coursNodeJs

04-API

Installations

PS C:\Workspaces\coursNodeJs\04-API> **npm init -y**

PS C:\Workspaces\coursNodeJs\04-API> npm install nodemon express cors pg

création: ***.gitignore***

PS C:\Workspaces\coursNodeJs\04-API> **npm start**

Lancer l'extension Thunder NewRequest

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
    res.send('Hello World!')
  })

  app.listen(port, () => {
    console.log(`Example app listening on port ${port}`)
  })
```

PS C:\Workspaces\coursNodeJs\04-API> **psql -U postgres -h localhost**

postgres=# **\l**

```
const {Client} = require('pg')

const client = new Client({
    host: "localhost",
    user: "postgres",
    port: 5432,
    password: "postgres",
    database: "my_first_api"
})

module.exports = client
```

postgres=# **CREATE DATABASE my_first_api;**

postgres=# **\c my_first_api;** Vous êtes maintenant connecté à la base de données « my_first_api » en tant qu'utilisateur « postgres ».

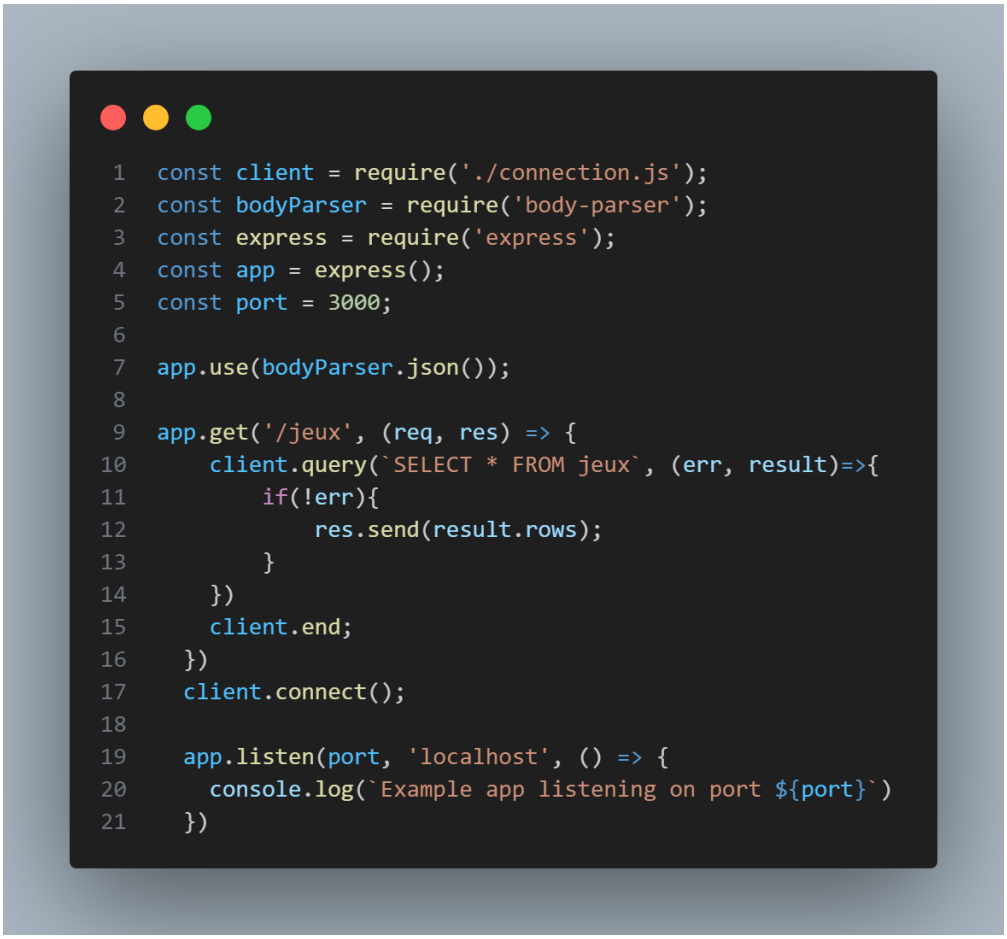
Creation TABLE jeux

my_first_api=# CREATE TABLE jeux (my_first_api(# id SERIAL PRIMARY KEY, my_first_api(# name VARCHAR (50) NOT NULL, my_first_api(# release_date DATE,
my_first_api(# console_id INT DEFAULT 1 my_first_api(#); CREATE TABLE

Creation du fichier de données : addGame.sql

```
INSERT INTO jeux ('name', 'release_date') VALUES
(`Tetris`, '1989-06-14'),
(`The Legend of Zelda: Link's Awakening`, '1993-06-06'),
(`Super Mario Land`, '1989-04-21'),
(`Pokémon Red Version`, '1996-09-28'),
```

Modification du fichier app pour le lier à body-parser

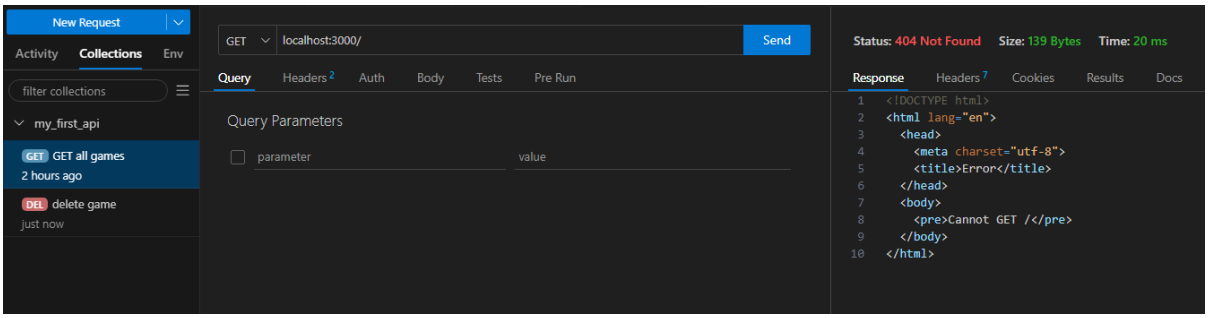


```
const client = require('./connection.js');
const bodyParser = require('body-parser');
const express = require('express');
const app = express();
const port = 3000;

app.use(bodyParser.json());

app.get('/jeux', (req, res) => {
    client.query(`SELECT * FROM jeux`, (err, result)=>{
        if(!err){
            res.send(result.rows);
        }
    })
    client.end;
  })
  client.connect();

  app.listen(port, 'localhost', () => {
    console.log(`Example app listening on port ${port}`)
  })
```

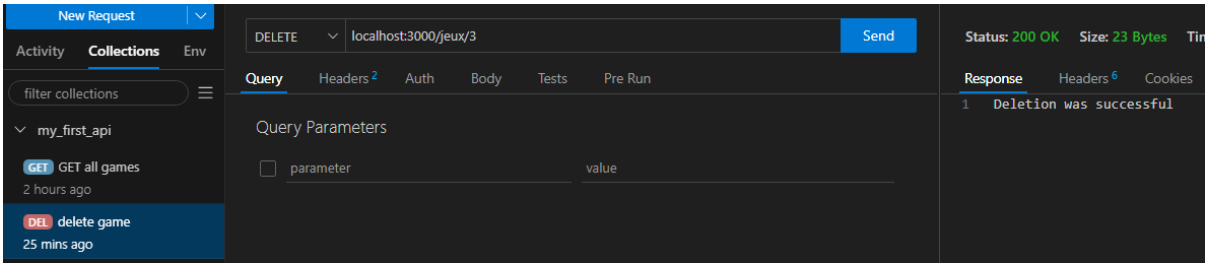
Installation de body-parser pour afficher les données dans le fichier json

PS C:\Workspaces\coursNodeJs\04-API> **npm install body-parser;**

Gestion de THUNDER REQUEST pour afficher la table, supprimer une ligne...



```
// effacer un jeu
app.delete('/jeux/:id', (req, res)=> {
let insertQuery = `delete from jeux where id=${req.params.id}`

client.query(insertQuery, (err, result)=>{
    if(!err){
        res.send('Deletion was successful')
    }
    else{ console.log(err.message) }
})
client.end;
})
```



Pour gérer les erreurs si le jeu n'existe pas :

```
// afficher un jeu
app.get('/jeux/:id', (req, res) => {
    client.query(`SELECT * FROM jeux WHERE id=${req.params.id}`,
    // conditions avec une TURNER (au lieu de if...)
    (err, data) => !err ? !data.rows ? res.send('Game was not found') : res.send(data.rows[0]) : res.send(err.message));
    // version longue avec if-else
    // {
    //     if (!err) {
    //         if (data.rows.length === 0) {
    //             res.send('Game was not found')
    //         } else {
    //             res.send(data.rows[0])
    //         }
    //     } else {
    //         res.send(err.message)
    //     }
    // }
    client.end;
})
```

```
// effacer un jeu
app.delete('/jeux/:id', (req, res) => {
    let insertQuery = `delete from jeux where id=${req.params.id}`;

    client.query(insertQuery, (err, result) => {
        if (!err) {
            if (result.rowCount === 0) {
                res.send('Deletion was not found')
            } else {
                res.send('Deletion was successful')
            }
        }
        else { console.log(err.message) }
    })
    client.end;
})
```

Insertion d'un nouveau jeu

```
//ajouter un jeu
app.post('/jeux', (req, res) => {
    const jeu = req.body;
    let insertQuery = `insert into jeux(name, release_date) values('${jeu.name}', '${jeu.release_date}')`

    client.query(insertQuery, (err, result) => {
        if (!err) {
            res.send('Insertion was successful')
        }
        else { console.log(err.message) }
    })
    client.end;
})
```