

Résumé jour 1 semaine 3

Configurer la connexion à la base de données

1. Créer un fichier de configuration pour la base de données

Dans ton projet backend, crée un dossier config

build	31/01/2025 23:29
config	31/01/2025 13:43
node_modules	31/01/2025 23:29
public	31/01/2025 13:03

Dans ce dossier, crée un fichier database.js avec le contenu suivant :

```
Users > alexl > Documents > menz-biz-backend > config > JS database.js > ...  
const { Pool } = require('pg');  
  
const pool = new Pool({  
  connectionString: process.env.DATABASE_URL,  
  ssl: {  
    rejectUnauthorized: false, // Nécessaire pour Supabase  
  },  
});  
  
module.exports = pool;
```

2. Utiliser le pool de connexions dans ton backend

Dans tes routes ou contrôleurs, importe le pool de connexions :

```
const express = require('express');  
const pool = require('../config/database');  
const router = express.Router();
```

Utilise pool.query() pour exécuter des requêtes SQL. Par exemple :

```

server.js  JS database.js  JS invoices.js X
Users > alexl > Documents > menz-biz-backend > routes > JS invoices.js > ...
1  const express = require('express');
2  const pool = require('../config/database');
3  const router = express.Router();
4
5  // Route pour récupérer toutes les factures
6  router.get('/', async (req, res) => {
7    try {
8      const result = await pool.query('SELECT * FROM invoices');
9      res.json(result.rows); // Renvoie les résultats au client
10   } catch (err) {
11     console.error(err);
12     res.status(500).send('Erreur de base de données');
13   }
14 });
15
16 // Route pour créer une nouvelle facture
17 router.post('/', async (req, res) => {
18   const { client_name, amount, status } = req.body;
19   try {
20     const result = await pool.query(
21       'INSERT INTO invoices (client_name, amount, status) VALUES ($1, $2, $3)'
22       [client_name, amount, status]
23     );
24     res.json(result.rows[0]); // Renvoie la facture créée
25   } catch (err) {
26     console.error(err);
27     res.status(500).send('Erreur de base de données');
28   }
29 });
30
31 module.exports = router;

```

Ici, on utilise `pool.query()` pour exécuter des requêtes SQL :

- `SELECT * FROM invoices` : Récupère toutes les factures.
- `INSERT INTO invoices` : Ajoute une nouvelle facture.

2. Importer les routes dans server.js

Dans ton fichier `server.js`, importe les routes et configure Express :

```

> Users > alexl > Documents > menz-biz-backend > JS server.js > ...
1  const express = require('express');
2  const invoicesRouter = require('../routes/invoices'); // Importe les routes des fa
3  require('dotenv').config(); // Charge les variables d'environnement
4
5  const app = express();
6  const port = process.env.PORT || 3000;
7
8  // Middleware pour parser le JSON
9  app.use(express.json());
10
11 // Utilise les routes des factures
12 app.use('/invoices', invoicesRouter);
13
14 // Démarrer le serveur
15 app.listen(port, () => {
16   console.log(`Serveur démarré sur http://localhost:${port}`);
17 });

```

Erreur rencontrée :

```
alex1@DESKTOP-290K77K MINGW64 ~/Documents/menz-biz-backend (main)
$ node server.js
node:internal/modules/cjs/loader:1247
  throw err;
  ^

Error: Cannot find module '../config/database'
Require stack:
- C:\Users\alex1\Documents\menz-biz-backend\server.js
  at Function._resolveFilename (node:internal/modules/cjs/loader:1244:15)
  at Function._load (node:internal/modules/cjs/loader:1070:27)
  at TracingChannel.traceSync (node:diagnostics_channel:322:14)
  at wrapModuleLoad (node:internal/modules/cjs/loader:217:24)
  at Module.require (node:internal/modules/cjs/loader:1335:12)
  at require (node:internal/modules/helpers:136:16)
  at Object.<anonymous> (C:\Users\alex1\Documents\menz-biz-backend\server.js:5:14)
  at Module._compile (node:internal/modules/cjs/loader:1562:14)
  at Object..js (node:internal/modules/cjs/loader:1699:10)
  at Module.load (node:internal/modules/cjs/loader:1313:32) {
  code: 'MODULE_NOT_FOUND',
  requireStack: [ 'C:\\Users\\alex1\\Documents\\menz-biz-backend\\server.js' ]
}

Node.js v22.13.1
```

solution :

```
3  const bodyParser = require('body-parser');
4  const invoicesRoutes = require('./routes/invoices'); // Invoices routes
5  const pool = require('../config/database'); // Connexion à la base de données
6
7  const app = express();
8
9  // Middleware pour parser le corps des requêtes
10 app.use(bodyParser.json());
11 app.use(bodyParser.urlencoded({ extended: true }));
12
13 // Routes
14 app.use('/api', invoicesRoutes);
15
16 // Démarrage du serveur
17 const PORT = 3000;
18 app.listen(PORT, () => {
19   console.log(`Serveur démarré sur le port ${PORT}`);
20 });
```

fin jour 1