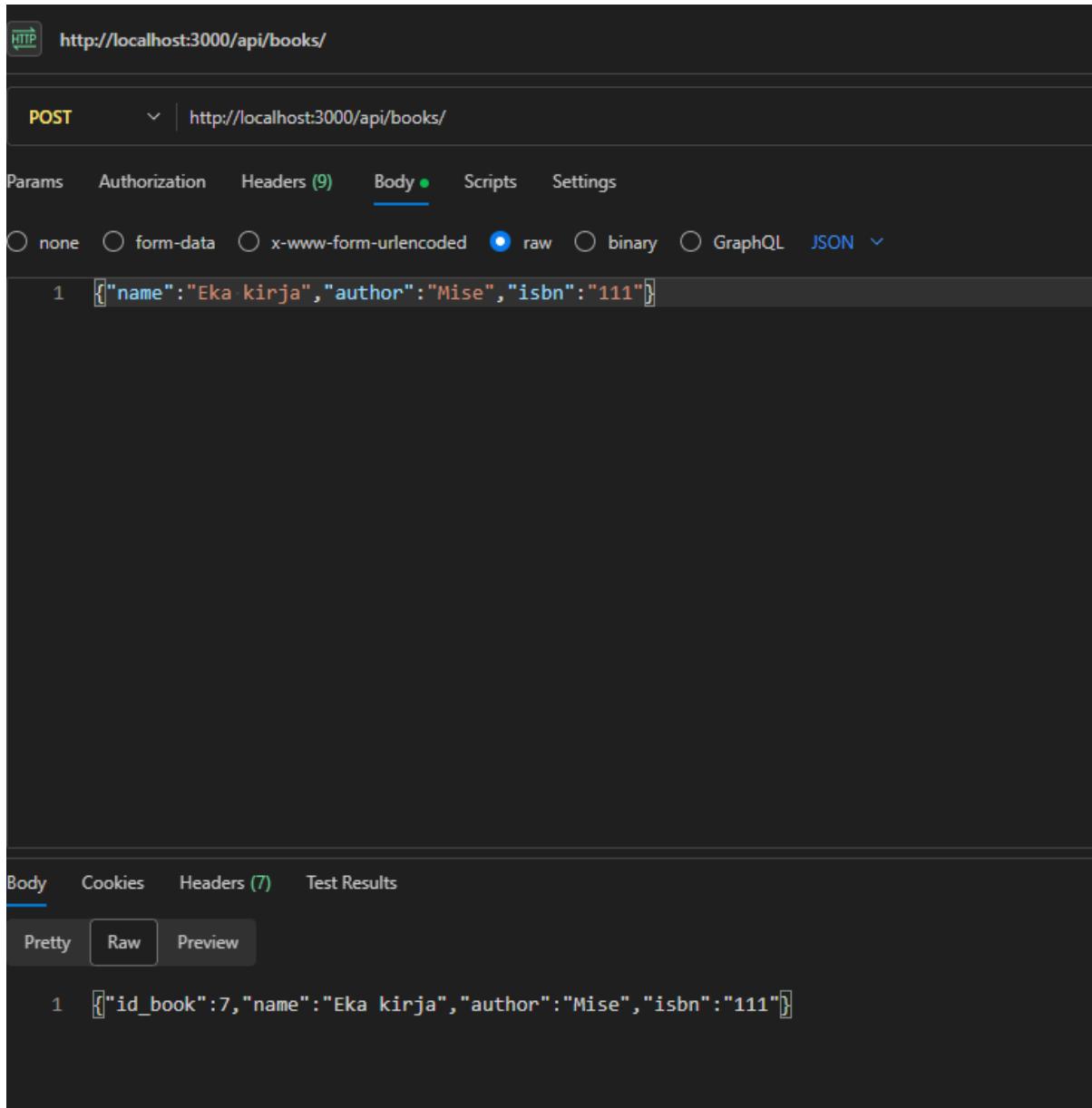


## 20: Tietokantayhteys ja CRUD Nodella ja Expressillä

### POST



The screenshot shows a POST request to `http://localhost:3000/api/books/`. The request method is POST, and the URL is `http://localhost:3000/api/books/`. The Body tab is selected, showing the raw JSON payload:

```
1  [{"name": "Eka kirja", "author": "Mise", "isbn": "111"}]
```

The response tab shows the returned JSON object:

```
1  {"id_book": 7, "name": "Eka kirja", "author": "Mise", "isbn": "111"}
```

Kirja lisättiin tietokantaan POST-pyynnöllä.  
Pyyntö lähetettiin JSON-muodossa ja palvelin palautti lisätyn kirjan id:n.

## GET

The screenshot shows the POSTMAN application interface. At the top, there's a header bar with the URL `http://localhost:3000/api/books/`. Below the header, the method is set to `GET`, and the URL is again displayed. The `Body` tab is selected, showing options for `none`, `form-data`, `x-www-form-urlencoded`, `raw`, `binary`, `GraphQL`, and `JSON`. The `JSON` option is currently selected. In the main body area, there is a single digit `1`. At the bottom, the response section shows the status as `200 OK`, time as `4 ms`, and size as `564 B`. The `Body` tab is selected, displaying the JSON response:

```
[{"id_book":1,"name":"Testikirja","author":"Tekijä","isbn":"123"}, {"id_book":2,"name":"Testikirja","author":"Tekijä","isbn":"123"}, {"id_book":3,"name":"Testikirja","author":"Tekijä","isbn":"123"}, {"id_book":4,"name":"Testikirja","author":"Tekijä","isbn":"123"}, {"id_book":7,"name":"Eka kirja","author":"Mise","isbn":"111"}]
```

GET-pyynnöllä haettiin kaikki tietokannassa olevat kirjat.  
Palvelin palauttaa kirjalistan JSON-muodossa.

## PUT / GET

The screenshot shows two requests made using the Postman application.

**PUT Request:**

- URL: `http://localhost:3000/api/books/7`
- Method: `PUT`
- Body (raw JSON):

```
1 {
2   "name": "Eka kirja (päivitetty)",
3   "author": "Mise",
4   "isbn": "222" //VAIHDETTU
5 }
```

**Response:**

- Status: 200 OK
- Time: 13 ms
- Size: 245 B
- Body (Pretty):

```
1 {"ok":true}
```

**GET Request:**

- URL: `http://localhost:3000/api/books`
- Method: `GET`
- Body (raw JSON):

```
1
```

**Response:**

- Status: 200 OK
- Time: 4 ms
- Size: 578 B
- Body (Pretty):

```
1 [{"id_book":1,"name":"Testikirja","author":"Tekij\u00e4","isbn":"123"}, {"id_book":2,"name":"Testikirja","author":"Tekij\u00e4","isbn":"123"}, {"id_book":3,"name":"Testikirja","author":"Tekij\u00e4","isbn":"123"}, {"id_book":4,"name":"Testikirja","author":"Tekij\u00e4","isbn":"123"}, {"id_book":7,"name":"Eka kirja (p\u00e4ivitetty)","author":"Mise","isbn":"222"}]
```

PUT- päivitys olemassa olevan id.

Päivityksen onnistuminen varmistettiin GET-pyynnöllä.

## DELETE / GET

The screenshot shows the Postman interface with a DELETE request to `http://localhost:3000/api/books/7`. The Body tab is selected, showing a raw JSON payload of `1 {"ok":true}`. The response tab shows the result of the delete operation.

Body

Pretty | Raw | Preview | **JSON** | `→`

1 [{"ok":true}]

The screenshot shows the Postman interface with a GET request to `http://localhost:3000/api/books`. The response tab shows the list of books returned by the GET request.

GET | http://localhost:3000/api/books | **Send**

Body

Pretty | Raw | Preview | `→`

1 [{"id\_book":1,"name":"Testikirja","author":"Tekijä","isbn":"123"}, {"id\_book":2,"name":"Testikirja","author":"Tekijä","isbn":"123"}, {"id\_book":3,"name":"Testikirja","author":"Tekijä","isbn":"123"}, {"id\_book":4,"name":"Testikirja","author":"Tekijä","isbn":"123"}]

DELETE-pyynnöllä poistettiin kirja tietokannasta id:n (7) perusteella.  
Poiston jälkeen kirja ei enää näkynyt GET-pyynnön tuloksissa.