

Tietokannat ja rajapinnat TH17

CRUD

Tässä esimerkki kuvaa Create Read Update ja Delete järjestyksessä.

Tietokantojen tekeminen

```
author VARCHAR(255)
isbn VARCHAR(20)' at line 4
mysql> show tables;
Empty set (0.00 sec)

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 (0.11 sec)

mysql> show tables;
+-----+
| Tables_in_books |
+-----+
| book             |
+-----+
1 row in set (0.00 sec)
```

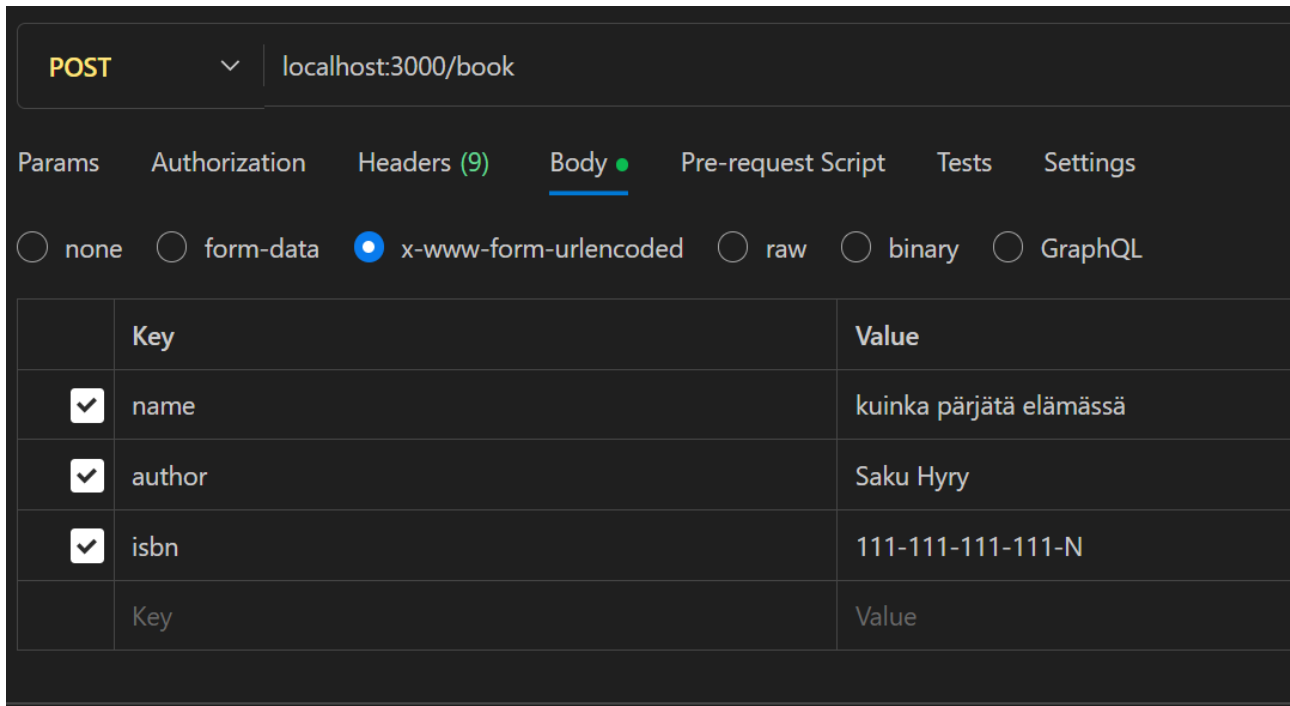
```
mysql> create table user
-> (
-> id_user INT primary key auto_increment,
-> username VARCHAR(20) UNIQUE,
-> password VARCHAR(255)
-> );
Query OK, 0 rows affected (0.09 sec)

mysql> show tables;
+-----+
| Tables_in_books |
+-----+
| book             |
| user             |
+-----+
2 rows in set (0.00 sec)
```

sitten ensimmäiseen asiaan eli CREATE(post)

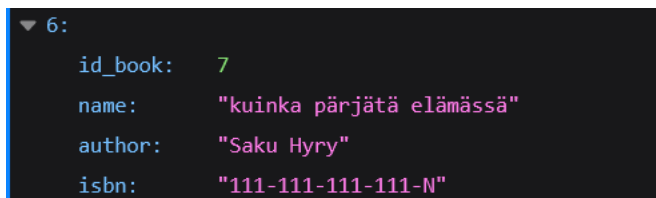
```
isbn: 030 030 003 111-N
▼ 2:
  id_book: 3
  name: null
  author: null
  isbn: null
▼ 3:
  id_book: 4
  name: null
  author: null
  isbn: null
▼ 4:
  id_book: 5
  name: null
  author: null
  isbn: null
▼ 5:
  id_book: 6
  name: null
  author: null
  isbn: null
▼ 6:
  id_book: 7
  name: "kuinka pärjätä elämässä"
  author: "Saku Hyry"
  isbn: "111-111-111-111-N"
```

Ylemmässä kuvassa tein vahingossa postia kun piti get:illä saada kuvaa, ja Bob Ross:ia lainaten ”there are no mistakes, just happy accidents”.



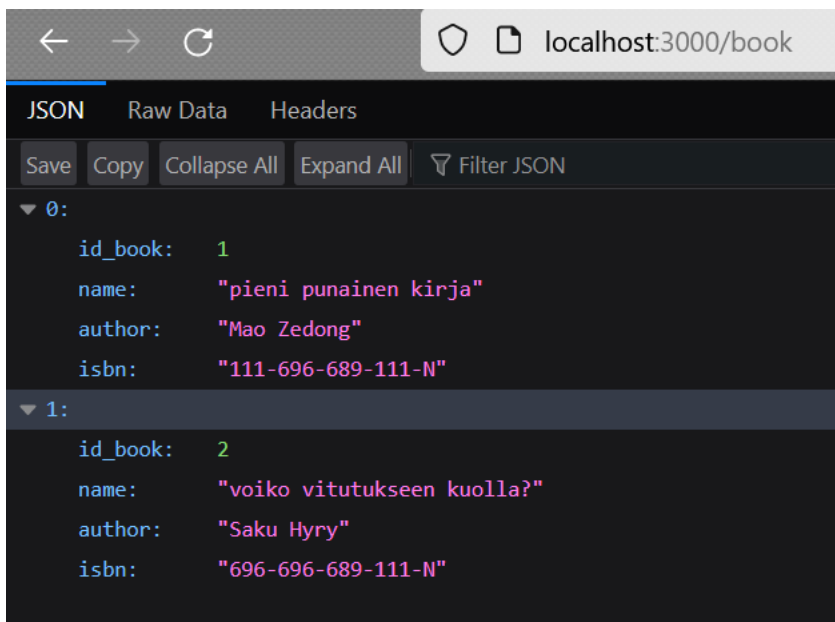
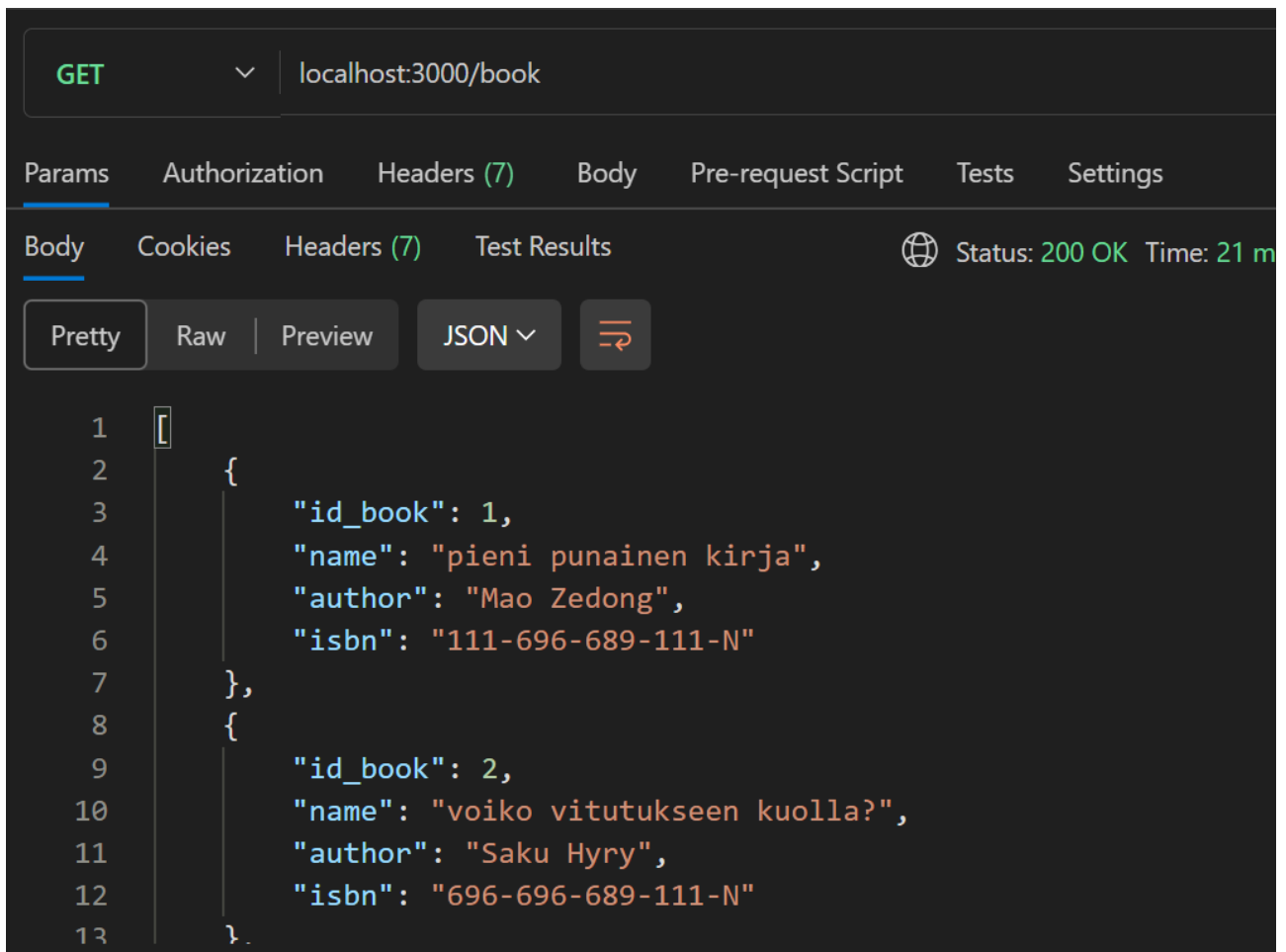
The screenshot shows a REST client interface with a POST request to `localhost:3000/book`. The 'Body' tab is selected, and the 'x-www-form-urlencoded' radio button is chosen. The form data is as follows:

	Key	Value
<input checked="" type="checkbox"/>	name	kuinka pärjätä elämässä
<input checked="" type="checkbox"/>	author	Saku Hyry
<input checked="" type="checkbox"/>	isbn	111-111-111-111-N
	Key	Value



```
▼ 6:
  id_book: 7
  name: "kuinka pärjätä elämässä"
  author: "Saku Hyry"
  isbn: "111-111-111-111-N"
```

Seuraava asia olisi READ(get)




Seuraava onkin UPDATE(put)


PUT localhost:3000/book/7

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

Key	Value
-----	-------

Body Cookies Headers (7) Test Results  Status: 200 OK Time: 8

Pretty Raw Preview JSON 

```

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

```

▼ 6:

```

id_book:    7
name:       "elämässä ei voi pärjätä"
author:     "Saku Hyry"
isbn:       "111-111-111-111-N"

```

```
mysql> select*from book;
```

id_book	name	author	isbn
1	pieni punainen kirja	Mao Zedong	111-696-689-111-N
2	voiko vitutukseen kuolla?	Saku Hyry	696-696-689-111-N
3	NULL	NULL	NULL
4	NULL	NULL	NULL
5	NULL	NULL	NULL
6	NULL	NULL	NULL
7	elämässä ei voi pärjätä	Saku Hyry	111-111-111-111-N

7 rows in set (0.00 sec)

Kuva tämän hetkisestä tilanteesta PUT:in jälkeen.

ja viimeisenä DELETE(delete)yllättikö :D?

The screenshot shows a REST client interface with a dark theme. At the top, a dropdown menu is set to 'DELETE' and the URL is 'localhost:3000/book/3'. Below this, there are tabs for 'Params', 'Authorization', 'Headers (9)', 'Body', and 'Pre-reqs'. The 'Params' tab is selected, showing a 'Query Params' section with a table that has one column labeled 'Key'. Below the tabs, there are buttons for 'Body', 'Cookies', 'Headers (7)', and 'Test Results'. The 'Body' tab is selected, showing a 'Pretty' button, a 'Raw' button, a 'Preview' button, a 'JSON' dropdown, and a refresh icon. The main area displays a JSON response with line numbers 1 through 9 on the left. The JSON is formatted as follows:

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

```
mysql> select*from book;
```

id_book	name	author	isbn
1	pieni punainen kirja	Mao Zedong	111-696-689-111-N
2	voiko vitutukseen kuolla?	Saku Hyry	696-696-689-111-N
3	NULL	NULL	NULL
4	NULL	NULL	NULL
5	NULL	NULL	NULL
6	NULL	NULL	NULL
7	elämässä ei voi pärjätä	Saku Hyry	111-111-111-111-N

```
7 rows in set (0.00 sec)
```



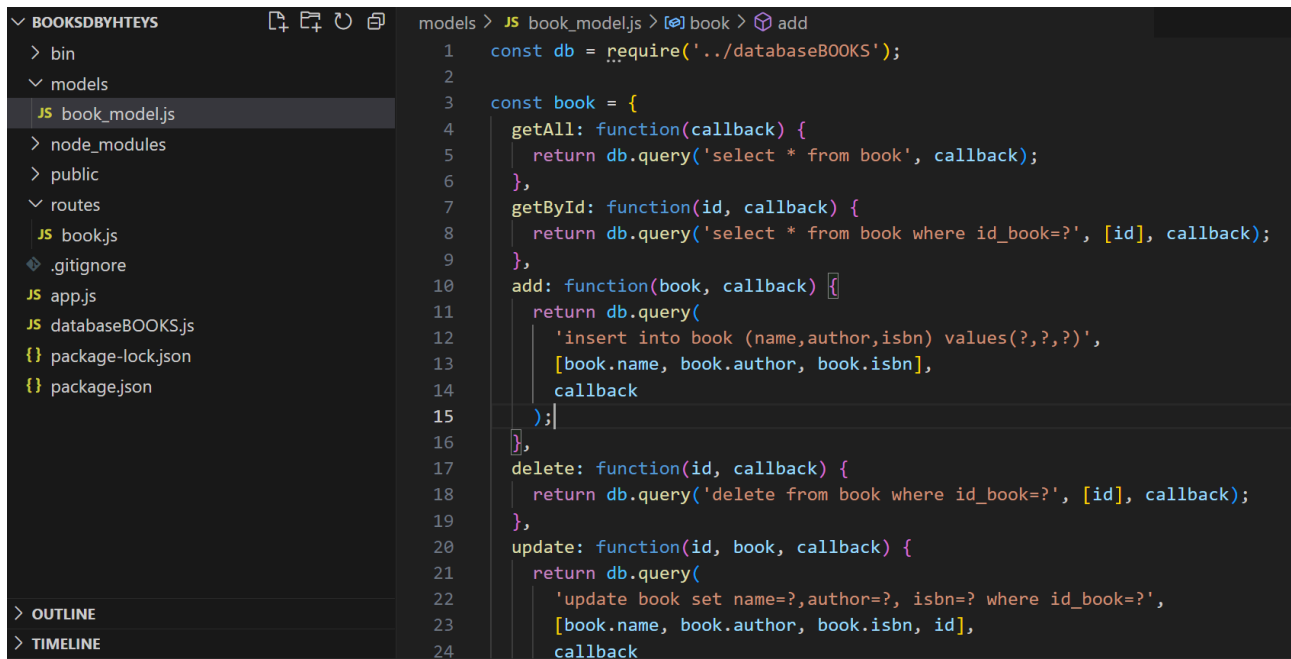
```
mysql> select*from book;
```

id_book	name	author	isbn
1	pieni punainen kirja	Mao Zedong	111-696-689-111-N
2	voiko vitutukseen kuolla?	Saku Hyry	696-696-689-111-N
4	NULL	NULL	NULL
5	NULL	NULL	NULL
6	NULL	NULL	NULL
7	elämässä ei voi pärjätä	Saku Hyry	111-111-111-111-N

```
6 rows in set (0.00 sec)
```

Tässä kuva ennen ja jälkeen deleten ja voipi huomata että tämä toimi.

tässä vielä kuva osasta koodista ja rakenteesta.



```
models > JS book_model.js > book > add
1  const db = require('../databaseBOOKS');
2
3  const book = {
4    getAll: function(callback) {
5      return db.query('select * from book', callback);
6    },
7    getById: function(id, callback) {
8      return db.query('select * from book where id_book=?', [id], callback);
9    },
10   add: function(book, callback) {
11     return db.query(
12       'insert into book (name,author,isbn) values(?,?,?)',
13       [book.name, book.author, book.isbn],
14       callback
15     );
16   },
17   delete: function(id, callback) {
18     return db.query('delete from book where id_book=?', [id], callback);
19   },
20   update: function(id, book, callback) {
21     return db.query(
22       'update book set name=?,author=?, isbn=? where id_book=?',
23       [book.name, book.author, book.isbn, id],
24       callback
25     );
26   }
27 }
```