

## Viikko 7: Tehtävä 17 - Tietokantayhteys ja CRUD Nodella ja Expressillä

Luodaan tarvittavat taulut ja esitellään ne describe komennolla:

```
mysql> create table book(  
  -> id_book INT primary key auto_increment,  
  -> name VARCHAR(255),  
  -> author VARCHAR(255),  
  -> isbn VARCHAR(20));  
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create table borrower(  
  -> id_borrower INT primary key auto_increment,  
  -> fname VARCHAR(40),  
  -> lname VARCHAR(40),  
  -> streetaddress VARCHAR(40));  
Query OK, 0 rows affected (0.03 sec)
```

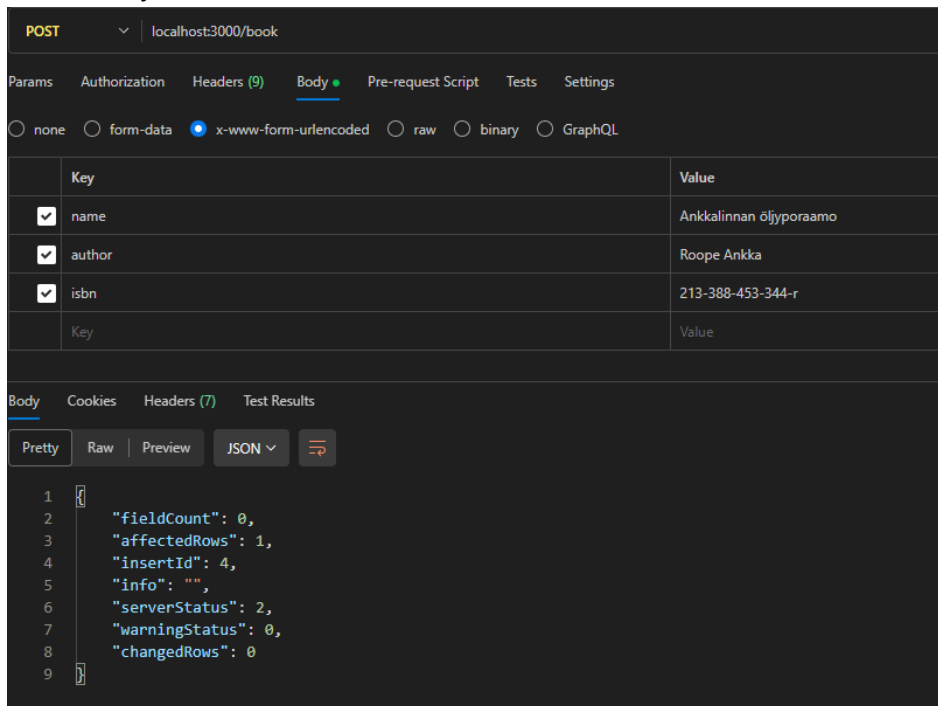
```
mysql> desc book  
  -> ;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id_book | int           | NO   | PRI | NULL    | auto_increment |  
| name    | varchar(255)  | YES  |     | NULL    |                 |  
| author  | varchar(255)  | YES  |     | NULL    |                 |  
| isbn    | varchar(20)   | YES  |     | NULL    |                 |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

```
mysql> desc borrower;  
+-----+-----+-----+-----+-----+-----+  
| Field          | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id_borrower    | int           | NO   | PRI | NULL    | auto_increment |  
| fname          | varchar(40)   | YES  |     | NULL    |                 |  
| lname          | varchar(40)   | YES  |     | NULL    |                 |  
| streetaddress  | varchar(40)   | YES  |     | NULL    |                 |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.01 sec)
```

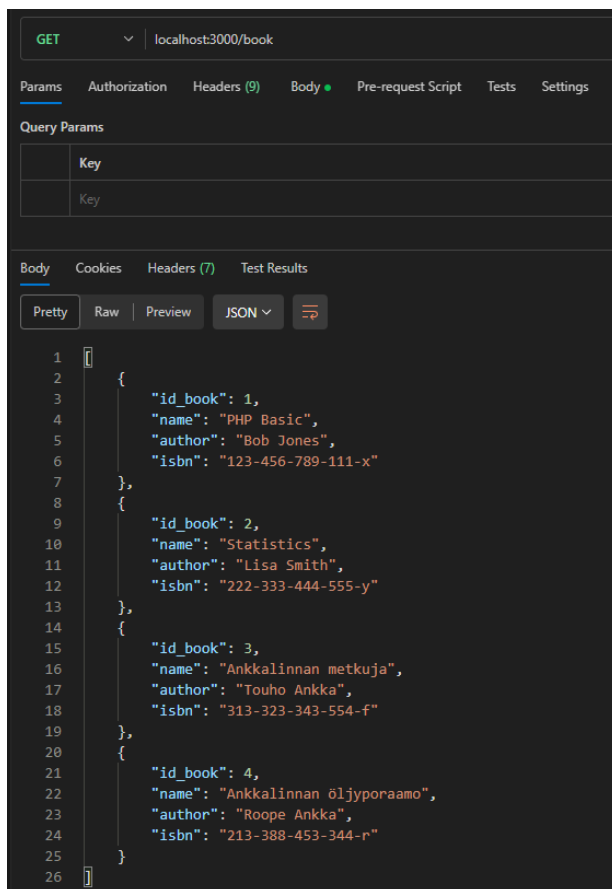
## Viikko 7: Tehtävä 17 - Tietokantayhteys ja CRUD Nodella ja Expressillä

Tämän jälkeen suoritettiin GET, POST, PUT ja DELETE käskyt Postman-sovelluksella:

Lisätään kirja POST-metodilla:

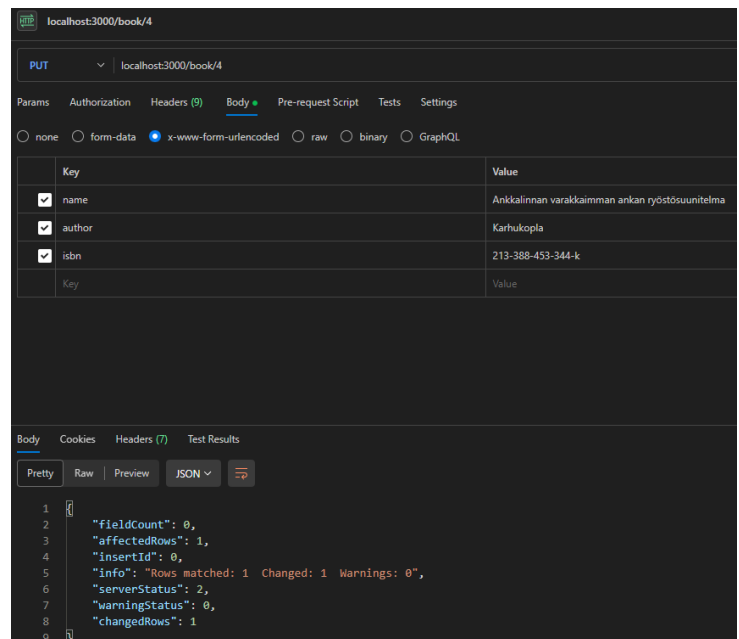


Esitellään kaikki kirjat GET-metodilla:



Päivitetään kirjan (`id_book = 4`) tiedot

PUT-metodilla:



Tietokannat ja rajapinnat

TVT23SPL

Juho Kupila

## Viikko 7: Tehtävä 17 - Tietokantayhteys ja CRUD Nodella ja Expressillä

Päivitys todettiin laittomaksi, joten poistetaan kirja tietokannasta DELETE-metodilla, jossa

id\_book = 4:

The screenshot shows a REST client interface with the following details:

- URL:** localhost:3000/book/4
- Method:** DELETE
- Query Params:** A table with two rows, each containing a 'Key' column.
- Body:** The response is displayed in JSON format, showing a successful deletion with the following fields:

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "info": "",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 0
9 }
```

## Viikko 7: Tehtävä 17 - Tietokantayhteys ja CRUD Nodella ja Expressillä

Suoritetaan nyt metodit borrower-taululle:

POST-metodi (lisätään tietoja borrower-tauluun):

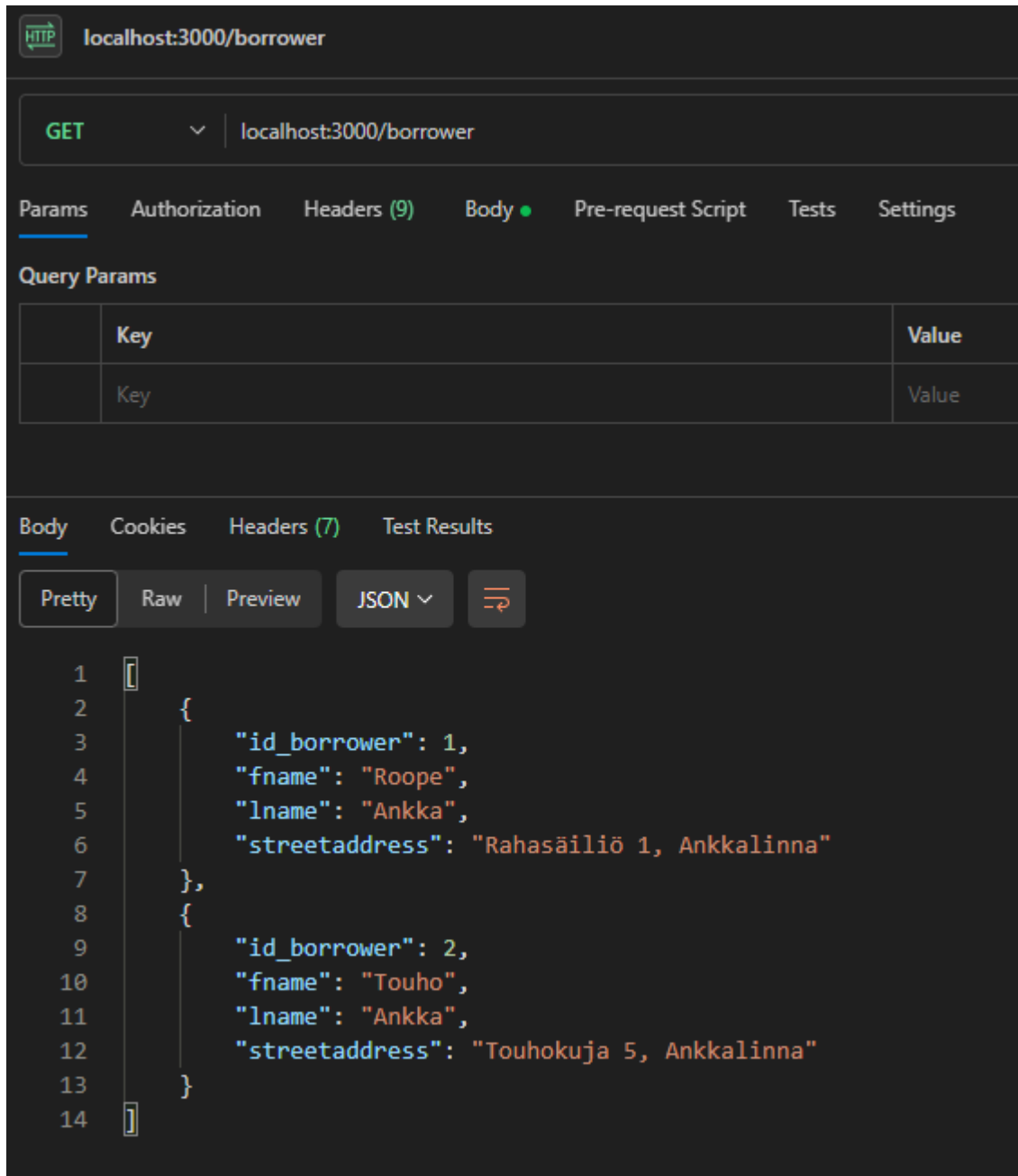
The screenshot shows a REST client interface with a POST request to `localhost:3000/borrower`. The 'Body' tab is selected, and the 'x-www-form-urlencoded' format is chosen. The request body contains the following form data:

	Key	Value
<input checked="" type="checkbox"/>	fname	Touho
<input checked="" type="checkbox"/>	lname	Ankka
<input checked="" type="checkbox"/>	streetaddress	Touhokuja 5, Ankkalinn
	Key	Value

The response is shown in the 'Body' tab, displaying a JSON object in 'Pretty' format:

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 2,
5   "info": "",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 0
9 }
```

GET-metodi (haetaan kaikki tiedot borrower-taulusta):



The screenshot shows a web browser's developer tools interface. The top bar indicates the URL is `localhost:3000/borrower` and the method is `GET`. The 'Params' tab is selected, showing a table with two columns: 'Key' and 'Value'. Below this, the 'Body' tab is selected, showing a JSON response. The JSON is formatted as a pretty-printed array of two objects. The first object has the following properties: `id_borrower`: 1, `fname`: "Roope", `lname`: "Ankka", and `streetaddress`: "Rahasäiliö 1, Ankkalinn". The second object has the following properties: `id_borrower`: 2, `fname`: "Touho", `lname`: "Ankka", and `streetaddress`: "Touhokuja 5, Ankkalinn".

Key	Value
Key	Value

```
1 [
2   {
3     "id_borrower": 1,
4     "fname": "Roope",
5     "lname": "Ankka",
6     "streetaddress": "Rahasäiliö 1, Ankkalinn"
7   },
8   {
9     "id_borrower": 2,
10    "fname": "Touho",
11    "lname": "Ankka",
12    "streetaddress": "Touhokuja 5, Ankkalinn"
13  }
14 ]
```

## Viikko 7: Tehtävä 17 - Tietokantayhteys ja CRUD Nodella ja Expressillä

PUT-metodi (päivitetään tiedot, jossa id\_borrower = 2):

The screenshot shows a REST client interface with a PUT request to `localhost:3000/borrower/2`. The request body is configured as `x-www-form-urlencoded` with the following data:

Key	Value
fname	Pelle
lname	Peloton
streetaddress	Keksintökuja 2, Ankkalinn

The response body is shown in JSON format:

```
{
  "fieldCount": 0,
  "affectedRows": 1,
  "insertId": 0,
  "info": "Rows matched: 1 Changed: 1 Warnings: 0",
  "serverStatus": 2,
  "warningStatus": 0,
  "changedRows": 1
}
```

DELETE-metodi(poistetaan tiedot, jossa id\_borrower = 2):

The screenshot shows a REST client interface with the following details:

- URL:** localhost:3000/borrower/2
- Method:** DELETE
- Params:** Authorization, Headers (9), Body, Pre-request Script, Tests
- Query Params:** A table with two rows, each containing a 'Key' column.
- Body:** Pretty, Raw, Preview, JSON (selected), and a refresh icon.
- Response:** A JSON object with the following fields:

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "info": "",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 0
9 }
```