

Időpontok kezelése API

Adatbázis feltöltése tesztadatokkal.

```
MariaDB [idopontfoglalo]> INSERT INTO idopontok (szolgaltatok_id, datum, ido, foglalhato)
-> SELECT
->   s.id,
->   '2025-06-10' AS datum,
->   t.ido,
->   1 AS foglalhato
-> FROM szolgaltatok AS s
-> CROSS JOIN (
->   SELECT '09:00:00' AS ido UNION ALL
->   SELECT '10:00:00'          UNION ALL
->   SELECT '11:00:00'          UNION ALL
->   SELECT '12:00:00'          UNION ALL
->   SELECT '13:00:00'
-> ) AS t
-> ORDER BY s.id, t.ido;
Query OK, 125 rows affected (0.008 sec)
Records: 125  Duplicates: 0  Warnings: 0
```

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

id	szolgaltatok_id	datum	ido	foglalhato
1	1	2025-06-10	09:00:00	1
2	1	2025-06-10	10:00:00	1
3	1	2025-06-10	11:00:00	1
4	1	2025-06-10	12:00:00	1
5	1	2025-06-10	13:00:00	1
6	2	2025-06-10	09:00:00	1
7	2	2025-06-10	10:00:00	1
8	2	2025-06-10	11:00:00	1
9	2	2025-06-10	12:00:00	1
10	2	2025-06-10	13:00:00	1
11	3	2025-06-10	09:00:00	1
12	3	2025-06-10	10:00:00	1
13	3	2025-06-10	11:00:00	1
14	3	2025-06-10	12:00:00	1
15	3	2025-06-10	13:00:00	1
16	4	2025-06-10	09:00:00	1
17	4	2025-06-10	10:00:00	1
18	4	2025-06-10	11:00:00	1
19	4	2025-06-10	12:00:00	1
20	4	2025-06-10	13:00:00	1
21	5	2025-06-10	09:00:00	1
22	5	2025-06-10	10:00:00	1
23	5	2025-06-10	11:00:00	1
24	5	2025-06-10	12:00:00	1
25	5	2025-06-10	13:00:00	1
26	6	2025-06-10	09:00:00	1
27	6	2025-06-10	10:00:00	1

Időpontok listázása

Forrásfájl

```
<?php
// Betöltöm az adatbázis kapcsolatot.
require_once "kapcsolat.php";

// Beállítom, hogy a válasz JSON formátumban történjen.
header('Content-Type: application/json');

// Csak GET metódust engedélyezek, más esetben hibát küldök vissza.
if ($_SERVER['REQUEST_METHOD'] !== 'GET') {
    http_response_code(405);
    echo json_encode(['error' => 'Csak GET metódussal érhető el ez a végpont.']);
    exit;
}

try {
    // Lekérdezem az összes időpontot a szolgáltató nevekkkel együtt, időrendbe
    rendezve.
```

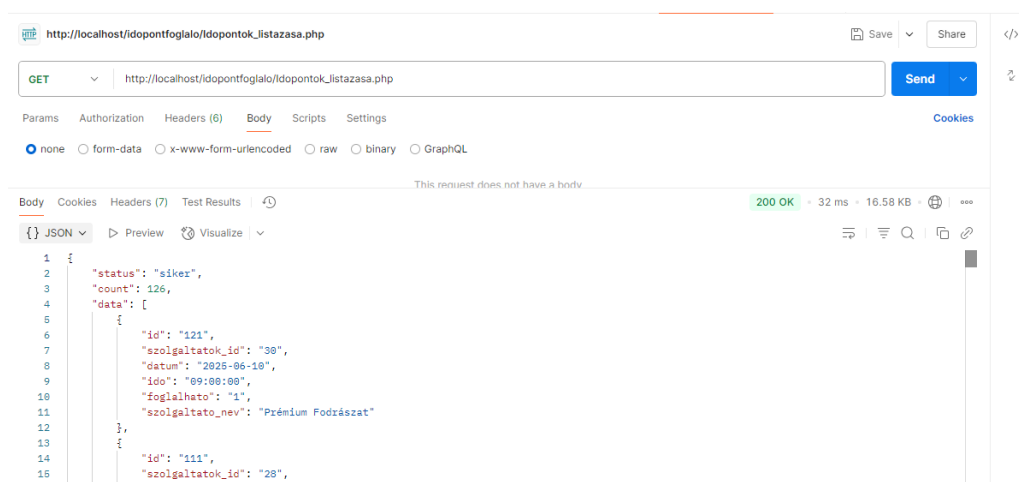
```
$stmt = $pdo->query("
    SELECT
        i.id,
        i.szolgaltatok_id,
        i.datum,
        i.ido,
        i.foglalhato,
        sz.nev AS szolgaltato_nev
    FROM idopontok i
    LEFT JOIN szolgaltatok sz ON i.szolgaltatok_id = sz.id
    ORDER BY i.datum, i.ido
");

// Lekérem az eredményeket tömb formátumban.
$idopontok = $stmt->fetchAll();

// Visszaküldöm a választ JSON formátumban, benne az összes időponttal és a
darabszámmal.
echo json_encode([
    'status' => 'siker',
    'count'  => count($idopontok),
    'data'   => $idopontok
]);

} catch (PDOException $e) {
    // Hiba esetén 500-as hibakódot és részletes hibaüzenetet küldök vissza.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba: ' . $e->getMessage()]);
}
```

Postman tesztelés



Időpontok létrehozása

Forrásfájl

```
<?php
// Csatlakozom az adatbázishoz.
require_once "kapcsolat.php";

// Beállítom, hogy a válasz JSON formátumú legyen.
header('Content-Type: application/json');

// Csak POST metódus engedélyezett, más esetben 405 hibát küldök vissza.
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405);
    echo json_encode(['error' => 'Csak POST metódussal érhető el ez a végpont.']);
    exit;
}

// Beolvasom a JSON formátumú bemenetet és tömbbé alakítom.
$input = json_decode(file_get_contents('php://input'), true);

// Ellenőrzöm, hogy minden szükséges adat megérkezett-e.
if (
    !is_array($input) ||
    !isset($input['szolgaltatok_id'], $input['datum'], $input['ido'])
) {
    http_response_code(400);
    echo json_encode(['error' => 'Hiányzó mezők (szolgaltatok_id, datum, ido).']);
    exit;
}

// Az értékeket előkészítem a beszúráshoz.
$szId = (int)$input['szolgaltatok_id'];
$datum = trim($input['datum']);
$ido = trim($input['ido']);
$fogl = isset($input['foglalhato']) ? (int)$input['foglalhato'] : 1; // ha nincs megadva, 1-et állítok be

try {
    // Elkészítem a SQL utasítást az új időpont beszúráshoz.
    $stmt = $pdo->prepare("
        INSERT INTO idopontok (szolgaltatok_id, datum, ido, foglalhato)
        VALUES (:szid, :datum, :ido, :foglalhato)
    ");

    // Lefuttatom a lekérdezést a megadott adatokkal.
    $stmt->execute([
        ':szid' => $szId,
        ':datum' => $datum,
        ':ido' => $ido,
```

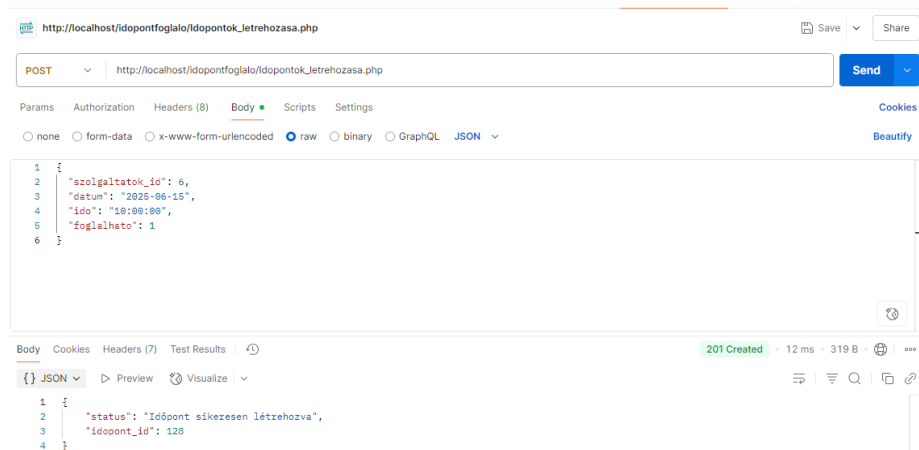
Szakmai Gyakorlat I. – Backend fejlesztési szakasz

Orosz Kristóf – EYZWG9

```
        ':foglalhato' => $fogl
    });

    // Sikeres beszárás esetén visszaküldöm az új rekord azonosítóját.
    http_response_code(201);
    echo json_encode([
        'status' => 'Időpont sikeresen létrehozva',
        'idopont_id' => (int)$pdo->lastInsertId()
    ]);
} catch (PDOException $e) {
    // Ha hiba történik, visszajelzést adok róla JSON formátumban.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba: ' . $e->getMessage()]);
}
```

Postman tesztelés



Időpontok módosítása

Forrásfájl

```
<?php
// Betöltöm az adatbázis kapcsolatot.
require_once "kapcsolat.php";
header('Content-Type: application/json');

// Csak PUT metódust engedek, egyéb esetben hibát dobok.
if ($_SERVER['REQUEST_METHOD'] !== 'PUT') {
    http_response_code(405);
    echo json_encode(['error' => 'Csak PUT metódussal érhető el ez a végpont.']);
    exit;
}

// Ellenőrzöm, hogy kapok-e érvényes id-t GET paraméterként.
if (empty($_GET['id']) || !ctype_digit($_GET['id'])) {
```

Szakmai Gyakorlat I. – Backend fejlesztési szakasz

Orosz Kristóf – EYZWG9

```
        http_response_code(400);
        echo json_encode(['error' => 'Hiányzó vagy érvénytelen id.']);
        exit;
    }
    $id = (int)$_GET['id']; // Az ID számot egész számként tárolom.

    // Beolvasom a JSON-bemenetet, és tömbbé alakítom.
    $input = json_decode(file_get_contents('php://input'), true);
    if (!is_array($input)) {
        http_response_code(400);
        echo json_encode(['error' => 'Nem sikerült értelmezni a JSON bemenetet.']);
        exit;
    }

    // Ellenőrzöm, hogy az adott ID-val létezik-e időpont az adatbázisban.
    try {
        $stmtChk = $pdo->prepare('SELECT COUNT(*) FROM idopontok WHERE id = :id');
        $stmtChk->execute([':id' => $id]);
        if ($stmtChk->fetchColumn() == 0) {
            http_response_code(404);
            echo json_encode(['error' => 'Nincs ilyen id-jú időpont.']);
            exit;
        }
    } catch (PDOException $e) {
        http_response_code(500);
        echo json_encode(['error' => 'Adatbázis hiba (ellenőrzés): ' . $e->getMessage()]);
        exit;
    }

    // Felépítem a frissítendő mezők listáját és az értékeket.
    $fields = [];
    $params = [':id' => $id];

    if (isset($input['szolgaltatok_id'])) {
        $fields[] = 'szolgaltatok_id = :szid';
        $params[':szid'] = (int)$input['szolgaltatok_id'];
    }
    if (isset($input['datum'])) {
        $fields[] = 'datum = :datum';
        $params[':datum'] = trim($input['datum']);
    }
    if (isset($input['ido'])) {
        $fields[] = 'ido = :ido';
        $params[':ido'] = trim($input['ido']);
    }
    if (isset($input['foglalhato'])) {
```

Szakmai Gyakorlat I. – Backend fejlesztési szakasz

Orosz Kristóf – EYZWG9

```
$fields[] = 'foglalhato = :fogl';
$params[':fogl'] = (int)$input['foglalhato'];
}

// Ha egyetlen mező sincs megadva, nem végzek módosítást.
if (empty($fields)) {
    http_response_code(400);
    echo json_encode(['error' => 'Nincs módosítandó mező.']);
    exit;
}

// Összeállítom a dinamikus SQL lekérdezést.
$sql = 'UPDATE idopontok SET ' . implode(', ', $fields) . ' WHERE id = :id';

try {
    // Lefuttatom a frissítést az előkészített adatokkal.
    $stmt = $pdo->prepare($sql);
    $stmt->execute($params);
    echo json_encode(['status' => 'Időpont sikeresen módosítva']);
} catch (PDOException $e) {
    // Hiba esetén részletes hibaüzenetet küldök vissza.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (update idopont): ' . $e-
    >getMessage()]);
    exit;
}
```

Postman tesztelés

The screenshot shows a Postman interface for testing a REST API. The URL is `http://localhost/idopontfoglalo/ldopontok_modositasa.php?id=12`. The method is `PUT`. The request body is a JSON object:

```
{
  "datum": "2025-06-20",
  "ido": "14:00:00",
  "foglalhato": 0
}
```

The response is a `200 OK` status with a response time of `12 ms` and a size of `301 B`. The response body is a JSON object:

```
{
  "status": "Időpont sikeresen módosítva"
}
```

MariaDB ellenőrzés

12	3	2025-06-20	14:00:00	0
----	---	------------	----------	---

Időpontok törlése

Forrásfájl

```
<?php
// Csatlakozom az adatbázishoz a külön fájlban tárolt kapcsolat segítségével.
require_once "kapcsolat.php";

// Beállítom a válasz formátumát JSON-ra.
header('Content-Type: application/json');

// Ellenőrzöm, hogy a kérés módszere valóban DELETE-e.
if ($_SERVER['REQUEST_METHOD'] !== 'DELETE') {
    http_response_code(405); // Nem engedélyezett módszer
    echo json_encode(['error' => 'Csak DELETE módszerrel érhető el ez a
végpont.']);
    exit;
}

// Ellenőrzöm, hogy kaptam-e érvényes numerikus id paramétert.
if (empty($_GET['id']) || !ctype_digit($_GET['id'])) {
    http_response_code(400); // Rossz kérés
    echo json_encode(['error' => 'Hiányzó vagy érvénytelen id.']);
    exit;
}

$id = (int)$_GET['id']; // Az ID-t egész számmá alakítom.

try {
    // Lekérdezem, hogy létezik-e az adott ID-jú időpont.
    $stmtChk = $pdo->prepare('SELECT COUNT(*) FROM idopontok WHERE id = :id');
    $stmtChk->execute([':id' => $id]);
    if ($stmtChk->fetchColumn() == 0) {
        http_response_code(404); // Nem található.
        echo json_encode(['error' => 'Nincs ilyen id-jú időpont.']);
        exit;
    }
} catch (PDOException $e) {
    // Hiba történt az ellenőrzés közben.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (ellenőrzés): ' . $e-
>getMessage()]);
    exit;
}

try {
```

Szakmai Gyakorlat I. – Backend fejlesztési szakasz

Orosz Kristóf – EYZWG9

```
// Törölöm az időpontot az adatbázisból.
$stmt = $pdo->prepare('DELETE FROM idopontok WHERE id = :id');
$stmt->execute([':id' => $id]);

// Visszajelzést adok, ha sikeres volt a törlés.
echo json_encode(['status' => 'Időpont sikeresen törölve']);
} catch (PDOException $e) {
    // Hiba történt a törlés során.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (delete idopont): ' . $e-
>getMessage()]);
    exit;
}
```

Postman tesztelés

