

Tietokannat - Tehtävä 16

```
//Importing required modules and libraries
const express = require('express');
const mysql = require('mysql');
const bodyParser = require('body-parser');
const app = express();

//Configure the database connection
const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'Roope123',
  database: 'kirjasto'
});

//Connecting to the database
connection.connect((err) => {
  if (err) throw err;
  console.log('Connected to MySQL Database');
});

//Middleware
app.use(bodyParser.urlencoded({extended: false}));
app.use(bodyParser.json());

//GET request to fetch all books
app.get('/books', (req, res) => {
  const sql = 'SELECT * FROM teos';
  connection.query(sql, (err, results) => {
    if (err) throw err;
    res.send(results);
  });
});

//GET request to fetch a specific book by id
app.get('/books/:id', (req, res) => {
  const id = req.params.id;
  const sql = 'SELECT * FROM teos WHERE id = ${id}';
  connection.query(sql, (err, result) => {
    if (err) throw err;
    res.send(result);
  });
});

//POST request to add a new book
app.post('/books', (req, res) => {
  const book = req.body;

  if (!book.name || !book.author) {
    console.log(book);
    console.log("Required info missing.");

    return;
  }

  const sql = 'INSERT INTO teos (nimi, tekija) VALUES (" + book.name + ', ' +
  book.author + ')';
  connection.query(sql, book, (err, result) => {
    if (err) throw err;
    res.send('Book added successfully');
  });
});

//PUT request to update a book
app.put('/books/:id', (req, res) => {
  const id = req.params.id;
  const book = req.body;

  if (!id || !book.name || !book.author) {
    console.log(book);
    console.log("Required info missing.");


    return;
  }

  const sql = 'UPDATE teos SET nimi = "${book.name}", tekija = ${book.author}
  WHERE id = ${id}';
  connection.query(sql, book, (err, result) => {
    if (err) throw err;
    res.send('Book updated successfully');
  });
});

//DELETE request to delete a book
app.delete('/books/:id', (req, res) => {
  const id = req.params.id;
  const sql = 'DELETE FROM teos WHERE id = ${id}';
  connection.query(sql, (err, result) => {
    if (err) throw err;
    res.send('Book deleted successfully');
  });
});

//Starting the server
app.listen(3000, () => {
  console.log('Server started on port 3000');
});
```

Tietokannat - Tehtävä 16


HOPPSCOTCH  Star 50,135

REST

GraphQL

Realtime

Settings

GET 

http://localhost:3000/books

Parameters

Body

Headers

Authorization

Pre-request Script

Tests

Query Parameters

Parameter 3	Value 3
-------------	---------

Status: 200 • OK Time: 41 ms Size: 296 B

JSON

Raw

Headers 7

Test Results

Response Body

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

[
 {
 "id": 1,
 "nimi": "Tuntematon sotilas",
 "tekija": 1
 },
 {
 "id": 2,
 "nimi": "Kalevala",
 "tekija": 2
 },
 {
 "id": 3,
 "nimi": "MAOL-taulukot : Matematiikka, fysiikka, kemia",
 "tekija": 3
 },
 {
 "id": 5,
 "nimi": "Pietarin elämänkerta - Joku Edition",
 "tekija": 4
 },
 {

Tietokannat - Tehtävä 16

HOPPSCOTCH

Star50,135

REST

GraphQL

Realtime

Settings

GET

http://localhost:3000/books/5

ParametersBodyHeadersAuthorizationPre-request ScriptTests

Query Parameters

Parameter 3

Value 3

Status: 200 • OKTime: 95 msSize: 68 B

JSONRawHeaders7Test Results

Response Body

1

2

3

4

5

6

7

[
 {
 "id": 5,
 "nimi": "Pietarin elämänkerta - Joku Edition",
 "tekija": 4
 }
]

HOPPSCOTCH

Star50,135

REST

GraphQL

Realtime

Settings

POST

http://localhost:3000/books

ParametersBodyHeadersAuthorizationPre-request ScriptTests

Content Typeapplication/jsonOverride

Raw Request Body

1

2

3

4

{
 "name": "Miksi tehtävien teko on tärkeää",
 "author": 1
}

Status: 200 • OKTime: 120 msSize: 23 B

HTMLRawHeaders7Test Results

Response Body

1

Book added successfully

