

Tehtävät 22 ja 24 valmistelut:

Borrower-taulun luonti:

```
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 (0.02 sec)
```

Todetaan, että taulu näyttää oikealta.

```
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 (0.00 sec)
```

Henri Kemppainen TVT25SPL

Postman-haku Hannu-käyttäjänä

The screenshot displays the Postman interface for a REST client. The top bar shows the URL `localhost:3000/book` and the HTTP method `GET`. The request body is a JSON object: `{ "username": "Hannu", "password": "onnekas" }`. The response status is `200 OK` with a time of `5 ms` and a size of `401 B`. The response body is a JSON object: `{ "success": true, "message": "Login OK", "username": "Hannu", "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Ikhbm51IiwiaWF0IjoxNzcxMzI3OTAxLCJleHAiOjE3NzEzMDF9.JQ3L2UYm5LtdhIfdR3A26J6GmAXyVMsBNNIKGd1dAKg" }`. The bottom panel shows the terminal output of the `nodemon start` command, displaying the results of the `GET /favicon.ico`, `POST /user`, `POST /login`, and `GET /book` requests.

```
PS C:\Users\heke_OneDrive\Työpöytä\Henri koulu palautukset\Tietokannat\Viikko 7> nodemon start
GET /favicon.ico 401 0.275 ms - 12
POST /user 200 64.256 ms - 242
success
POST /login 200 58.355 ms - 220
[
  {
    id_book: 1,
    name: 'PHP Basic',
    author: 'Bob Jones',
    isbn: '123-456-789-111-x'
  },
  {
    id_book: 2,
    name: 'Statistics',
    author: 'Lisa Smith',
    isbn: '222-333-444-555-y'
  }
]
GET /book 200 1.742 ms - 165
```

Lisätään uusi tietue book-tauluun käyttäen postmania. Siirsin app.use('/book', bookRouter):in pois salauksen takaa testailun ajaksi.

The image shows the Postman interface for a POST request to `localhost:3000/book`. The request body is set to `x-www-form-urlencoded` with the following data:

Key	Value
name	Testikirja
author	Testikirjailija
isbn	123-456

The response is a JSON object: `{ "fieldCount": 0, "affectedRows": 1, "insertId": 5, "info": "", "serverStatus": 2, "warningStatus": 0, "changedRows": 0 }`. The status is 200 OK, and the time taken is 41 ms.

Below the Postman interface, a terminal window shows the command `nodemon start` being executed in a Node.js environment. The output shows the application starting successfully and the POST request being received.

Ja sinne se ilmestyi!

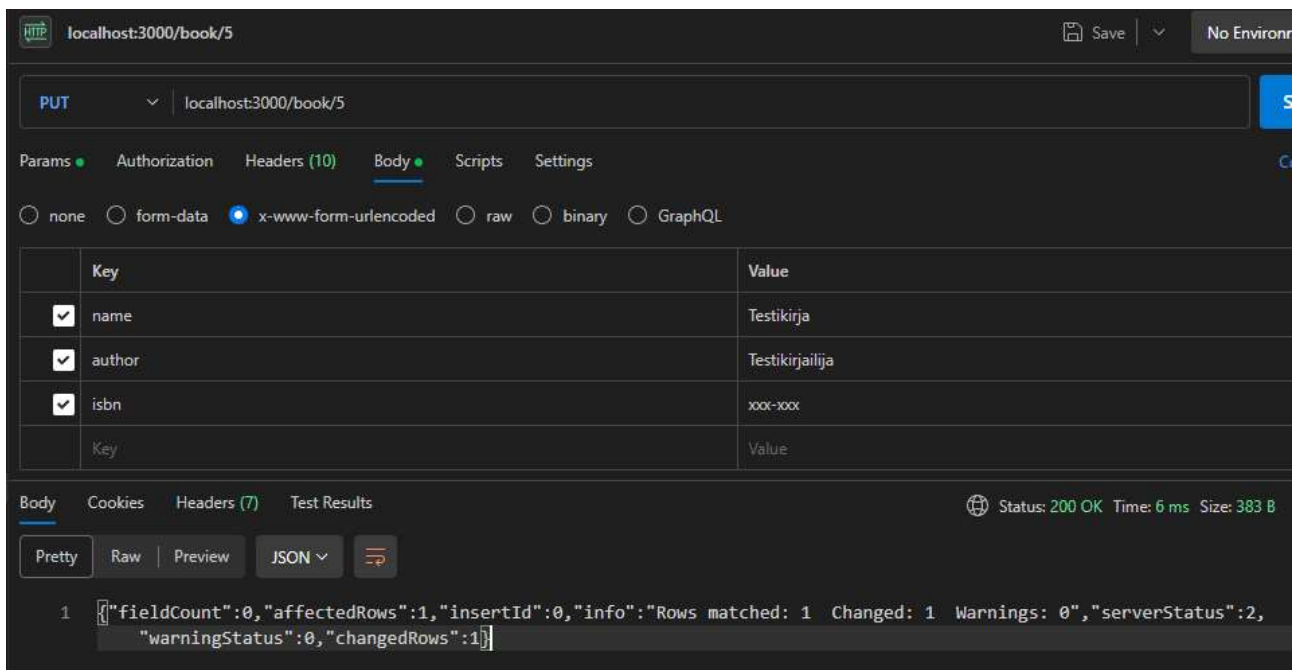
The image shows the Postman interface for a GET request to `localhost:3000/book`. The response is a JSON array of book records:

```
[{"id_book": 1, "name": "PHP Basic", "author": "Bob Jones", "isbn": "123-456-789-111-x"}, {"id_book": 2, "name": "Statistics", "author": "Lisa Smith", "isbn": "222-333-444-555-y"}, {"id_book": 5, "name": "Testikirja", "author": "Testikirjailija", "isbn": "123-456"}]
```

The status is 200 OK, and the time taken is 37 ms. The size of the response is 479 B.

Below the Postman interface, a terminal window shows the command `nodemon start` being executed. The output shows the application starting successfully and the GET request being received, displaying the book records in the terminal.

Tietueen muokkaus:

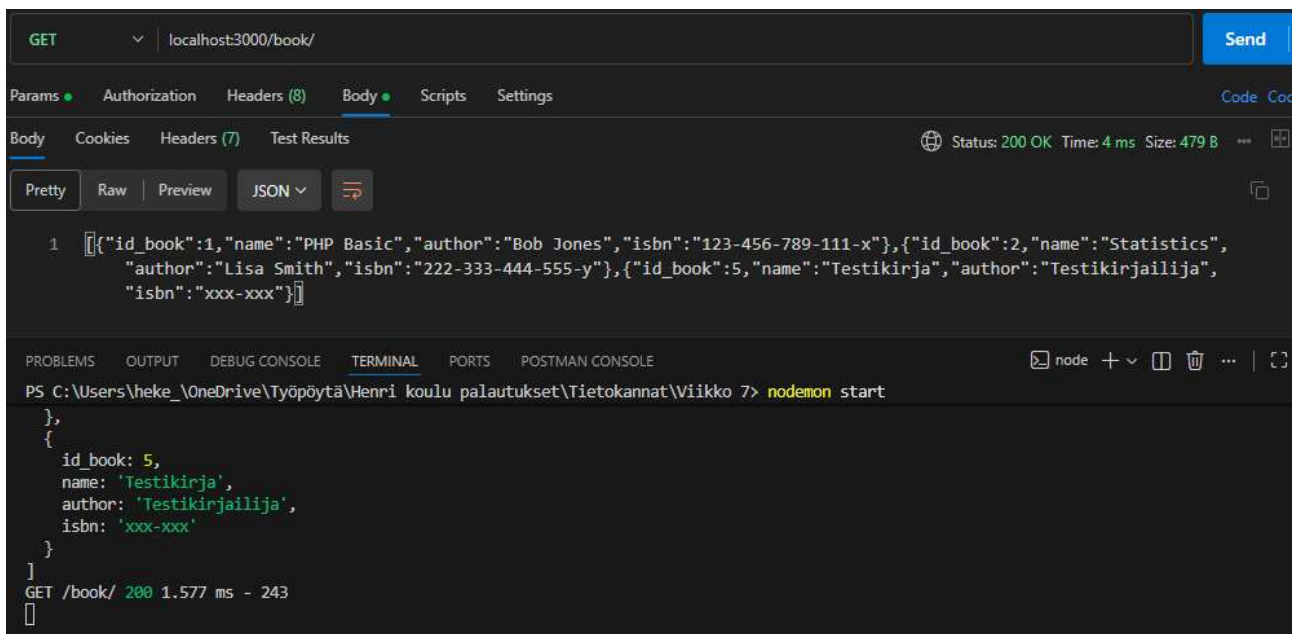


A screenshot of the Postman application showing a PUT request to `localhost:3000/book/5`. The request body is `x-www-form-urlencoded` with the following data:

Key	Value
name	Testikirja
author	Testikirjailija
isbn	xxx-xxx

The response is a JSON object: `{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "Rows matched: 1 Changed: 1 Warnings: 0", "serverStatus": 2, "warningStatus": 0, "changedRows": 1 }`. The status is 200 OK, Time: 6 ms, Size: 383 B.

Uusi tulostus:



A screenshot of the Postman application showing a GET request to `localhost:3000/book/`. The response is a JSON array of book objects:

```
[{"id_book": 1, "name": "PHP Basic", "author": "Bob Jones", "isbn": "123-456-789-111-x"}, {"id_book": 2, "name": "Statistics", "author": "Lisa Smith", "isbn": "222-333-444-555-y"}, {"id_book": 5, "name": "Testikirja", "author": "Testikirjailija", "isbn": "xxx-xxx"}]
```

Below the Postman interface, a terminal window shows the command `nodemon start` and the output of the GET request:

```
PS C:\Users\heke_OneDrive\Työpöytä\Henri koulu palautukset\Tietokannat\Viikko 7> nodemon start
},
{
  id_book: 5,
  name: 'Testikirja',
  author: 'Testikirjailija',
  isbn: 'xxx-xxx'
}
]
GET /book/ 200 1.577 ms - 243
```

Muokkaus toimi!

Viimeisenä poistaminen:

The screenshot shows a Postman interface with a DELETE request to `localhost:3000/book/5`. The response is a JSON object: `{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0, "changedRows": 0 }`. Below the response, the terminal shows the command `nodemon start` and the output of the application, which displays the book details for `id_book: 5`.

```
PS C:\Users\heke_OneDrive\Työpöytä\Henri koulu palautukset\Tietokannat\Viikko 7> nodemon start
},
{
  id_book: 5,
  name: 'Testikirja',
  author: 'Testikirjailija',
  isbn: 'xxx-xxx'
}
]
GET /book/ 200 1.577 ms - 243
DELETE /book/5 200 2.684 ms - 107
[]
```

Todetaan id_book: 5 tietueen poistuneen taulusta. Toimii!

The screenshot shows a Postman interface with a GET request to `localhost:3000/book/`. The response is a JSON array of two book objects: `[{"id_book": 1, "name": "PHP Basic", "author": "Bob Jones", "isbn": "123-456-789-111-x"}, {"id_book": 2, "name": "Statistics", "author": "Lisa Smith", "isbn": "222-333-444-555-y"}]`. Below the response, the terminal shows the command `nodemon start` and the output of the application, which displays the book details for `id_book: 1` and `id_book: 2`.

```
PS C:\Users\heke_OneDrive\Työpöytä\Henri koulu palautukset\Tietokannat\Viikko 7> nodemon start
{
  id_book: 1,
  name: 'PHP Basic',
  author: 'Bob Jones',
  isbn: '123-456-789-111-x'
},
{
  id_book: 2,
  name: 'Statistics',
  author: 'Lisa Smith',
  isbn: '222-333-444-555-y'
}
]
GET /book/ 200 1.373 ms - 165
[]
```

Henri Kemppainen TVT25SPL

Borrower-taulun komennot on toteutettu samaan tyyliin, mutta korvattu kenttien nimet

id_book → id_borrower,

name → fname,

author → lname,

isbn → streetaddress