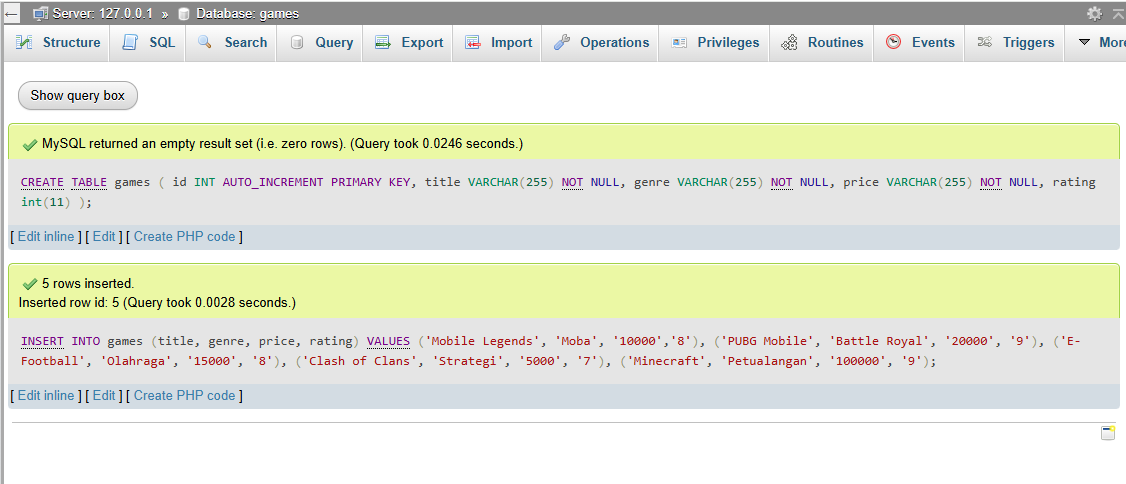
Nama : Irwan Elsandhy

Nim : 21.01.55.0007

1. Membuat Database

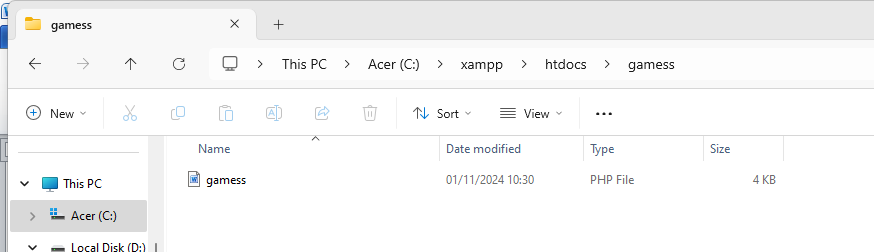
[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) games ( id INT AUTO\_INCREMENT PRIMARY KEY, title VARCHAR(255) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL, genre VARCHAR(255) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL, price VARCHAR(255) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL, rating int(11) );

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO games (title, genre, price, rating) [VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values) ('Mobile Legends', 'Moba', '10000','8'), ('PUBG Mobile', 'Battle Royal', '20000', '9'), ('E-Football', 'Olahraga', '15000', '8'), ('Clash of Clans', 'Strategi', '5000', '7'), ('Minecraft', 'Petualangan', '100000', '9');



1. Membuat File di File C

Membuat folder di dalam C-Xampp-Htdock-gamess.php



1. Program yang ada di gamess.php

<?php

header("Content-Type: application/json; charset=UTF-8");

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE");

header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, genreization, X-Requested-With");

$method = $\_SERVER['REQUEST\_METHOD'];

$request = [];

if (isset($\_SERVER['PATH\_INFO'])) {

$request = explode('/', trim($\_SERVER['PATH\_INFO'],'/'));

}

function getConnection() {

$host = 'localhost';

$db = 'gamess';

$user = 'root';

$pass = ''; // Ganti dengan password MySQL Anda jika ada

$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";

$options = [

PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC,

PDO::ATTR\_EMULATE\_PREPARES => false,

];

try {

return new PDO($dsn, $user, $pass, $options);

} catch (\PDOException $e) {

throw new \PDOException($e->getMessage(), (int)$e->getCode());

}

}

function response($status, $data = NULL) {

header("HTTP/1.1 " . $status);

if ($data) {

echo json\_encode($data);

}

exit();

}

$db = getConnection();

switch ($method) {

case 'GET':

if (!empty($request) && isset($request[0])) {

$id = $request[0];

$stmt = $db->prepare("SELECT \* FROM games WHERE id = ?");

$stmt->execute([$id]);

$games = $stmt->fetch();

if ($games) {

response(200, $games);

} else {

response(404, ["message" => "gamess not found"]);

}

} else {

$stmt = $db->query("SELECT \* FROM games");

$games = $stmt->fetchAll();

response(200, $games);

}

break;

case 'POST':

$data = json\_decode(file\_get\_contents("php://input"));

if (!isset($data->title) || !isset($data->genre) || !isset($data->price) || !isset($data->rating)) {

response(400, ["message" => "Missing required fields"]);

}

$sql = "INSERT INTO games (title, genre, price, rating) VALUES (?, ?, ?, ?)";

$stmt = $db->prepare($sql);

if ($stmt->execute([$data->title, $data->genre, $data->price, $data->rating])) {

response(201, ["message" => "Games created", "id" => $db->lastInsertId()]);

} else {

response(500, ["message" => "Failed to create games"]);

}

break;

case 'PUT':

if (empty($request) || !isset($request[0])) {

response(400, ["message" => "games ID is required"]);

}

$id = $request[0];

$data = json\_decode(file\_get\_contents("php://input"));

if (!isset($data->title) || !isset($data->genre) || !isset($data->price) || !isset($data->rating)) {

response(400, ["message" => "Missing required fields"]);

}

$sql = "UPDATE games SET title = ?, genre = ?, price = ?, rating = ? WHERE id = ?";

$stmt = $db->prepare($sql);

if ($stmt->execute([$data->title, $data->genre, $data->price, $data->rating, $id])) {

response(200, ["message" => "Games updated"]);

} else {

response(500, ["message" => "Failed to update games"]);

}

break;

case 'DELETE':

if (empty($request) || !isset($request[0])) {

response(400, ["message" => "gamess ID is required"]);

}

$id = $request[0];

$sql = "DELETE FROM games WHERE id = ?";

$stmt = $db->prepare($sql);

if ($stmt->execute([$id])) {

response(200, ["message" => "games deleted"]);

} else {

response(500, ["message" => "Failed to delete games"]);

}

break;

default:

response(405, ["message" => "Method not allowed"]);

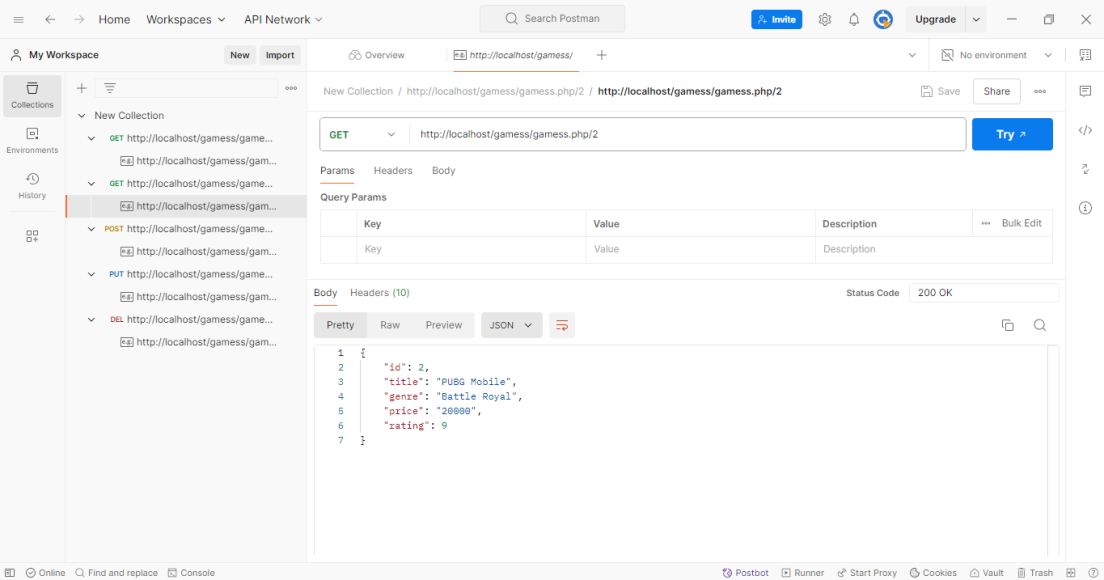
break;

}

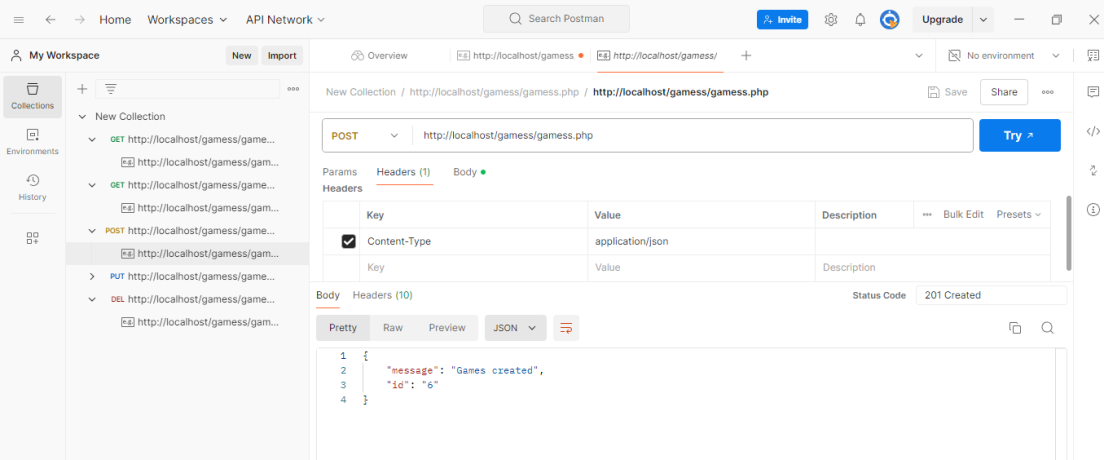
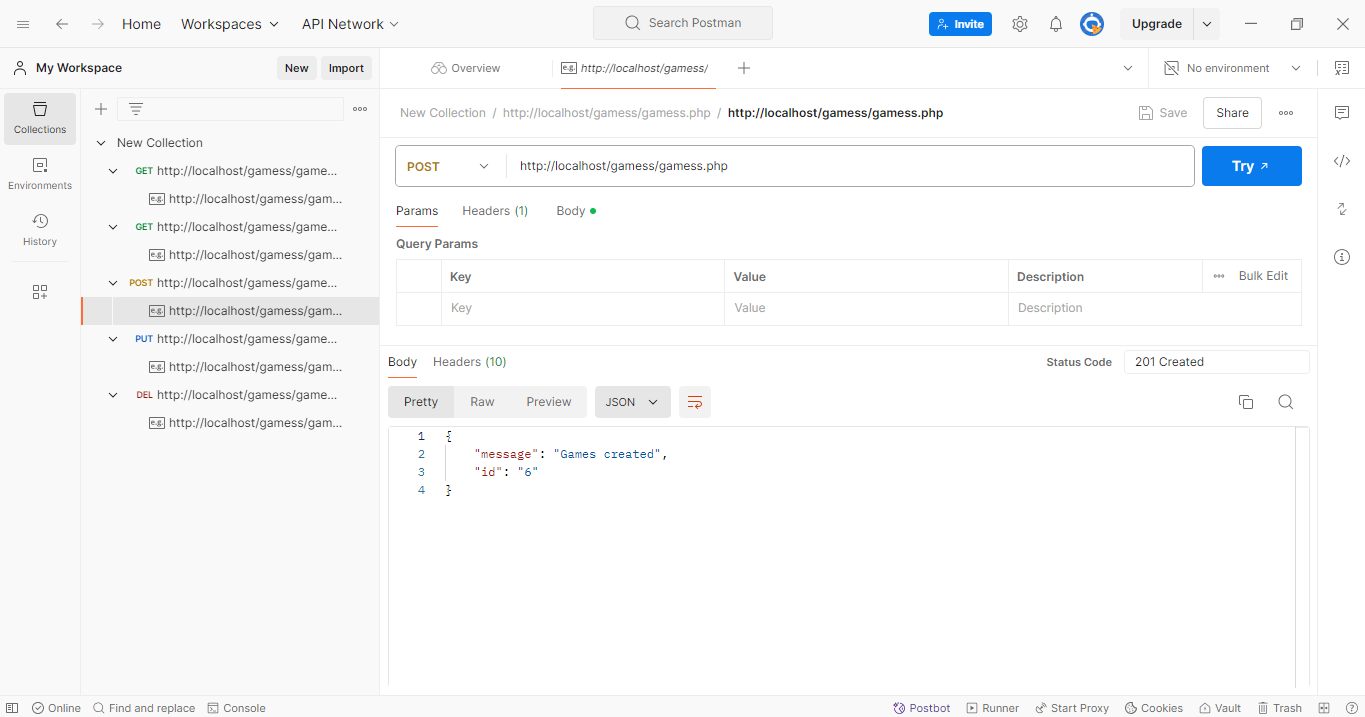
?>

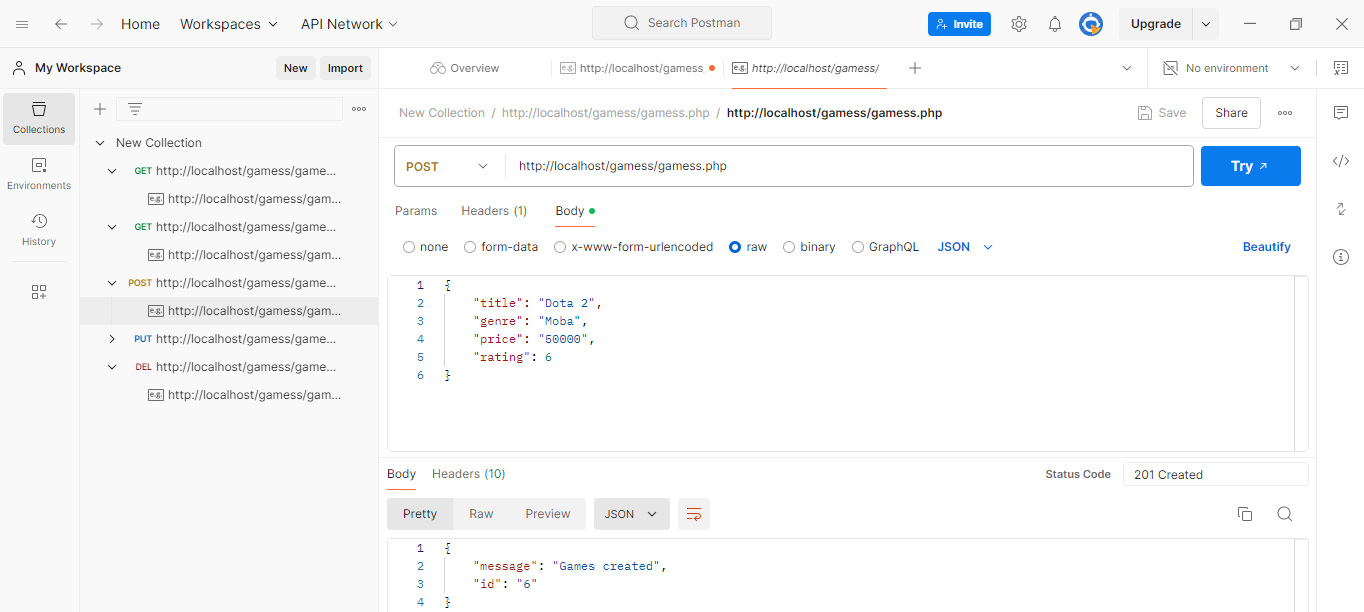
1. Hasil Untuk Menampilkan Seluruh Data

GET -> <http://localhost/gamess/gamess.php/2>

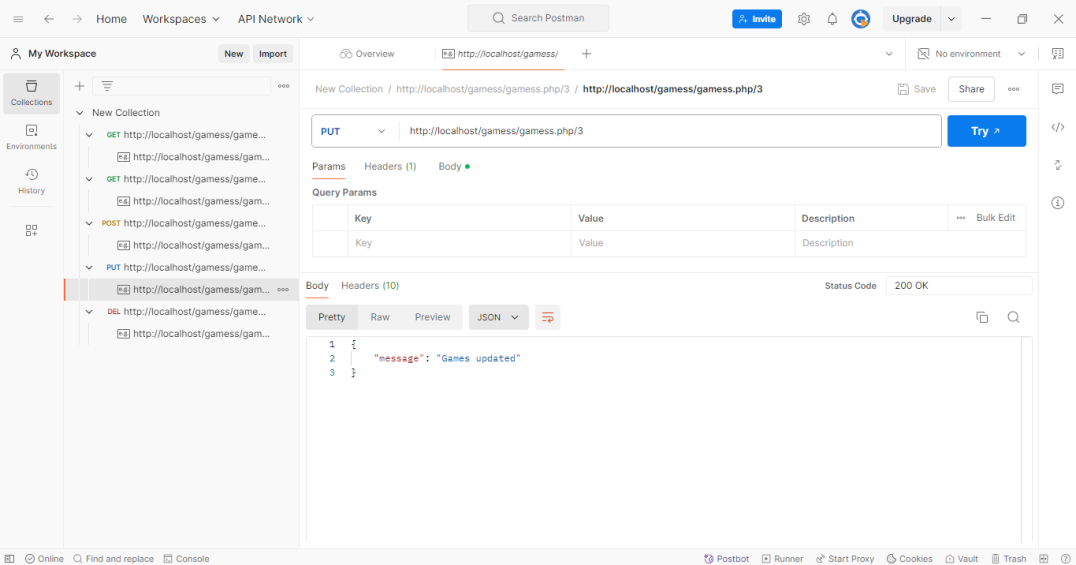


1. Post





1. PUT



1. DELETE

