

CRUD-komennot BOOK-tauluun

Luetaan GET:llä:

The screenshot shows the Postman interface with a GET request to `localhost:3000/book/` successfully executed. The response is a JSON array of three book objects.

Request Details:

- Method: GET
- URL: `localhost:3000/book/`
- Query Params: None

Response Details:

- Status: 200 OK
- Time: 62 ms
- Size: 498 B

Response Body (JSON):

```
1 {
2   {
3     "id_book": 1,
4     "name": "PHP Basic",
5     "author": "Bob Jones",
6     "isbn": "123-456-789-111-x"
7   },
8   {
9     "id_book": 2,
10    "name": "Statistics",
11    "author": "Lisa Smith",
12    "isbn": "222-333-444-555-y"
13  },
14  {
15    "id_book": 3,
16    "name": "C Programming Language",
17    "author": "Denis Ritchie",
18    "isbn": "123-456-443-xyz"
19  }
20 }
```

Start working with APIs

Next: Save requests. [Show me](#) 67%

Päivitetään PUT:lla:

The screenshot shows the Postman application interface. On the left sidebar, the 'Environments' tab is selected. The main workspace displays a PUT request to `localhost:3000/book/3`. The request body is configured with the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id_book	NULL	
<input checked="" type="checkbox"/> name	C Programming Language	
<input checked="" type="checkbox"/> author	Denis Ritchie	
<input checked="" type="checkbox"/> isbn	123-456-443-yyy	
Key	Value	Description

The response is displayed in the 'Body' tab, showing a JSON object with the following fields:

```
1  {
2    "fieldCount": 0,
3    "affectedRows": 1,
4    "insertId": 0,
5    "serverStatus": 2,
6    "warningCount": 0,
7    "message": "(Rows matched: 1 Changed: 1 Warnings: 0",
8    "protocol41": true,
9    "changedRows": 1
10 }
```

At the bottom left, a progress bar indicates 'Start working with APIs' at 67%, with the next step being 'Save requests.' and a 'Show me' link.

Uusi luku ja nähdään että tiedot isb:ssä muuttui:

The screenshot displays the Postman application interface. The left sidebar shows the 'My Workspace' section with 'Environments' selected. The main panel shows a GET request to 'localhost:3000/book/3' with the 'Body' tab selected. The request body is a JSON object with the following data:

KEY	VALUE	DESCRIPTION
id_book	NULL	
name	C Programming Language	
author	Denis Ritchie	
isbn	123-456-443-yyy	

The response body is shown in the 'Body' tab, displaying a JSON object with the following data:

```
{  "id_book": 3,  "name": "C Programming Language",  "author": "Denis Ritchie",  "isbn": "123-456-443-yyy"}
```

The status bar at the bottom indicates a 200 OK response with a 13 ms response time and 332 B of data. The bottom status bar also shows 'Start working with APIs' with a progress bar at 67% and 'Next: Save requests.' with a 'Show me' link.

Poistetaan kirja book_id 3:

The screenshot shows the Postman interface with a DELETE request to `localhost:3000/book/3`. The request is configured with the following parameters:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id_book	NULL	
<input checked="" type="checkbox"/> name	C Programming Language	
<input checked="" type="checkbox"/> author	Denis Ritchie	
<input checked="" type="checkbox"/> isbn	123-456-443-yyy	
Key	Value	Description

The response is a JSON object:

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "serverStatus": 2,
6   "warningCount": 0,
7   "message": "",
8   "protocol41": true,
9   "changedRows": 0
10 }
```

At the bottom, there is a progress bar for "Start working with APIs" at 67% and a "Show me" link.

Uusi luku näyttää, että kirjaa ei enää ole:

The screenshot shows the Postman API client interface. The left sidebar contains navigation options: Collections, APIs, Environments (selected), Mock Servers, Monitors, Flows, and History. The main workspace displays a GET request to `localhost:3000/book/` with the `x-www-form-urlencoded` body type. The request is saved and has a status of `200 OK` with a response time of `11 ms` and a size of `401 B`. The response body is shown in JSON format, containing an array of two book objects.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> <code>id_book</code>	<code>NULL</code>	
<input checked="" type="checkbox"/> <code>name</code>	<code>C Programming Language</code>	
<input checked="" type="checkbox"/> <code>author</code>	<code>Denis Ritchie</code>	
<input checked="" type="checkbox"/> <code>isbn</code>	<code>123-456-443-yyy</code>	
Key	Value	Description

```
1 {
2   {
3     "id_book": 1,
4     "name": "PHP Basic",
5     "author": "Bob Jones",
6     "isbn": "123-456-789-111-x"
7   },
8   {
9     "id_book": 2,
10    "name": "Statistics",
11    "author": "Lisa Smith",
12    "isbn": "222-333-444-555-y"
13  }
14 }
```

Start working with APIs
67%
Next: Save requests. [Show me](#)

Lisätään kirja uudelleen POST:lla:

The screenshot displays the Postman application interface. The top navigation bar includes links for Home, Workspaces, API Network, and Explore, along with a search bar and utility buttons like 'Invite' and 'Upgrade'. The left sidebar shows the 'My Workspace' section with a tree view containing Collections, Globals, APIs, Environments (highlighted), Mock Servers, Monitors, Flows, and History.

The main workspace area is configured for a POST request to the endpoint `localhost:3000/book/`. The request body is set to 'x-www-form-urlencoded' and contains the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id_book	NULL	
<input checked="" type="checkbox"/> name	C Programming Language	
<input checked="" type="checkbox"/> author	Denis Ritchie	
<input checked="" type="checkbox"/> isbn	123-456-443-yyy	
Key	Value	Description

The response section at the bottom shows the 'Body' tab with a JSON response in 'Pretty' format:

```
1 {
2   "id_book": "NULL",
3   "name": "C Programming Language",
4   "author": "Denis Ritchie",
5   "isbn": "123-456-443-yyy"
6 }
```

At the bottom left, a progress bar indicates 'Start working with APIs' at 67%, with the next step being 'Save requests.' and a 'Show me' link. The bottom status bar includes icons for Online, Find and Replace, Console, Cookies, Capture requests, Runner, Trash, and other utility icons.

Luetaan tiedot GET:llä:

The screenshot displays the Postman API client interface. The left sidebar shows the 'My Workspace' section with a collection named 'Rp2'. The main panel shows a GET request to 'localhost:3000/book/'. The request is configured with the 'Body' tab selected, showing a JSON response. The response is a list of three book objects, each with 'id_book', 'name', 'author', and 'isbn' fields.

Request Details:

- Method: GET
- URL: localhost:3000/book/
- Body Type: x-www-form-urlencoded

Response Details:

- Status: 200 OK
- Time: 11 ms
- Size: 498 B

Response Body (JSON):

```
1 {
2   {
3     "id_book": 1,
4     "name": "PHP Basic",
5     "author": "Bob Jones",
6     "isbn": "123-456-789-111-x"
7   },
8   {
9     "id_book": 2,
10    "name": "Statistics",
11    "author": "Lisa Smith",
12    "isbn": "222-333-444-555-y"
13  },
14  {
15    "id_book": 4,
16    "name": "C Programming Language",
17    "author": "Denis Ritchie",
18    "isbn": "123-456-443-yyy"
19  }
20 }
```

BORROWER-taulun CRUD-komennot

GET:

The screenshot shows the Postman interface with a GET request to `localhost:3000/borrower` using the `x-www-form-urlencoded` content type. The response is a JSON array of two objects, displayed in the 'Body' tab.

Request Details:

- Method: GET
- URL: `localhost:3000/borrower`
- Content Type: `x-www-form-urlencoded`

Response Body (JSON):

```
1 {
2   {
3     "id_borrower": 1,
4     "fname": "Aku",
5     "lname": "Ankka",
6     "streetAddress": "Parasitiitie 21"
7   },
8   {
9     "id_borrower": 2,
10    "fname": "Hannu",
11    "lname": "Hanhi",
12    "streetAddress": "Onnelantie 10"
13  }
14 }
```

Status: 200 OK, 17 ms, 401 B

POST:

My Workspace

NewImport

OverviewOverview

POST localhost:3000

Rp2GET {{base_url}}

+

...

Rp2

</>

Collections

Globals

Rp2

✓

Environments

Mock Servers

Monitors

Flows

History

Start working with APIs

67%

Next: Save requests.

Show me

localhost:3000/borrower

Save

</>

POSTlocalhost:3000/borrower

Send

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookies

● none● form-data● x-www-form-urlencoded● raw● binary● GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> id_borrower	NULL			
<input checked="" type="checkbox"/> fname	Roope			
<input checked="" type="checkbox"/> lname	Ankka			
<input checked="" type="checkbox"/> streetAddress	kultakaivos 30			
Key	Value	Description		

BodyCookiesHeaders (7)Test Results

200 OK13 ms322 BSave Response

PrettyRawPreviewVisualizeJSON

```
1  {
2    "id_borrower": "NULL",
3    "fname": "Roope",
4    "lname": "Ankka",
5    "streetAddress": "kultakaivos 30"
6  }
```

OnlineFind and ReplaceConsole

CookiesCapture requestsRunnerTrash

PUT:

My Workspace

NewImport

OverviewOverviewPUT localhost:3000/borrower/3GET {{base_url}}Rp2

Collections

APIs

Environments

Mock Servers

Monitors

Flows

History

Start working with APIs

67%

Next: Save requests.

Show me

localhost:3000/borrower/3

Save

Send

PUTlocalhost:3000/borrower/3

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQL

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> id_borrower	NULL		
<input checked="" type="checkbox"/> fname	Roope		
<input checked="" type="checkbox"/> lname	Ankka		
<input checked="" type="checkbox"/> streetAddress	Ankkalinn 7		
Key	Value	Description	

BodyCookiesHeaders (7)Test Results

200 OK23 ms404 BSave Response

PrettyRawPreviewVisualizeJSON

```
1  {
2    "fieldCount": 0,
3    "affectedRows": 1,
4    "insertId": 0,
5    "serverStatus": 2,
6    "warningCount": 0,
7    "message": "(Rows matched: 1 Changed: 1 Warnings: 0",
8    "protocol41": true,
9    "changedRows": 1
10 }
```

OnlineFind and ReplaceConsole

CookiesCapture requestsRunnerTrash

DELETE:

My Workspace

NewImport

OverviewOverview

DEL localhost:3000

Rp2GET {{base_url}}

Rp2

Collections

APIs

Environments

Mock Servers

Monitors

Flows

History

New Collection

New Collection

localhost:3000/borrower/3

Save

Send

DELETElocalhost:3000/borrower/3

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQL

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> id_borrower	NULL		
<input checked="" type="checkbox"/> fname	Roope		
<input checked="" type="checkbox"/> lname	Ankka		
<input checked="" type="checkbox"/> streetAddress	Ankkalinn 7		
Key	Value	Description	

BodyCookiesHeaders (7)Test Results

200 OK19 ms363 BSave Response

PrettyRawPreviewVisualizeJSON

```
1  {
2    "fieldCount": 0,
3    "affectedRows": 1,
4    "insertId": 0,
5    "serverStatus": 2,
6    "warningCount": 0,
7    "message": "",
8    "protocol41": true,
9    "changedRows": 0
10 }
```

Start working with APIs

67%

Next: Save requests.

Show me

OnlineFind and ReplaceConsole

CookiesCapture requestsRunnerTrash

Autetikointi Basic Auth-menetelmällä

Lisätään käyttäjä user_tablesiin:

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The left sidebar contains a 'Collections' panel with 'New Collection' and 'New Collection' buttons. The main area displays a POST request to 'localhost:3000/user'. The request body is set to 'x-www-form-urlencoded' and contains the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id_user	NULL	
<input checked="" type="checkbox"/> username	Aku	
<input checked="" type="checkbox"/> password	3333	
Key	Value	Description

The response is shown in the 'Body' tab, indicating a status of 200 OK, time of 158 ms, and size of 288 B. The response body is a JSON object:

```
1 {
2   "id_user": "NULL",
3   "username": "Aku",
4   "password": "3333"
5 }
```

At the bottom, there is a progress bar for 'Start working with APIs' at 67% and a 'Next: Save requests.' button.

Suojataan kaikki taulut:

```
1  var express = require('express');
2  var path = require('path');
3  var cookieParser = require('cookie-parser');
4  var logger = require('morgan');
5  const db = require('./database');
6  const bcrypt = require('bcryptjs');
7  const basicAuth = require('express-basic-auth');
8
9  var bookRouter = require('./routes/book');
10 var borrowerRouter = require('./routes/borrower');
11 var userRouter = require('./routes/user');
12
13 var app = express();
14
15 app.use(basicAuth({users: { 'Aku': '3333' }}})
16 app.use(logger('dev'));
17 app.use(express.json());
18 app.use(express.urlencoded({ extended: false }));
19 app.use(cookieParser());
20 app.use(express.static(path.join(__dirname, 'public')));
21
22
23 app.use('/book', bookRouter);
24 app.use('/borrower', borrowerRouter);
25 app.use('/user', userRouter);
26
27
28 module.exports = app;
29
```

Testataan suojaksen toiminta GET:llä:

The screenshot shows the Postman application interface. The top bar includes navigation links (Home, Workspaces, API Network, Explore), a search bar, and utility buttons (Invite, Upgrade). The left sidebar displays the 'My Workspace' section with a 'Collections' list containing 'New Collection' and 'New Collection'. The main workspace area shows a GET request to 'localhost:3000/book'. The 'Authorization' tab is selected, showing 'Basic Auth' with a warning message: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables'. The 'Username' field is filled with 'Aku' and the 'Password' field with '3333'. The 'Show Password' checkbox is checked. Below the authorization section, the 'Body' tab is selected, showing a JSON response in 'Pretty' format. The response is an array of three book objects. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 63 ms', and 'Size: 498 B'. A 'Save Response' button is visible. At the bottom left, a 'Start working with APIs' progress bar shows 67% completion.

localhost:3000/book

GET localhost:3000/book

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Username Aku

Password 3333

☒ Show Password

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 63 ms Size: 498 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id_book": 1,
4     "name": "PHP Basic",
5     "author": "Bob Jones",
6     "isbn": "123-456-789-111-x"
7   },
8   {
9     "id_book": 2,
10    "name": "Statistics",
11    "author": "Lisa Smith",
12    "isbn": "222-333-444-555-y"
13  },
14  {
15    "id_book": 4,
16    "name": "C Programming Language",
17    "author": "Denis Ritchie",
18    "isbn": "123-456-443-yyy"
19  }
20 ]
```

Start working with APIs 67%

Next: Save requests. [Show me](#)

Online Find and Replace Console Cookies Capture requests Runner Trash

Ei toimi väärillä tiedoilla:

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore'. The left sidebar shows 'My Workspace' with options for 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The main panel displays a request to 'localhost:3000/book' using the 'GET' method. The 'Authorization' tab is selected, showing 'Basic Auth' with a warning: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables'. The 'Username' field contains 'Aku' and the 'Password' field contains '4444'. The 'Show Password' checkbox is checked. The 'Body' tab is also visible, showing a status of '401 Unauthorized' with a message 'Status: 401 Unauthorized Time: 8 ms Size: 236 B'. The bottom status bar shows 'Online', 'Find and Replace', 'Console', 'Cookies', 'Capture requests', 'Runner', 'Trash', and other utility icons.

My Workspace **New** **Import** Overview Overview GET localhost:3000/book Rp2 GET {{base_url}} + ... Rp2

Collections + **New Collection** **New Collection**

APIs

Environments

Mock Servers

Monitors

Flows

History

localhost:3000/book

GET localhost:3000/book **Send**

Params **Authorization** Headers (10) Body Pre-request Script Tests Settings **Cookies**

Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)

Username Aku

Password 4444

☒ Show Password

Body Cookies Headers (7) Test Results Status: 401 Unauthorized Time: 8 ms Size: 236 B **Save Response**

Pretty Raw Preview Visualize HTML

1

Start working with APIs 67% Next: Save requests. [Show me](#)

Online Find and Replace Console Cookies Capture requests Runner Trash