

## Szolgáltatók funkció működés

### PHP forrásfájlok

#### Szolgáltatók létrehozása

```
<?php
// Betöltöm a PDO-kapcsolatot létrehozó fájlt, hogy legyen $pdo objektumom.
require_once "kapcsolat.php";

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

// Ellenőrzöm, hogy a kliens csak POST metódussal érhesse el ezt a végpontot.
// Ha más metódus érkezik, 405-ös hibával kilépek.
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405);
    echo json_encode(['error' => 'Csak POST metódussal érhető el ez a végpont.']);
    exit;
}

// Beolvasom a bejövő JSON-t, és asszociatív tömbbé alakítom.
$input = json_decode(file_get_contents('php://input'), true);

// Ellenőrzöm, hogy a JSON egy tömb legyen, és tartalmazza a 'nev', 'leiras',
// 'szolgaltatas_tipusok_id' kulcsokat.
// Az 'aktiv' mezőnek pedig vagy 0-nak, vagy 1-nek kell lennie.
if (
    !is_array($input) ||
    !isset($input['nev'], $input['leiras'], $input['szolgaltatas_tipusok_id']) ||
    (!isset($input['aktiv']) && $input['aktiv'] !== 0 && $input['aktiv'] !== 1)
) {
    http_response_code(400);
    echo json_encode(['error' => 'Hiányzó vagy érvénytelen mezők.']);
    exit;
}

// Kitisztítom és változóba mentem a bemenetet.
$nev = trim($input['nev']);
$leiras = trim($input['leiras']);
$szolgaltatas_tipusok_id = (int)$input['szolgaltatas_tipusok_id'];
$aktiv = (int)$input['aktiv'];

// Ellenőrzöm, hogy a név, leírás nem üres, és a típus-id pozitív szám-e.
if ($nev === '' || $leiras === '' || $szolgaltatas_tipusok_id <= 0) {
    http_response_code(400);
    echo json_encode(['error' => 'A név, leírás és szolgáltatástípus azonosító nem lehet üres.']);
}
```

```

        exit;
    }

    // Megnézem, hogy valóban létezik-e a megadott szolgáltatás-típus a táblában.
    // Ha nem, visszadobom a 400-as hibát.
    try {
        $stmtFK = $pdo->prepare('SELECT COUNT(*) FROM szolgáltatás_tipusok WHERE id = :tid');
        $stmtFK->bindParam(':tid', $szolgáltatás_tipusok_id, PDO::PARAM_INT);
        $stmtFK->execute();
        if ($stmtFK->fetchColumn() == 0) {
            http_response_code(400);
            echo json_encode(['error' => 'A megadott szolgáltatás-típus nem létezik.']);
            exit;
        }
    } catch (PDOException $e) {
        // Ha a lekérdezés közben hiba történik, 500-as hibát küldök a részletes üzenettel.
        http_response_code(500);
        echo json_encode(['error' => 'Adatbázis hiba (típus ellenőrzés): ' . $e->getMessage()]);
        exit;
    }

    // Most beszúrom az új szolgáltatót az adatbázisba.
    try {
        $stmt = $pdo->prepare('
            INSERT INTO szolgáltatók (szolgáltatás_tipusok_id, nev, leiras, aktiv)
            VALUES (:tid, :nev, :leiras, :aktiv)
        ');
        $stmt->bindParam(':tid', $szolgáltatás_tipusok_id, PDO::PARAM_INT);
        $stmt->bindParam(':nev', $nev, PDO::PARAM_STR);
        $stmt->bindParam(':leiras', $leiras, PDO::PARAM_STR);
        $stmt->bindParam(':aktiv', $aktiv, PDO::PARAM_INT);
        $stmt->execute();

        // Sikeres beszúrás esetén kiolvasom az új rekord AUTO_INCREMENT értékét.
        $newId = $pdo->lastInsertId();
        http_response_code(201);
        echo json_encode([
            'status' => 'Szolgáltató sikeresen létrehozva',
            'szolgáltató_id' => (int)$newId
        ]);
        exit;
    } catch (PDOException $e) {
        // Ha az INSERT során adatbázis-hiba történik, 500-as hibakódot küldök vissza.
        http_response_code(500);
    }

```

```
        echo json_encode(['error' => 'Adatbázis hiba (szolgáltató létrehozása): ' .  
$e->getMessage()]);  
        exit;  
    }
```

## Szolgáltatók módosítása

```
<?php  
// Betöltöm a kapcsolat.php fájlt, hogy a $pdo változó elérhető legyen az  
adatbázis-műveletekhez.  
require_once "kapcsolat.php";  
  
// Beállítom, hogy minden válasz JSON formátumban érkezzen a kliens felé.  
header('Content-Type: application/json');  
  
// Ellenőrzöm, hogy valóban PUT módszerrel hívják-e meg ezt a végpontot.  
if ($_SERVER['REQUEST_METHOD'] !== 'PUT') {  
    // Ha nem PUT, 405-ös (Method Not Allowed) HTTP státuszkódot küldök  
    http_response_code(405);  
    echo json_encode(['error' => 'Csak PUT módszerrel érhető el ez a végpont.']);  
    exit;  
}  
  
// Ellenőrzöm, hogy szerepel-e az URL-ben az "id" paraméter, és az csak számokat  
tartalmaz-e.  
if (!isset($_GET['id']) || !ctype_digit($_GET['id'])) {  
    // Ha nincs érvényes id, 400-as (Bad Request) státuszt küldök  
    http_response_code(400);  
    echo json_encode(['error' => 'Hiányzó vagy érvénytelen szolgáltató-id.']);  
    exit;  
}  
// Átalakítom az id-t integerre.  
$szolgáltato_id = (int)$_GET['id'];  
  
// A beérkező JSON-t beolvasom és asszociatív tömbbé alakítom.  
$input = json_decode(file_get_contents('php://input'), true);  
if (!is_array($input)) {  
    // Ha nem tudtam tömbbé dekódolni a JSON-t, 400-as hibát adok vissza  
    http_response_code(400);  
    echo json_encode(['error' => 'Nem sikerült értelmezni a JSON bemenetet.']);  
    exit;  
}  
  
try {  
    // Lekérdezem, hogy létezik-e ilyen "id"-jű rekord a szolgáltatók táblában.  
    $stmtCheck = $pdo->prepare('SELECT COUNT(*) FROM szolgáltatok WHERE id =  
:id');  
    $stmtCheck->bindParam(':id', $szolgáltato_id, PDO::PARAM_INT);
```

```
$stmtCheck->execute();
if ($stmtCheck->fetchColumn() == 0) {
    // Ha nem létezik, 404-es (Not Found) hibát küldök.
    http_response_code(404);
    echo json_encode(['error' => 'A megadott szolgáltató nem található.']);
    exit;
}
} catch (PDOException $e) {
    // Ha adatbázis hiba történik a létezés-ellenőrzés során, 500-as hibát adok
    vissza.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (létezés ellenőrzés): ' . $e-
    >getMessage()]);
    exit;
}

// Összeállítom, mely mezőket (oszlopokat) szeretnék frissíteni, és az ezekhez
tartozó paramétereket.
$fieldsToUpdate = [];
$params = [ ':id' => $szolgáltato_id ];

// Ha a bejövő JSON tartalmaz "nev"-et, akkor hozzáadom a frissítendő mezők közé.
if (isset($input['nev'])) {
    $fieldsToUpdate[] = 'nev = :nev';
    $params[':nev'] = trim($input['nev']);
}
// Ha tartalmaz "leiras"-t, hozzáadom.
if (isset($input['leiras'])) {
    $fieldsToUpdate[] = 'leiras = :leiras';
    $params[':leiras'] = trim($input['leiras']);
}
// Ha tartalmaz "szolgaltatas_tipusok_id"-t, hozzáadom és integer típusra cast-
olom.
if (isset($input['szolgaltatas_tipusok_id'])) {
    $fieldsToUpdate[] = 'szolgaltatas_tipusok_id = :tid';
    $params[':tid'] = (int)$input['szolgaltatas_tipusok_id'];
}
// Ha tartalmaz "aktiv"-ot, hozzáadom és integerre cast-olom.
if (isset($input['aktiv'])) {
    $fieldsToUpdate[] = 'aktiv = :aktiv';
    $params[':aktiv'] = (int)$input['aktiv'];
}

// Ha nem volt egyetlen frissítendő mező sem, 400-as hibát küldök.
if (empty($fieldsToUpdate)) {
    http_response_code(400);
    echo json_encode(['error' => 'Nincs megadva egyetlen módosítandó mező sem.']);
    exit;
}
```

```
}

// Ha a szolgáltatás_típusok_id mezőt is adják, előbb ellenőrzöm, hogy az id
létezik-e a szolgáltatás_típusok táblában.
if (isset($params[':tid'])) {
    try {
        $stmtFK = $pdo->prepare('SELECT COUNT(*) FROM szolgáltatás_típusok WHERE
id = :tid');
        $stmtFK->bindParam(':tid', $params[':tid'], PDO::PARAM_INT);
        $stmtFK->execute();
        if ($stmtFK->fetchColumn() == 0) {
            // Ha nincs ilyen típus, 400-as hibát küldök.
            http_response_code(400);
            echo json_encode(['error' => 'A megadott (új) szolgáltatás-típus nem
létezik.']);
            exit;
        }
    } catch (PDOException $e) {
        // Ha adatbázis hiba adódik a típus ellenőrzésénél, 500-as hibát küldök.
        http_response_code(500);
        echo json_encode(['error' => 'Adatbázis hiba (típus ellenőrzés): ' . $e-
>getMessage()]);
        exit;
    }
}

// Összerakom az UPDATE SQL utasítást a kiválasztott mezőkkel.
$sql = 'UPDATE szolgáltatások SET ' . implode(', ', $fieldsToUpdate) . ' WHERE id =
:id';

try {
    // Előkészítem és végrehajtom a frissítést a paraméterekkel.
    $stmt = $pdo->prepare($sql);
    $stmt->execute($params);

    // Ha sikeres a módosítás, 200-as státuszt és egy JSON-feleletet küldök.
    http_response_code(200);
    echo json_encode(['status' => 'Szolgáltató sikeresen módosítva']);
    exit;
} catch (PDOException $e) {
    // Ha adatbázis hiba történik a frissítéskor, 500-as hibát küldök.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (szolgáltató módosítása): ' . $e-
>getMessage()]);
    exit;
}
```

## Szolgáltatók törlése

```
<?php
// Betöltöm a PDO-kapcsolatot kezelő fájlt, hogy legyen $pdo objektumom.
require_once "kapcsolat.php";

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

// Ellenőrzöm, hogy a kérés csak DELETE metódussal érhető el.
// Ha más metódust látok, 405-ös hibával kilépek.
if ($_SERVER['REQUEST_METHOD'] !== 'DELETE') {
    http_response_code(405);
    echo json_encode(['error' => 'Csak DELETE metódussal érhető el ez a
végpont.']);
    exit;
}

// Lekérem az id-t a query stringből, és ellenőrzöm, hogy számjegyekből áll-e.
// Ha hiányzik vagy nem digitális, 400-as hibát küldök.
if (empty($_GET['id']) || !ctype_digit($_GET['id'])) {
    http_response_code(400);
    echo json_encode(['error' => 'Hiányzó vagy érvénytelen id.']);
    exit;
}

$id = (int)$_GET['id']; // Ellenőrzöm, hogy egész szám az id.

// Először ellenőrzöm, létezik-e a megadott id-val szolgáltató.
// Ha nincs ilyen sor a táblában, 404-es hibát adok vissza.
try {
    $stmtChk = $pdo->prepare('SELECT COUNT(*) FROM szolgaltatok WHERE id = :id');
    $stmtChk->execute([':id' => $id]);
    if ($stmtChk->fetchColumn() == 0) {
        http_response_code(404);
        echo json_encode(['error' => 'A megadott id-val nem található
szolgáltatató.']);
        exit;
    }
} catch (PDOException $e) {
    // Ha az ellenőrzés közben adatbázis-hiba történik, 500-as hibával válaszolok.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (ellenőrzés): ' . $e-
>getMessage()]);
    exit;
}
```

```
}

// Ha tényleg létezik az id, végrehajtom a DELETE utasítást.
try {
    $stmt = $pdo->prepare('DELETE FROM szolgaltatok WHERE id = :id');
    $stmt->execute([':id' => $id]);

    // Sikeres törlés esetén 200-as státusszal visszaküldöm az eltávolított id-t.
    http_response_code(200);
    echo json_encode([
        'status' => 'Sikeres törlés',
        'deleted_id' => $id
    ]);
    exit;
} catch (PDOException $e) {
    // Ha a törlés közben hiba történik, 500-as hibával jelzem a részleteket.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (delete): ' . $e->getMessage()]);
    exit;
}
```

## HTML forrásfájlok

### Szolgáltatók létrehozása

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <!-- Dokumentum típus és nyelv -->
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <!-- Oldal címe -->
    <title>Szolgáltató létrehozása</title>
    <!-- Menü stíluslap -->
    <link rel="stylesheet" href="menu.css">
    <style>
        /* Alapértelmezett test és háttér */
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }
        /* Űrlap konténer */
        form {
            max-width: 400px;
            margin: 40px auto;
            padding: 20px;
```

```
    background: white;
    border-radius: 10px;
    box-shadow: 0 2px 8px rgba(0,0,0,0.1);
}
/* Űrlap mezők és gomb */
form input,
form select,
form button {
    width: 100%;
    margin: 10px 0;
    padding: 10px;
    font-size: 16px;
    border-radius: 5px;
    border: 1px solid #ccc;
}
/* Gomb színezése és kurzor */
form button {
    background-color: #007BFF;
    color: white;
    border: none;
    cursor: pointer;
}
form button:hover {
    background-color: #0056b3;
}
/* Segélykategória stílusok */
.color {
    color: red;
}
/* Rejtett elem osztály */
.hidden {
    display: none !important;
}
/* Mobilra reagáló stílus */
@media (max-width: 500px) {
    form {
        margin: 20px 10px;
        padding: 15px;
    }
    form input,
    form select,
    form button {
        font-size: 15px;
        padding: 9px;
    }
    h2 {
        font-size: 20px;
        padding: 0 10px;
    }
}
```



```
    }
  }
</style>
</head>
<body>
  <!-- Fejléc és navigáció -->
  <header>
    <a href="index.html" class="logo">Időpontfoglaló <span id="felhasznalo-
nev"></span></a>
    <nav>
      <a href="szolgaltatok.html">Szolgáltatók</a>
      <a href="idopontok.html">Időpontok</a>
      <a href="ertekelesek.html">Értékelések</a>
      <a href="bejelentkezes.html" id="nav-login"><b>Bejelentkezés</b></a>
      <a href="fiokom.html" id="nav-account" class="hidden">Fiókom</a>
      <a href="#" id="nav-logout" onclick="kijelentkezes()" class="hidden"><span
class="color">Kijelentkezés</span></a>
    </nav>
  </header>

  <!-- Címsor a form felett -->
  <h2 style="text-align: center;">Szolgáltató létrehozása</h2>
  <!-- Szolgáltató létrehozása űrlap -->
  <form id="letrehozForm">
    <input type="text" name="nev" placeholder="Szolgáltató neve" required />
    <input type="text" name="leiras" placeholder="Leírás" required />
    <input type="number" name="szolgaltatas_tipusok_id" placeholder="Típus ID"
required />
    <select name="aktiv" required>
      <option value="">-- Aktivitás kiválasztása --</option>
      <option value="1">Aktív</option>
      <option value="0">Inaktív</option>
    </select>
    <button type="submit">Létrehozás</button>
  </form>

  <!-- AJAX kód az űrlap elküldéséhez -->
  <script>
    const form = document.getElementById('letrehozForm');
    form.addEventListener('submit', async (e) => {
      e.preventDefault();
      const data = Object.fromEntries(new FormData(form));
      try {
        const response = await fetch('Szolgaltatok_letrehozasa.php', {
          method: 'POST',
          headers: { 'Content-Type': 'application/json' },
          body: JSON.stringify(data)
        });
      }
    });
```

```
        const result = await response.json();
        alert(result.status || result.error);
        if (response.ok) {
            window.location.href = 'szolgaltatok.html';
        }
    } catch (err) {
        alert('Hiba történt: ' + err.message);
    }
});
</script>

<!-- Kijelentkezés funkció -->
<script>
    function kijelentkezés() {
        sessionStorage.removeItem('felhasznalo');
        window.location.href = 'index.html';
    }
</script>

<!-- Felhasználó adatok beállítása a navigációhoz -->
<script>
    const user = JSON.parse(sessionStorage.getItem('felhasznalo'));
    if (user && user.nev) {
        document.getElementById('felhasznalo-nev').textContent = ` - ${user.nev}`;
        document.getElementById('nav-login').classList.add('hidden');
        document.getElementById('nav-account').classList.remove('hidden');
        document.getElementById('nav-logout').classList.remove('hidden');
    }
</script>
</body>
</html>
```

## Szolgáltatók módosítása

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <!-- Dokumentum típusa és nyelv beállítása -->
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <!-- Oldal címe -->
    <title>Szolgáltató módosítása</title>
    <!-- Menü stíluslap betöltése -->
    <link rel="stylesheet" href="menu.css">
    <style>
        /* Oldal alapstílusai */
        body {
            font-family: Arial, sans-serif;
```

```
background-color: #f4f4f4;
margin: 0;
padding: 0;
}
/* Űrlap stílusa */
form {
max-width: 400px;
margin: 40px auto;
padding: 20px;
background: white;
border-radius: 10px;
box-shadow: 0 2px 8px rgba(0,0,0,0.1);
}
/* Űrlap mezők és gomb */
form input,
form select,
form button {
width: 100%;
margin: 10px 0;
padding: 10px;
font-size: 16px;
border-radius: 5px;
border: 1px solid #ccc;
}
/* Gomb színezése */
form button {
background-color: #007BFF;
color: white;
border: none;
cursor: pointer;
}
form button:hover {
background-color: #0056b3;
}
/* Cím középre igazítása */
.center {
text-align: center;
}
/* Kijelentkezés piros színnel */
.color {
color: red;
}
/* Rejtett elemek elrejtése */
.hidden {
display: none !important;
}
/* Mobil nézet esetén */
@media (max-width: 500px) {
```

```

        form {
            margin: 20px 10px;
            padding: 15px;
        }
        form input,
        form select,
        form button {
            font-size: 15px;
            padding: 9px;
        }
        h2.center {
            font-size: 20px;
            padding: 0 10px;
        }
    }
</style>
</head>
<body>
    <!-- Fejléc és navigáció -->
    <header>
        <a href="index.html" class="logo">Időpontfoglaló <span id="felhasznalo-
nev"></span></a>
        <nav>
            <a href="szolgaltatok.html">Szolgáltatók</a>
            <a href="idopontok.html">Időpontok</a>
            <a href="ertekelesek.html">Értékelések</a>
            <a href="bejelentkezes.html" id="nav-login"><b>Bejelentkezés</b></a>
            <a href="fiokom.html" id="nav-account" class="hidden">Fiókom</a>
            <a href="#" id="nav-logout" onclick="kijelentkezes()" class="hidden"><span
class="color">Kijelentkezés</span></a>
        </nav>
    </header>

    <!-- Címsor a form felett -->
    <h2 class="center">Szolgáltató módosítása</h2>

    <!-- Szolgáltató adatainak módosító űrlapja -->
    <form id="modositForm">
        <!-- Szolgáltató neve -->
        <input type="text" name="nev" placeholder="Szolgáltató neve" required />
        <!-- Leírás -->
        <input type="text" name="leiras" placeholder="Leírás" required />
        <!-- Típus ID -->
        <input type="number" name="szolgaltatas_tipusok_id" placeholder="Típus ID"
required />
        <!-- Aktivitás kiválasztása -->
        <select name="aktiv">
            <option value="1">Aktív</option>

```

```
<option value="0">Inaktív</option>
</select>
<!-- Mentés gomb -->
<button type="submit">Módosítás mentése</button>
</form>

<script>
  // URL paraméterből ID kinyerése
  const urlParams = new URLSearchParams(window.location.search);
  const id = urlParams.get('id');
  if (!id) {
    alert("Hiányzó szolgáltató ID.");
    window.location.href = "szolgaltatok.html";
  }

  // Űrlap feltöltése meglévő adatokkal
  fetch("Szolgaltatok_listazasa.php")
    .then(res => res.json())
    .then(data => {
      const szolg = data.data.find(s => s.id == id);
      if (!szolg) {
        alert("Érvénytelen szolgáltató ID.");
        return;
      }
      const form = document.getElementById('modositForm');
      form.nev.value = szolg.nev;
      form.leiras.value = szolg.leiras;
      form.szolgaltatas_tipusok_id.value = szolg.szolgaltatas_tipusok_id || "";
      form.aktiv.value = szolg.aktiv;
    });

  // PUT kérés küldése módosításhoz
  const form = document.getElementById('modositForm');
  form.addEventListener('submit', async (e) => {
    e.preventDefault();
    const data = Object.fromEntries(new FormData(form));
    const response = await fetch(`Szolgaltatok_modositasa.php?id=${id}`, {
      method: 'PUT',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(data)
    });
    const result = await response.json();
    alert(result.status || result.error);
  });

  // Kijelentkezés funkció
  function kijelentkezés() {
    sessionStorage.removeItem('felhasznalo');
```

```
        window.location.href = "index.html";
    }

    // Navigáció frissítése bejelentkezett felhasználóval
    const user = JSON.parse(sessionStorage.getItem('felhasznalo'));
    if (user && user.nev) {
        document.getElementById('felhasznalo-nev').textContent = ` - ${user.nev}`;
        document.getElementById('nav-login').classList.add('hidden');
        document.getElementById('nav-account').classList.remove('hidden');
        document.getElementById('nav-logout').classList.remove('hidden');
    }
</script>
</body>
</html>
```

## Szolgáltatók törlése

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <!-- Dokumentum típusa és nyelv beállítása -->
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <!-- Oldal címe -->
    <title>Szolgáltató törlése</title>
    <!-- Menü stíluslap betöltése -->
    <link rel="stylesheet" href="menu.css" />
    <style>
        /* Oldal alapstílusai */
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }
        /* Űrlap doboz stílusai */
        form {
            max-width: 400px;
            margin: 40px auto;
            background: white;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 8px rgba(0,0,0,0.1);
        }
        /* Beviteli mezők és gombok */
        input,
        button {
            width: 95%;
```

```

        margin: 10px 0;
        padding: 10px;
        border-radius: 4px;
        font-size: 16px;
        border: 1px solid #ccc;
    }
    /* Gomb stílus */
    button {
        background-color: blue;
        color: white;
        border: none;
        cursor: pointer;
        text-align: center;
    }
    button:hover {
        background-color: blue;
    }
    /* Figyelmeztető szöveg piros színnel */
    .color {
        color: red;
        text-align: center;
    }
    /* Normál címszöveg */
    .color2 {
        color: black;
        text-align: center;
    }
    /* Rejtett elemek */
    .hidden {
        display: none !important;
    }
</style>
</head>
<body>
    <!-- Fejléc és navigáció -->
    <header>
        <a href="index.html" class="logo">Időpontfoglaló <span id="felhasznalo-
nev"></span></a>
        <nav>
            <a href="szolgaltatok.html">Szolgáltatók</a>
            <a href="idopontok.html">Időpontok</a>
            <a href="ertekelesek.html">Értékelések</a>
            <a href="bejelentkezes.html" id="nav-login"><b>Bejelentkezés</b></a>
            <a href="fiokom.html" id="nav-account" class="hidden">Fiókom</a>
            <a href="#" id="nav-logout" onclick="kijelentkezes()" class="hidden"><span
class="color">Kijelentkezés</span></a>
        </nav>
    </header>

```

```

<!-- Oldalcím -->
<h2 class="color2">Szolgáltató törlése</h2>

<!-- Törlő űrlap -->
<form id="torlesForm">
  <!-- Szolgáltató ID bevitele -->
  <input type="number" name="id" placeholder="Szolgáltató ID" required />
  <!-- Törlés gomb -->
  <button type="submit">Törlés</button>
</form>

<script>
  // Felhasználó adatainak ellenőrzése
  const user = JSON.parse(sessionStorage.getItem('felhasznalo'));
  if (user && user.nev) {
    document.getElementById('felhasznalo-nev').textContent = ` - ${user.nev}`;
  }

  // Csak adminok érhetik el az oldalt
  if (!user || user.szerepkor !== 'admin') {
    alert('Csak bejelentkezett admin használhatja ezt az oldalt.');
```

window.location.href = 'bejelentkezes.html';

```

  }

  // Törlés feldolgozása
  const form = document.getElementById('torlesForm');
  form.addEventListener('submit', async (e) => {
    e.preventDefault();
    const id = form.id.value;
    if (!confirm(`Biztosan törölni szeretné a(z) ${id}-es azonosítójú
szolgáltatót?`)) return;

    const response = await fetch(`Szolgaltatok_torlese.php?id=${id}`, { method:
'DELETE' });
    const result = await response.json();
    alert(result.status || result.error);
    if (response.ok) window.location.href = 'szolgaltatok.html';
  });

  // Kijelentkezés funkció
  function kijelentkezés() {
    sessionStorage.removeItem('felhasznalo');
    window.location.href = 'index.html';
  }

  // Navigációs elemek frissítése
  if (user && user.nev) {

```



## Szakmai Gyakorlat I. – Frontend – Backend integrációs szakasz

Orosz Kristóf – EYZWG9

```
document.getElementById('nav-login').classList.add('hidden');
document.getElementById('nav-account').classList.remove('hidden');
document.getElementById('nav-logout').classList.remove('hidden');
}
</script>
</body>
</html>
```

## HTML kimenetek

←

→

↺

localhost/idopontfoglalo/Szolgalatok\_letrehozasa.html

☆

📄

📄

📄

⋮

Időpontfoglaló - Kiss Dalma

SzolgáltatókIdőpontokÉrtékelésekFiókomKijelentkezés

Szolgáltató létrehozása

Szolgáltató neve

Leírás

Típus ID

-- Aktivitás kiválasztása --

Létrehozás

←

→

↺

localhost/idopontfoglalo/Szolgalatok\_modositasa.html?id=3

☆

📄

📄

📄

⋮

Időpontfoglaló - Kiss Dalma

SzolgáltatókIdőpontokÉrtékelésekFiókomKijelentkezés

Szolgáltató módosítása

EuroFogász Klinika

Modern implantológia és esztétikai fogászat

Típus ID

Aktív

Módosítás mentése

## Szakmai Gyakorlat I. – Frontend – Backend integrációs szakasz

Orosz Kristóf – EYZWG9

localhost/idopontfoglalo/Szolgaltatok\_torlese.html?id=3

Időpontfoglaló - Kiss Dalma

SzolgáltatókIdőpontokÉrtékelésekFiókomKijelentkezés

Szolgáltató törlése

Szolgáltató ID

Törlés