

December 24, 2025

SQLite(エスキューライト)は世界で最も広く使われているデータベースの一つ。アプリの中にすっぽり収まる、手軽で高性能な記録帳。本来データベースを使うには専用のサーバーを動かす必要があるが、SQLiteは1つのファイルの中に全てのデータが保存される。

1 親と帰省の持ち物を共有！

PHPのSQLite用ドライバをインストール

```
sudo apt update
sudo apt install php-sqlite3 -y
```

Apache再起動

```
sudo systemctl restart apache2
```

```
sudo nano /var/www/html/帰省.php
```

```
<?php
// データベース保存先 (/var/www/html/ に作成されます)
$db_file = __DIR__ . '/packing_list.db';
$db = new PDO("sqlite:$db_file");
$db->exec("CREATE TABLE IF NOT EXISTS items (id INTEGER PRIMARY KEY AUTOINCREMENT, content TEXT,
checked INTEGER, sort_order INTEGER)");

// API処理
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $input = json_decode(file_get_contents('php://input'), true);
    if ($input['action'] === 'save') {
        $db->exec("DELETE FROM items");
        $stmt = $db->prepare("INSERT INTO items (content, checked, sort_order) VALUES (?, ?, ?)");
        foreach ($input['items'] as $index => $item) {
            $stmt->execute([$item['text'], $item['checked'] ? 1 : 0, $index]);
        }
    }
    exit;
}

if (isset($_GET['action']) && $_GET['action'] === 'load') {
    $stmt = $db->query("SELECT content as text, checked FROM items ORDER BY sort_order ASC");
    echo json_encode($stmt->fetchAll(PDO::FETCH_ASSOC));
    exit;
}
?>
<!DOCTYPE html>
<html lang="ja">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>帰省持ち物リスト</title>
    <script src="https://cdn.jsdelivr.net/npm/sortablejs@1.15.0/Sortable.min.js"></script>
    <style>
        :root { --bg-color: #fdfaf7; --accent-color: #8c7851; --text-color: #4a4a4a; --card-bg: #
            ffffff; }
        body { background-color: var(--bg-color); color: var(--text-color); font-family: sans-serif;
            padding: 20px; }
        .container { max-width: 450px; margin: 0 auto; }
        header { text-align: center; margin-bottom: 20px; }
        .input-group { display: flex; gap: 10px; margin-bottom: 20px; }
        input[type="text"] { flex: 1; padding: 12px; border: 1px solid #ddd; border-radius: 8px;
            font-size: 16px; }
```

```

    button#add-btn { padding: 10px 20px; background: var(--accent-color); color: white; border:
        none; border-radius: 8px; cursor: pointer; }
    .item-list { list-style: none; padding: 0; }
    .item { background: var(--card-bg); padding: 15px; margin-bottom: 8px; border-radius: 10px;
        display: flex; align-items: center; box-shadow: 0 2px 5px rgba(0,0,0,0.05); cursor: grab
        ; }
    .item.checked span { text-decoration: line-through; color: #ccc; }
    .handle { margin-right: 15px; color: #ccc; font-size: 1.2rem; }
    .item span { flex: 1; font-size: 1rem; }
    .delete-btn { color: #eee; border: none; background: none; font-size: 1.5rem; cursor:
        pointer; padding-left: 10px; }
    .delete-btn:hover { color: #d9534f; }
</style>
</head>
<body>
<div class="container">
    <header>
        <h1 style="font-weight:300; letter-spacing: 2px;">帰省の準備</h1>
    </header>
    <div class="input-group">
        <input type="text" id="todo-input" placeholder="持ち物を入力..." onkeypress="if(event.key
            === 'Enter') addItem()">
        <button id="add-btn" onclick="addItem()">追加</button>
    </div>
    <ul id="list-container" class="item-list"></ul>
</div>
<script>
    const container = document.getElementById('list-container');
    const input = document.getElementById('todo-input');

    // 並び替えの有効化
    new Sortable(container, { animation: 150, handle: '.item', onEnd: saveToServer });

    async function saveToServer() {
        const items = Array.from(document.querySelectorAll('.item')).map(li => ({
            text: li.querySelector('span').innerText,
            checked: li.querySelector('input').checked
        }));
        await fetch('帰省.php', {
            method: 'POST',
            headers: {'Content-Type': 'application/json'},
            body: JSON.stringify({action: 'save', items: items})
        });
    }

    async function loadFromServer() {
        const res = await fetch('帰省.php?action=load');
        const items = await res.json();
        container.innerHTML = '';
        items.forEach(item => renderItem(item.text, item.checked == 1));
    }

    function renderItem(text, checked) {
        const li = document.createElement('li');
        li.className = 'item' + (checked ? ' checked' : '');
        li.innerHTML = `
            <div class="handle">☒</div>
            <input type="checkbox" onchange="toggleCheck(this)" ${checked ? 'checked' : ''} style="
                width:20px; height:20px; margin-right:10px;">
            <span>${text}</span>
            <button class="delete-btn" onclick="deleteItem(this)">&times;</button>
        `;
        container.appendChild(li);
    }

    function addItem() {
        if (!input.value.trim()) return;
        renderItem(input.value, false);
        input.value = '';
        saveToServer();
    }

```

```
}

function toggleCheck(cb) {
    cb.parentElement.classList.toggle('checked');
    saveToServer();
}

function deleteItem(btn) {
    btn.parentElement.remove();
    saveToServer();
}

window.onload = loadFromServer;
</script>
</body>
</html>
```

サーバーの所有者を変更する

```
sudo chown -R www-data:www-data /var/www/html
```

```
sudo chmod -R 755 /var/www/html
```