

w16-p1: create book_xx in postgresQL server, enter two data

The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree is expanded to show the 'public' schema, where the 'book_68' table is highlighted. The main pane displays a query: `SELECT * FROM public.book_68 ORDER BY id ASC`. Below the query, the 'Data output' tab shows the results of the query in a table format.

id	name	author
1	BookName_1	AAAB
2	Bookname_2	BbbA

w16-p2: get all data from table book_xx

The screenshot shows a web application running on localhost:3000/book_68. The page displays a list of books with the following data:

Id	Name	Author	Price	Action
2	Bookname_2	BbbA	100	Edit Delete
1	BookName_1	AAAB	100	Edit Delete

The background shows a code editor with the following JavaScript code in `book_68.js`:

```
const express = require('express');
const db = require('../utils/database');

router.get('/', async (req, res) => {
  try {
    const results = await db.query('SELECT * FROM book_68');
    console.log('results', JSON.stringify(results.rows));
    res.render('book_68/index', {
      data: results.rows,
      name: 'Hongmou Chang',
    });
  } catch (error) {
    console.log('error', error);
  }
});
```

The terminal output shows the following commands and responses:

```
GET /book_68 304 4.218 ms --
results [{"id":2,"name":"Bookname_2","author":"BbbA"}, {"id":1,"name":"BookName_1","author":"AAAB"}]
GET /book_68 304 3.632 ms --
results [{"id":2,"name":"Bookname_2","author":"BbbA"}, {"id":1,"name":"BookName_1","author":"AAAB"}]
GET /book_68 304 4.746 ms --
results [{"id":2,"name":"Bookname_2","author":"BbbA"}, {"id":1,"name":"BookName_1","author":"AAAB"}]
GET /book_68 304 2.982 ms --
```

w16-p3: /book_xx/create to book_xx/add_xx.ejs

The screenshot shows a VS Code editor on the left and a web browser on the right. In VS Code, the `book_68.js` file is open, showing a route for `/create` that renders the `add_68.ejs` template with a pre-filled ID of `409410668` and name `Hongmou Chang`. The terminal shows a successful GET request to `/book_68/create`. The browser on the right shows the rendered page at `localhost:3000/book_68/create`, displaying the ID and name in a header, followed by input fields for Name and Author, and an 'Add' button.

```
1 var express = require('express');
2 var router = express.Router();
3
4 const db = require('../utils/database');
5
6 /* READ */
7 router.get('/', async (req, res) => {
8   // ...
9 });
10
11 // CREATE
12 router.get('/create', (req, res) => {
13   res.render('book_68/add_68', {
14     id: '409410668',
15     name: 'Hongmou Chang',
16   });
17 });
18
19 module.exports = router;
```

GET /book_68/create 404 3.429 ms - 1886

results [{"id":1,"name":"Bookname_1","author":"AAB","price":111}, {"id":2,"name":"Bookname_2","author":"Bbba","price":222}]

GET /book_68 200 114.979 ms - 2715

GET /book_68/create 304 3.079 ms - -

GET /book_68/create 304 1.548 ms - -

w16-p4: POST /book_xx/create to create a book data

The screenshot shows a VS Code editor with two files open: `add_68.ejs` and `book_68.js`. In `add_68.ejs`, the `form` action is set to `/book_68/create` with the method `post`. In `book_68.js`, the `POST` route for `/create` is defined, which logs the request body, inserts the data into the database, and redirects to `/book_68`. The terminal shows a successful POST request to `/book_68/create` and the resulting database entries.

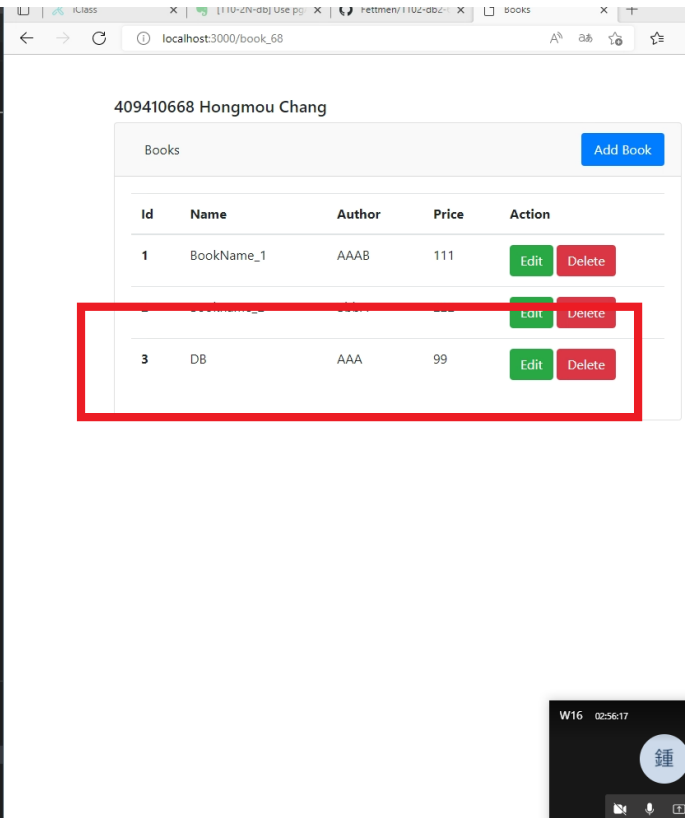
```
1 <doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6   <title>Add Book</title>
7   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
8 </head>
9 <body>
10   <div class="container" style="margin-top: 50px;">
11     <h2><%= id %> <%= name %></h2>
12     <div class="card">
13       <div class="card-header">
14         Add Book
15       </div>
16       <div class="card-body">
17         <form action="/book_68/create" method="post">
18           <div class="form-group">
19             <input type="text" value="<%= id %>" />
20           </div>
21           <div class="form-group">
22             <input type="text" value="<%= name %>" />
23           </div>
24           <div class="form-group">
25             <input type="text" value="<%= author %>" />
26           </div>
27           <div class="form-group">
28             <input type="text" value="<%= price %>" />
29           </div>
30           <div class="form-group">
31             <input type="submit" class="btn btn-info" value="Add" />
32           </div>
33         </form>
34       </div>
35     </div>
36   </div>
37 </body>
38 </html>
```

```
1 var express = require('express');
2 var router = express.Router();
3
4 const db = require('../utils/database');
5
6 //CREATE
7 router.post('/create', async (req, res) => {
8   console.log('body', req.body);
9
10   const name = req.body.name;
11   const author = req.body.author;
12   const price = req.body.price;
13
14   try {
15     const query = {
16       text: 'INSERT INTO book_68 (id, name, author, price) VALUES ($1,$2,$3,$4)',
17       values: [id, name, author, price],
18     };
19     await db.query(query);
20     res.redirect('/book_68');
21   } catch (error) {
22     console.log(error);
23   }
24 });
25
26 //READ
27 router.get('/', async (req, res) => {
28   // ...
29 });
30
31 router.get('/create', (req, res) => {
32   // ...
33 });
34
35 module.exports = router;
```

POST /book_68/create 302 131.531 ms - 60

results [{"id":1,"name":"Bookname_1","author":"AAB","price":111}, {"id":2,"name":"Bookname_2","author":"Bbba","price":222}, {"id":3,"name":"DB","author":"AAA","price":99}]

GET /book_68 200 13.174 ms - 3348



w16-LAST-log

```
a2fa5ff 409410668 Tue Jun 7 18:05:16 2022 +0800 ### w16-p4: POST /book_xx/create to create a book data
3934b53 409410668 Tue Jun 7 17:16:19 2022 +0800 ### w16-p2: get all data from table book_xx
e29d48a 409410668 Tue Jun 7 16:24:44 2022 +0800 ### w16-p2: get all data from table book_xx
e11db61 409410668 Tue Jun 7 15:50:22 2022 +0800 ### w16-p1: create book_xx in postgresSQL server, enter tv
```

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2022-06-06"
a2fa5ff 409410668 Tue Jun 7 18:05:16 2022 +0800 ### w16-p4: POST /book_xx/create to create a book data
3934b53 409410668 Tue Jun 7 17:16:19 2022 +0800 ### w16-p2: get all data from table book_xx
e29d48a 409410668 Tue Jun 7 16:24:44 2022 +0800 ### w16-p2: get all data from table book_xx
e11db61 409410668 Tue Jun 7 15:50:22 2022 +0800 ### w16-p1: create book_xx in postgresSQL server, enter two data
```

git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2022-06-06"