

# React, Node et MySQL Login (Partie 11)

## Amélioration de db.router et de db.service

**Étape 19 - Correction :** Création du routeur pour l'accès aux tables de la DB.

voir git : <https://github.com/laurentbedu/todo-api/tree/TP-11>

**Étape 20 :** Amélioration de db.service et de db.router

Dans db.service nous créons les 6 méthodes suivantes que nous allons ensuite utiliser dans le routeur

```
api > services > database.service.js > ...
19  async function query(sql) {
20    const connection = await connect();
21    const [rows] = await connection.query(sql);
22    return rows;
23  }
24
25  > async function selectAll(table) { ...
40  }
41
42  > async function selectOne(table, id) { ...
44  }
45
46  > async function createOne(table) { ...
48  }
49
50  > async function updateOne(table, id) { ...
52  }
53
54  > async function softDeleteOne(table, id) { ...
56  }
57
58  > async function hardDeleteOne(table, id) { ...
60  }
61
62  module.exports = {
63    query,
64    selectAll,
65    selectOne,
66    createOne,
67    updateOne,
68    softDeleteOne,
69    hardDeleteOne,
70  };
71
```

Nous modifions le db.router pour les utiliser, exemple pour la méthode selectAll

```
api > routers > JS db.router.js > ...
1  const express = require("express");
2  const dbRouter = express.Router();
3  const dbService = require("../services/database.service");
4
5  dbRouter.get("/:table", async (req, res) => {
6      const { table } = req.params;
7      const dbResp = await dbService.selectAll(table);
8      res.status(dbResp?.result ? 200 : 400).json(dbResp);
9      // const sql = `SELECT * FROM ${table} WHERE is_deleted = 0`;
10     // await query(sql)
11     //     .then((data) => {
12     //         res.json({
13     //             data,
14     //             result: true,
15     //             message: `all rows of table ${table} have been selected`,
16     //         });
17     //     })
18     //     .catch((err) => {
19     //         res.json({ data: null, result: false, message: err.message });
20     //     });
21 });
```

Il nous reste à reprendre (et adapter) le code qui était dans le routeur (code commenté ci-dessus) pour le déplacer dans la méthode selectAll du service

```
24
25  async function selectAll(table) {
26      const sql = `SELECT * FROM ${table} WHERE is_deleted = 0`;
27      let resp;
28      await query(sql)
29          .then((data) => {
30              resp = {
31                  data,
32                  result: true,
33                  message: `all rows of table ${table} have been selected`,
34              };
35          })
36          .catch((err) => {
37              resp = { data: null, result: false, message: err.message };
38          });
39      return resp;
40  }
```

**Étape 21 :** En vous aidant de l'étape 20, codez les 5 méthodes restantes pour les utiliser dans le routeur.