

Regisztráció API elkészítése és tesztelése

1.PHP forrásfájl

```
<?php
// Betöltöm az adatbázis-kapcsolatot létrehozó fájlt.
require_once 'kapcsolat.php';

// Beállítom, hogy JSON-t adjak vissza.
header('Content-Type: application/json');

// Csak a POST kéréseket fogadom el.
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405);
    echo json_encode(['error' => 'Csak POST metódussal érhető el ez a végpont.']);
    exit;
}

// A bejövő nyers JSON-t dekódolom tömbbé.
$input = json_decode(file_get_contents('php://input'), true);

// Ellenőrzöm, hogy minden szükséges mező létezik-e, és a szerepkor érvényes-e.
if (
    !isset($input['nev'], $input['email'], $input['jelszo'], $input['szerepkor'])
    ||
    !in_array($input['szerepkor'], ['felhasznalo', 'admin'], true)
) {
    http_response_code(400);
    echo json_encode(['error' => 'Nem sikerült értelmezni a JSON bemenetet, vagy hiányzó/érvénytelen mező.']);
    exit;
}

// A bemenetből kiolvasom és megtisztítom a stringeket.
$nev = trim($input['nev']);
$email = trim($input['email']);
$jelszo = trim($input['jelszo']);
$szerepkor = $input['szerepkor'];

// Ellenőrzöm, hogy a név, email és jelszó nem üres.
if ($nev === '' || $email === '' || $jelszo === '') {
    http_response_code(400);
    echo json_encode(['error' => 'A név, email és jelszó mezők nem lehetnek üresek.']);
    exit;
}
```

Szakmai Gyakorlat I. – Backend fejlesztési szakasz

Orosz Kristóf – EYZWG9

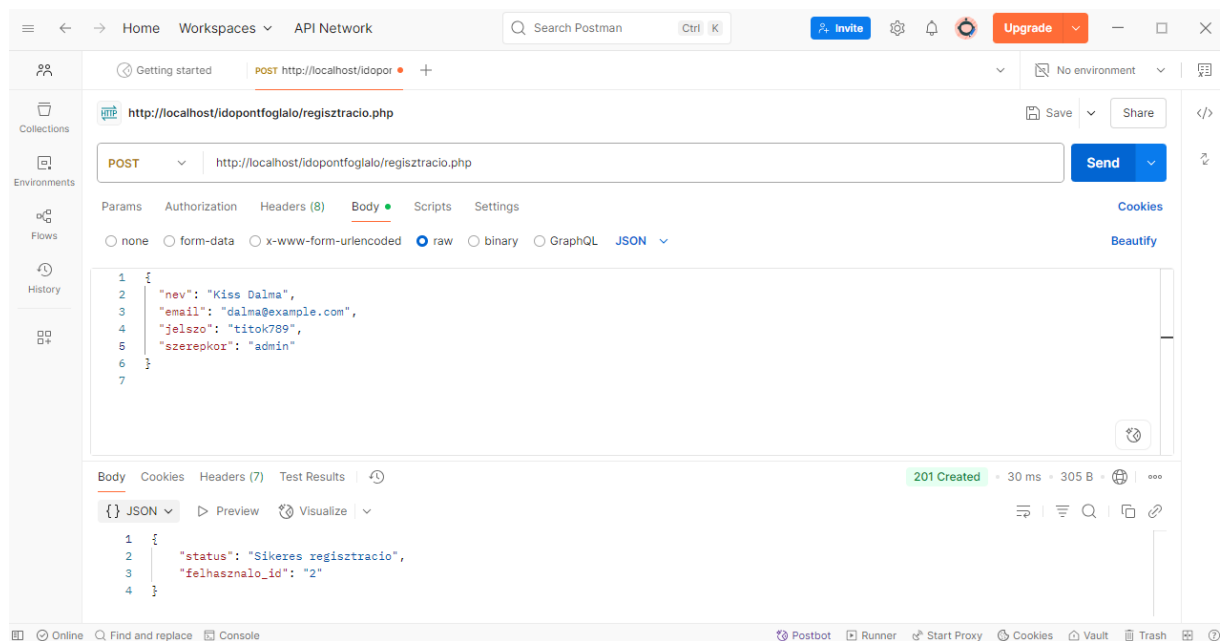
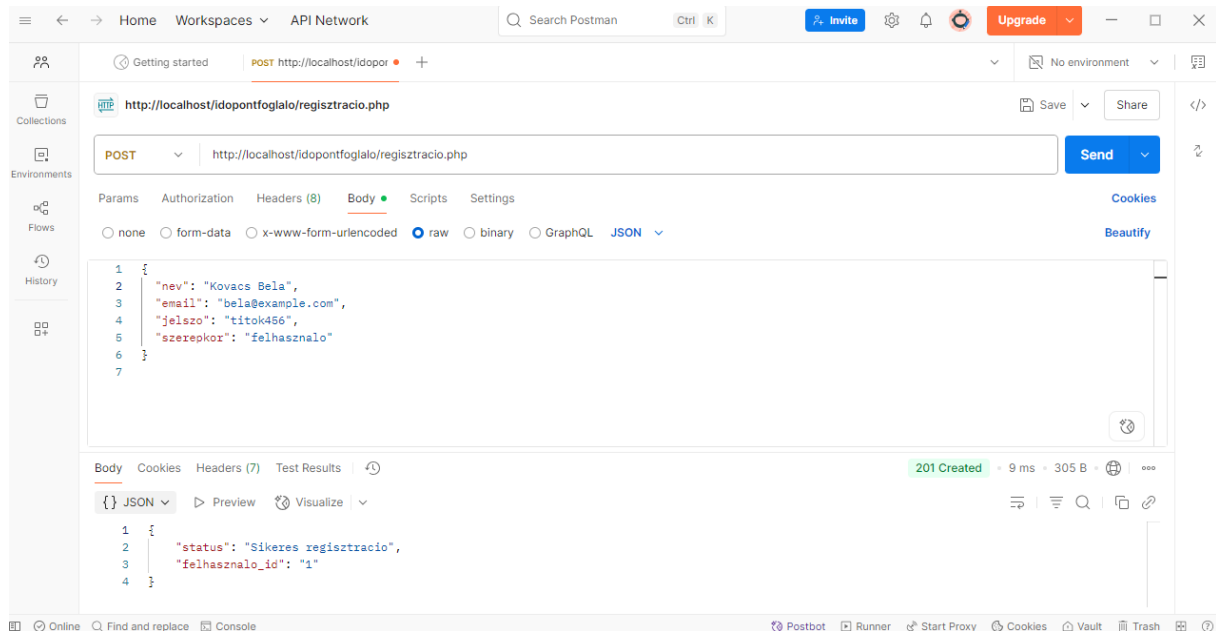
```
}

try {
    // Lekérdezem, hogy létezik-e már felhasználó ezzel az email címmel.
    $stmt = $pdo->prepare('SELECT COUNT(*) FROM felhasznalok WHERE email = :email');
    $stmt->bindParam(':email', $email, PDO::PARAM_STR);
    $stmt->execute();
    if ($stmt->fetchColumn() > 0) {
        http_response_code(409);
        echo json_encode(['error' => 'Ez az email már regisztrálva van.']);
        exit;
    }
} catch (PDOException $e) {
    // Ha adatbázis-hiba történik.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba: ' . $e->getMessage()]);
    exit;
}

try {
    // Beszúrom az új felhasználót az adatbázisba.
    $stmt = $pdo->prepare('INSERT INTO felhasznalok (nev, email, jelszo, szerep) VALUES (:nev, :email, :jelszo, :szerep)');
    $stmt->bindParam(':nev', $nev, PDO::PARAM_STR);
    $stmt->bindParam(':email', $email, PDO::PARAM_STR);
    $stmt->bindParam(':jelszo', $jelszo, PDO::PARAM_STR);
    $stmt->bindParam(':szerep', $szerep, PDO::PARAM_STR);
    $stmt->execute();

    // Lekérem az újonnan beszúrt felhasználó azonosítóját.
    $newId = $pdo->lastInsertId();
    http_response_code(201); // 201 Created
    echo json_encode([
        'status' => 'Sikeres regisztracio',
        'felhasznalo_id' => $newId
    ]);
    exit;
} catch (PDOException $e) {
    // Ha ismét adatbázis-hiba történik.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba: ' . $e->getMessage()]);
    exit;
}
```

2. API tesztelése Postman segítségével



3. Ellenőrzés MariaDB-ben.

```
MariaDB [idopontfoglalo]> SELECT * FROM Felhasznalok;
```

id	nev	szerepkor	email	jelszo	reg_datum
1	Kovacs Bela	felhasznalo	bela@example.com	titok456	2025-06-05 14:54:59
2	Kiss Dalma	admin	dalma@example.com	titok789	2025-06-05 14:56:09

```
2 rows in set (0.000 sec)
```