

React, Node et MySql Login (Partie 1)

Étape 1 : Créer une route sur l'API pour le login et la tester.

La route doit être créée en POST (pour pouvoir récupérer les données envoyées du Front) tout en haut de index.js (Afin d'être "interceptée" avant les autres requêtes)

```
index.js > ...
1  const express = require("express");
2  const app = express();
3  app.use(express.json());
4  const { query } = require("../api/services/database.service");
5  const cors = require("cors");
6  app.use(cors());
7
8  app.post("/login", async (req, res) => {
9    const { body } = req;
10    //console.log(body);
11    const sql = `SELECT * FROM customer
12                WHERE is_deleted = 0
13                AND email = '${body.email}'`;
14    await query(sql)
15      .then((json) => {
16        //console.log(json);
17        const user = json.length === 1 ? json.pop() : null;
18        //console.log(user);
19        if (user && user.pincodes === body.pincodes) {
20          const { id, email } = user;
21          const data = { id, email };
22          res.json({ data, result: true, message: `Login OK` });
23        } else {
24          throw new Error("Bad Login");
25        }
26      })
27      .catch((err) => {
28        res.json({ data: null, result: false, message: err.message });
29      });
30  });
31
32  app.get("/:table", async (req, res) => {
33    const { table } = req.params;
34    const sql = `SELECT * FROM ${table} WHERE is_deleted = 0`;
```

Ln 9 : Récupération du body de la requête HTTP.

Ln 14 : Requête en DB pour récupérer le(s) compte(s) correspondant à l'email reçu dans le body de la requête HTTP.

Ln 17: Une fois la réponse obtenue au format json, on récupère la ligne (si elle existe) dans user

Ln 19: si un user (avec l'email reçu dans la requête HTTP) existe et si le pincodes stocké en DB correspond au pincodes reçu dans la requête HTTP, on extrait l'id et l'email (Ln 20 et 21) que l'on renvoie dans la réponse (Ln 22)

Sinon on déclenche une erreur (Ln 24) qui sera interceptée par le catch (Ln 27) qui renverra une réponse "Bad Login" (Ln 28)

Étape 2 : Tester avec ThunderClient

Décommenter les console.log et mettez des points d'arrêt pour comprendre le cheminement et la logique

Tester tous les cas possible (ci-dessous listing des différents cas de test)

| Test de la route POST/login de l'API Rest | | | |
|---|---|--------------------|-------------------|
| Cas | Description du cas de test | Résultats attendus | Résultats obtenus |
| 1 | Récéption d'un email correspondant à un compte stocké en DB et du pincode associé | Login OK | Login OK |
| 2 | Récéption d'un email correspondant à un compte stocké en DB avec un pincode différent | Bad Login | Bad Login |
| 3 | Récéption d'un email ne correspondant pas à un compte stocké en DB | Bad Login | Bad Login |

Cas 1 :

POST

Status: 200 OK Size: 84 Bytes Time: 3.07 s

Query Headers² Auth **Body¹** Tests Pre R

Json Xml Text Form Form-encode Graph

Json Content Format

```
1 {
2   "email": "bedulaurent@gmail.com",
3   "pincode": "1234"
4 }
```

Response Headers⁷ Cookies Results

```
1 {
2   "data": {
3     "id": 1,
4     "email": "bedulaurent@gmail.com"
5   },
6   "result": true,
7   "message": "Login OK"
8 }
```

Cas 2 :

POST

Status: 200 OK Size: 50 Bytes Time: 5 ms

Query Headers² Auth **Body¹** Tests Pre R

Json Xml Text Form Form-encode Graph


Json Content Format

```
1 {
2   "email": "bedulaurent@gmail.com",
3   "pincode": "8888"
4 }
```

Response Headers⁷ Cookies Result

```
1 {
2   "data": null,
3   "result": false,
4   "message": "Bad Login"
5 }
```

Cas 3 :

| | | | | | | | | | |
|--|----------------------|-----------------------------|-------------------|--|-------|--|----------------------|---------|--------|
| POST  | | http://localhost:5000/login | Send | Status: 200 OK Size: 50 Bytes Time: 4 ms | | | | | |
| Query | Headers ² | Auth | Body ¹ | Tests | Pre R | Response | Headers ⁷ | Cookies | Result |
| <div>Json Xml Text Form Form-encode Graph</div> <div>Json Content Format</div> <div><pre>1 { 2 "email": "existepas@gmail.com", 3 "pincode": "8888" 4 }</pre></div> | | | | | | <pre>1 { 2 "data": null, 3 "result": false, 4 "message": "Bad Login" 5 }</pre> | | | |