```
// ===== [Insert database connection here] =====
// Replace these credentials with your actual database connection
$host = "localhost"; //Host-name, example: 127.0.0.1
$dbname = ""; //Name of Database
$user = "your_user"; //User of the Database, (Grant option)
$pass = "your_password"; //Strong Passwd
```

If you want to use a different user, copy and paste the following code and replace the values with your own credentials:

```
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Step 3: (Optional) Security

If you want to enable the **secure password** option, simply **uncomment** the following block of code in the API script by removing the `/*` and `*/`:

```
// Optional: enforce basic password strength
// =====[Strong Password]=====
/*
if (strlen($password) < 8 || !preg_match('/[A-Z]/', $password) || !preg_matc
    echo json_encode([
        "error" => "Password must be at least 8 characters, include a number
    ]);
    exit;
}
*/
// Hash password before storing it
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);
$defaultGameId = 4; // Default game preference
```

Step 4: Link with Your Own Project

These are the PHP parameters used by the API. If you want to integrate the API into your own project, you can use these parameters as needed. Some, such as `filePath` or `filename`, are customizable in case you prefer to use a different folder or file naming convention.

PARAMETRE	DESCRIPTION
<code>\$username</code>	Name of the user account
<code>\$email</code>	Mail of the account
<code>\$birthdate</code>	Birthdate of the account
<code>\$hashedPassword</code>	The password of the account hashed
<code>\$defaultGameId</code>	ID default (in my case 4)
<code>\$folderPath</code>	Path to the logs folder
<code>\$filename</code>	Name of the log: "new_ac_" . \$safeUsername . "_" . \$timestamp . ".json";
<code>\$timestamp</code>	Time (Now)