Node.js opgave dagbog 29-11-2023

|  |  |
| --- | --- |
|  | 1. Give admin fuld permission: GRANT ALTER ON \*.\* TO 'admin'@'localhost'; på MySQL |
|  | På VS code   1. Opdatere npm : npm install mysql@latest |
|  | På VS code   1. Installere igen: npm install mysql2 |
|  | 1. Opdatere MySQL Server authentication: ALTER USER 'admin'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'Test1234'; |
|  | 1. Undefined Port rettet fra index.js: |
|  | 1. Til: Den løste ”undefined port” |
|  | . For at teste om den henter data lavede jeg på index.js:  import db from './Config/db.config.js' |
|  | 1. Virker fino… |
| import db from './Config/db.config.js'  // Execute a database query to select titles from the 'song' table  db.query(`SELECT title FROM song`, (err, result) => {  if (err) {  console.error('Error executing database query:', err);  } else {  console.log('Database Query Result:', result);  }  // Log the value of DB\_PORT after the database query has completed  console.log('DB\_PORT:', process.env.DB\_PORT);  }); | 1. Selve koden I index.js |

## OPGAVE

## Opgave: Byg en MVC setup til artister:

|  |  |
| --- | --- |
| // artist.controller.js  import express from 'express';  import db from '../Config/db.config.js';  const router = express.Router();  // GET: List alle artister  router.get('/artists', (req, res) => {  const sql = 'SELECT \* FROM artist';  db.query(sql, (err, result) => {  if (err) {  return res.status(500).json({ error: 'Internal Server Error' });  }  return res.json(result);  });  });  // GET: Artist detaljer  router.get('/artists/:id', (req, res) => {  const artistId = req.params.id;  const sql = 'SELECT \* FROM artist WHERE id = ?';  db.query(sql, [artistId], (err, result) => {  if (err) {  return res.status(500).json({ error: 'Internal Server Error' });  }  if (result.length === 0) {  return res.status(404).json({ error: 'Artist not found' });  }  return res.json(result[0]);  });  });  // POST: Opret artist  router.post('/artists', (req, res) => {  const { name, genre } = req.body;  const sql = 'INSERT INTO artist (name, genre) VALUES (?, ?)';  db.query(sql, [name, genre], (err, result) => {  if (err) {  return res.status(500).json({ error: 'Internal Server Error' });  }  return res.json({ message: 'Artist created successfully', artistId: result.insertId });  });  });  // PUT: Opdater artist  router.put('/artists/:id', (req, res) => {  const artistId = req.params.id;  const { name, genre } = req.body;  const sql = 'UPDATE artist SET name = ?, genre = ? WHERE id = ?';  db.query(sql, [name, genre, artistId], (err, result) => {  if (err) {  return res.status(500).json({ error: 'Internal Server Error' });  }  if (result.affectedRows === 0) {  return res.status(404).json({ error: 'Artist not found' });  }  return res.json({ message: 'Artist updated successfully' });  });  });  // DELETE: Slet artist  router.delete('/artists/:id', (req, res) => {  const artistId = req.params.id;  const sql = 'DELETE FROM artist WHERE id = ?';  db.query(sql, [artistId], (err, result) => {  if (err) {  return res.status(500).json({ error: 'Internal Server Error' });  }  if (result.affectedRows === 0) {  return res.status(404).json({ error: 'Artist not found' });  }  return res.json({ message: 'Artist deleted successfully' });  });  });  export default router; | 1. lave artist.controller.js |
| import ArtistRouter from './Controllers/artist.controller.js'  app.use(express.json());  app.use(ArtistRouter); | 1. Tilføje import og app.use i index.js |
| // TESTE HENTE ARTISTER:  import db from './Config/db.config.js'  // Execute a database query to select titles from the 'song' table  db.query(`SELECT \* FROM artist`, (err, result) => {  if (err) {  console.error('Error executing database query:', err);  } else {  console.log('Database Query Result:', result);  }  // Log the value of DB\_PORT after the database query has completed  console.log('DB\_PORT:', process.env.DB\_PORT);  }); | 1. Teste den |