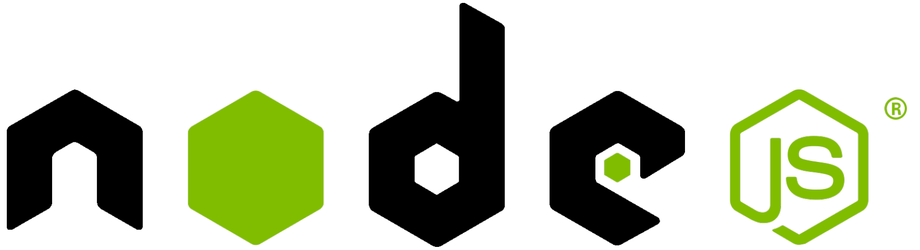
JAVASCRIPT

CSS-FRONT-END BACK-END
(API WITH PGSQL & MONGODB)

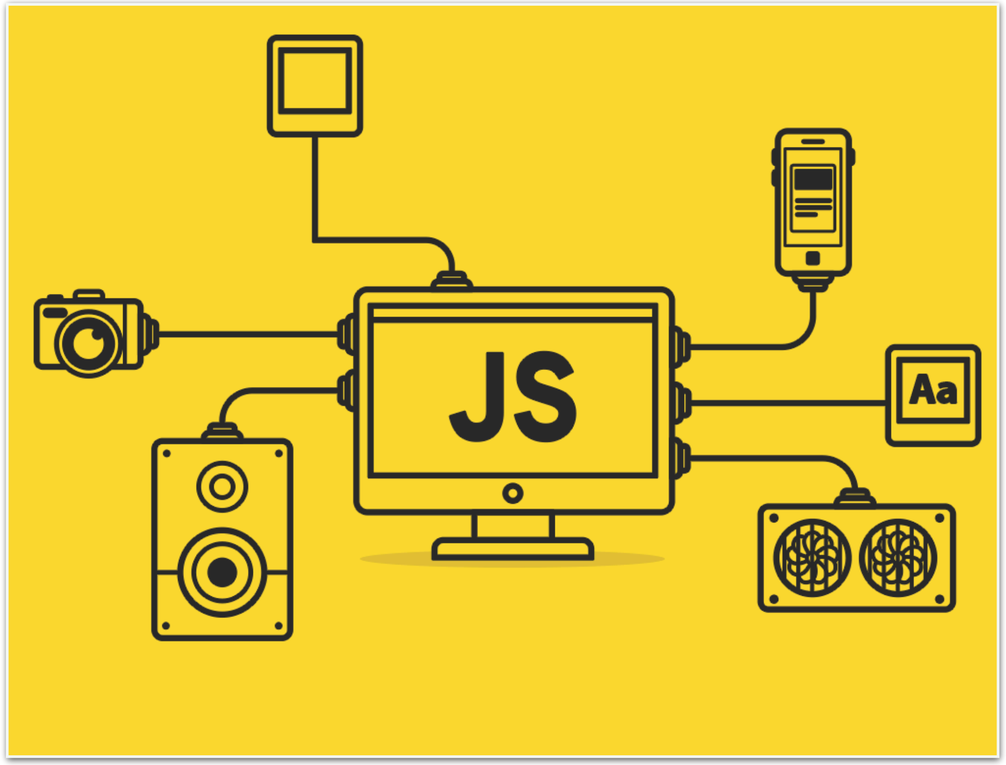
Anirach Mingkhwan

Anirach.m@fitm.kmutnb.ac.th



OUTLINE

- CSS-Front-End
- PostgreSQL
- MongoDB
- Postman Test
- Front-End connect with Back-End



CSS FRONT-END


HTML
@shippingcontainerworld
livinginacontainer.com
CSS
Zillow

Books

Title	Author	Actions
Runway	planB	View Edit Delete
Ghost	John.D	View Edit Delete
Dr. Who	Kong	View Edit Delete
Firework	Dobb	View Edit Delete

Create New Book

Books

TITLE	AUTHOR	ACTIONS
The Simpsons	Bart Simson	View Edit Delete
7000 Stars	Anirach	View Edit Delete
Water Mark	Jimmy	View Edit Delete
Dr. Who	Robot	View Edit Delete
Runway	planB	View Edit Delete

Create New Book

Update Book

Title:

The Simpsons

Author:

Bart Simson

Update

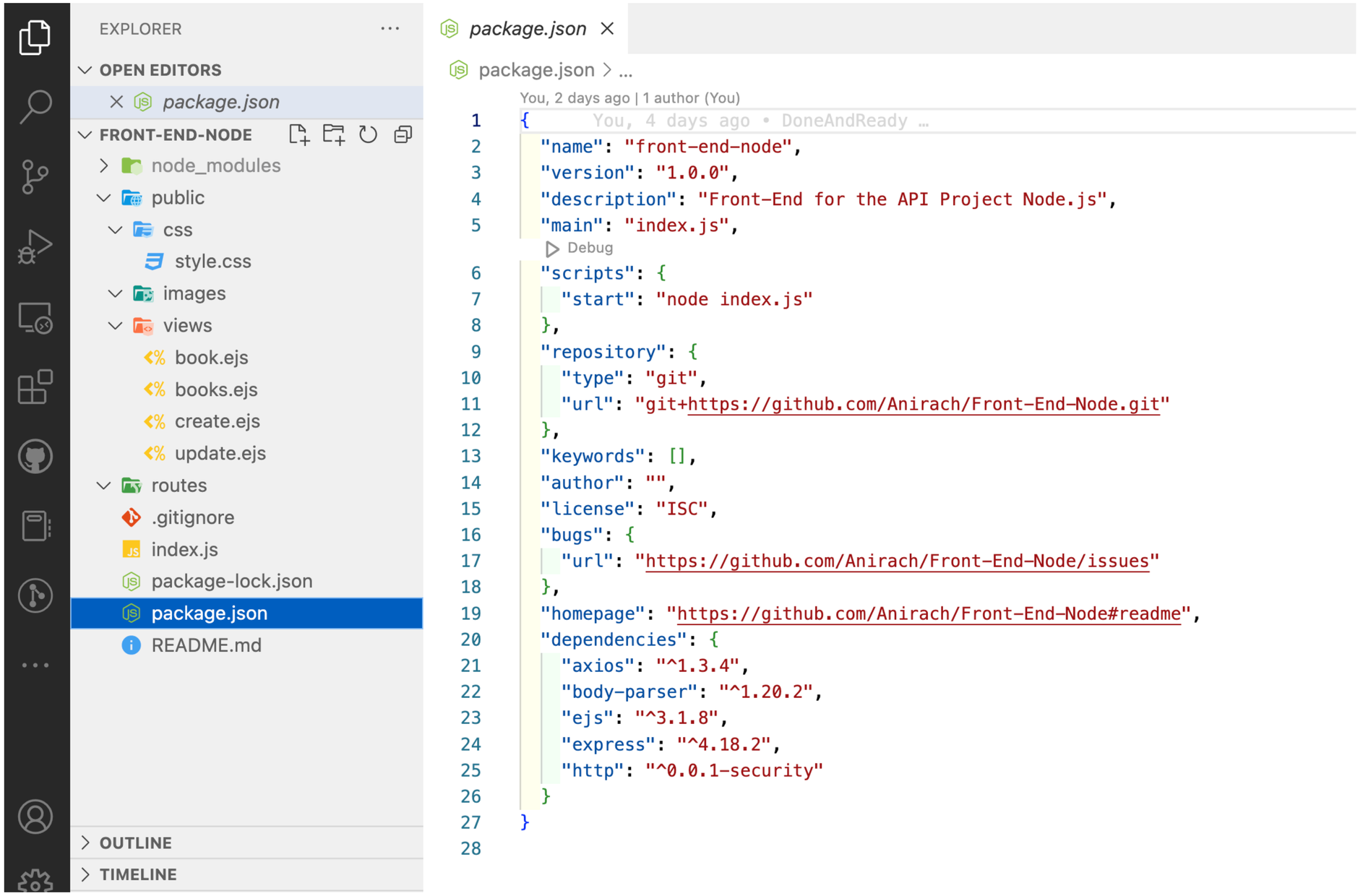
View Delete Back to List

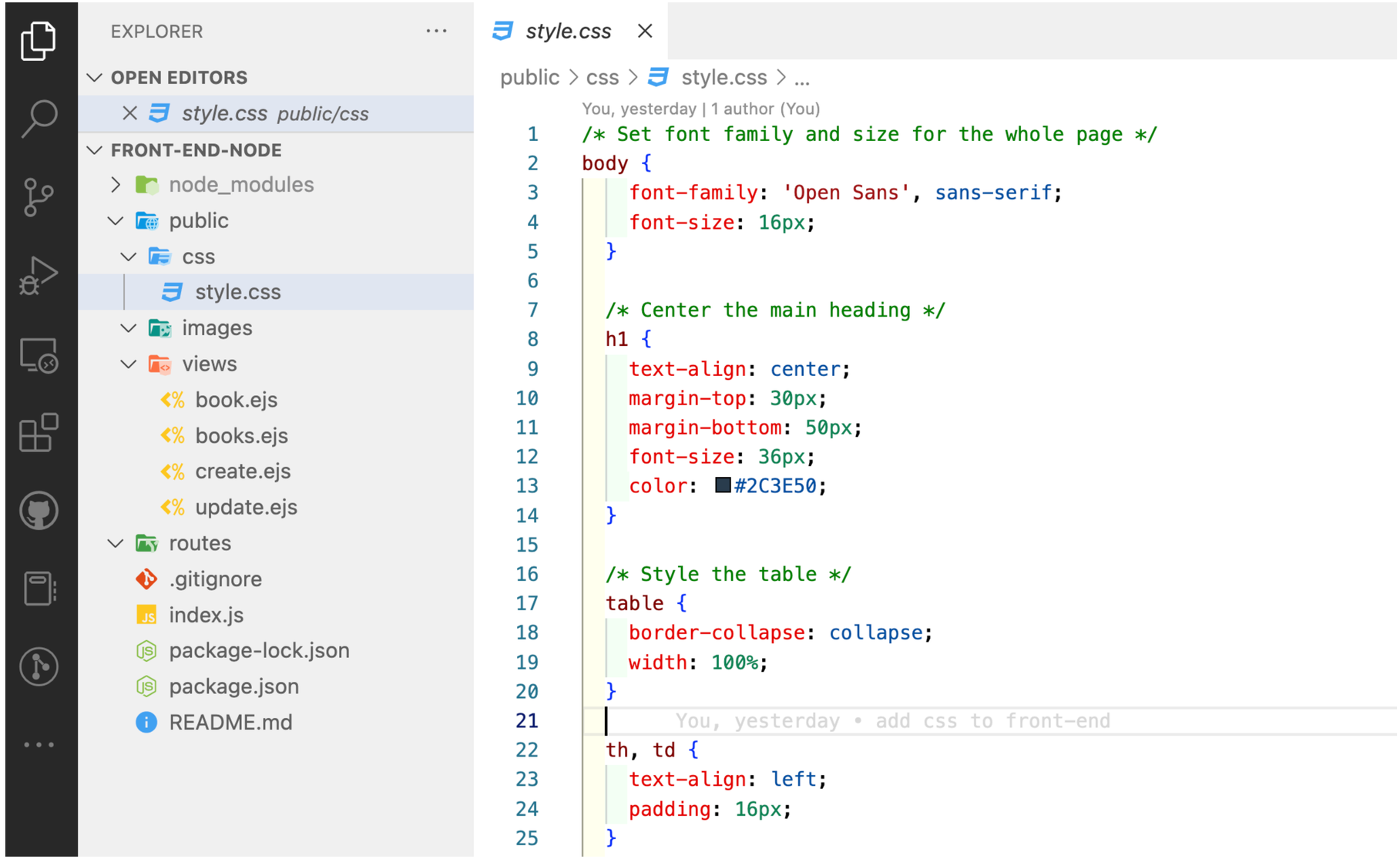
Directory Structure

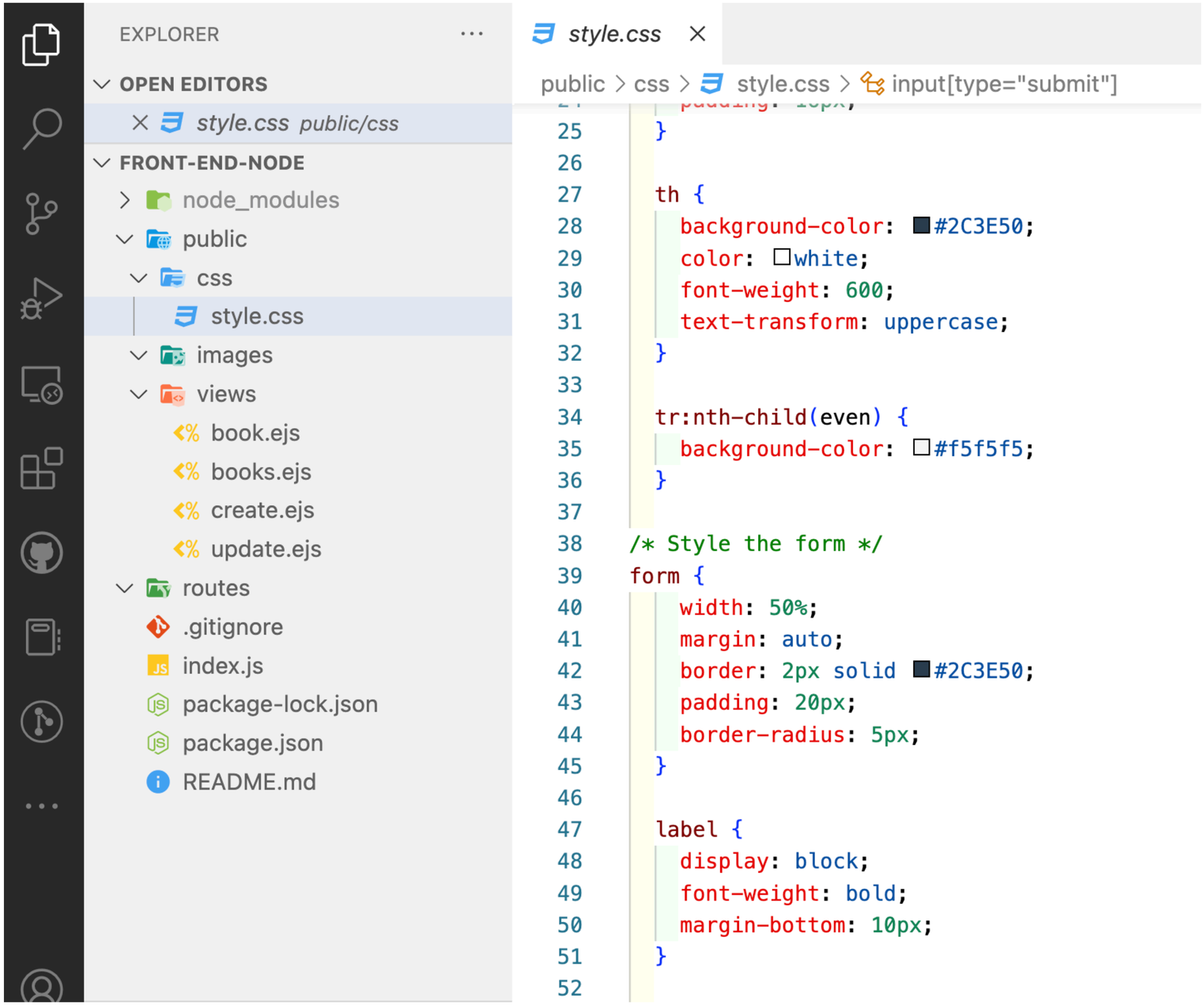
```
app/
-- views/
---- home.ejs
-- public/
---- css/
------ style.css
---- js/
------ script.js
-- node_modules/
-- package.json
-- server.js
```

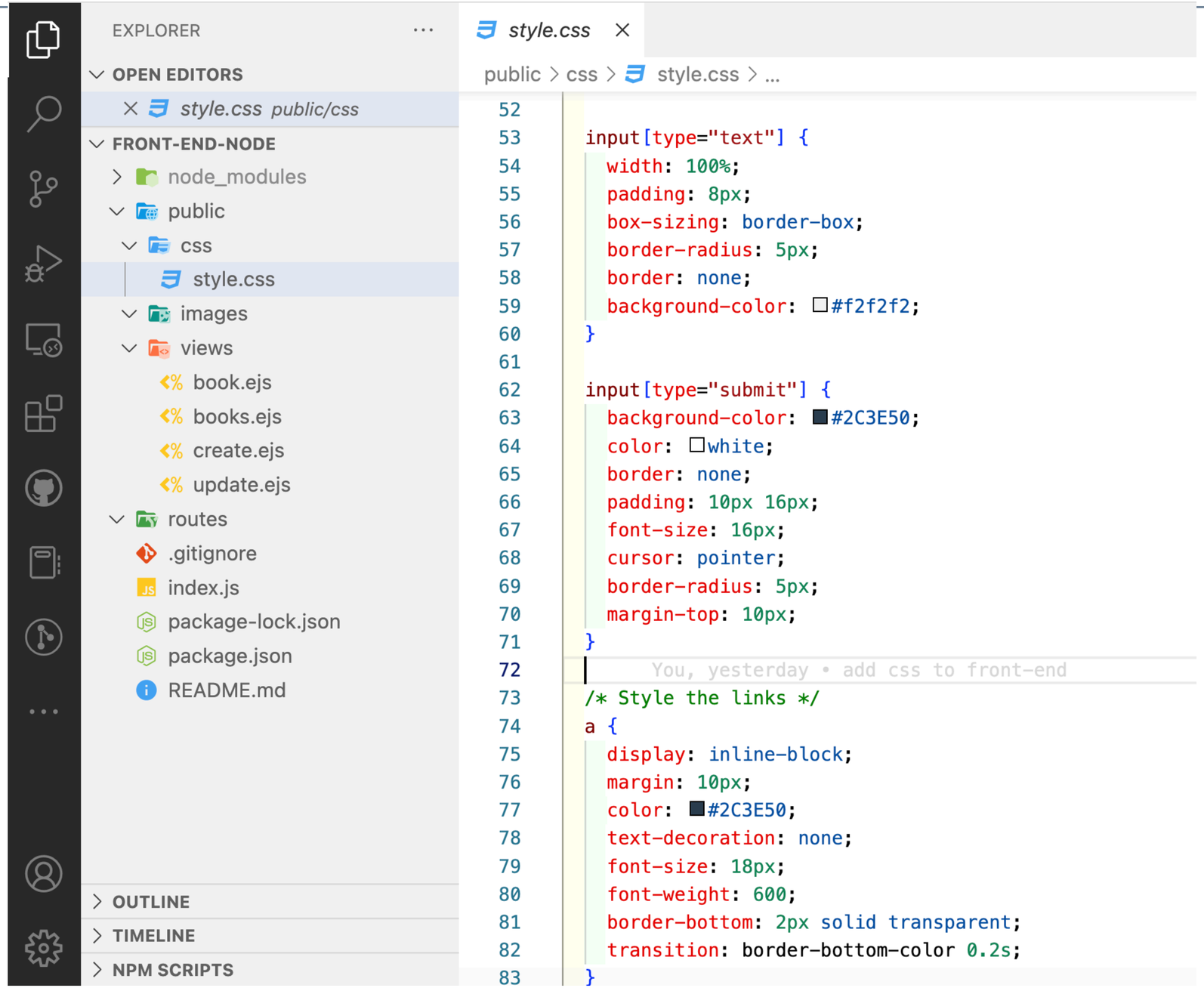
In this structure:

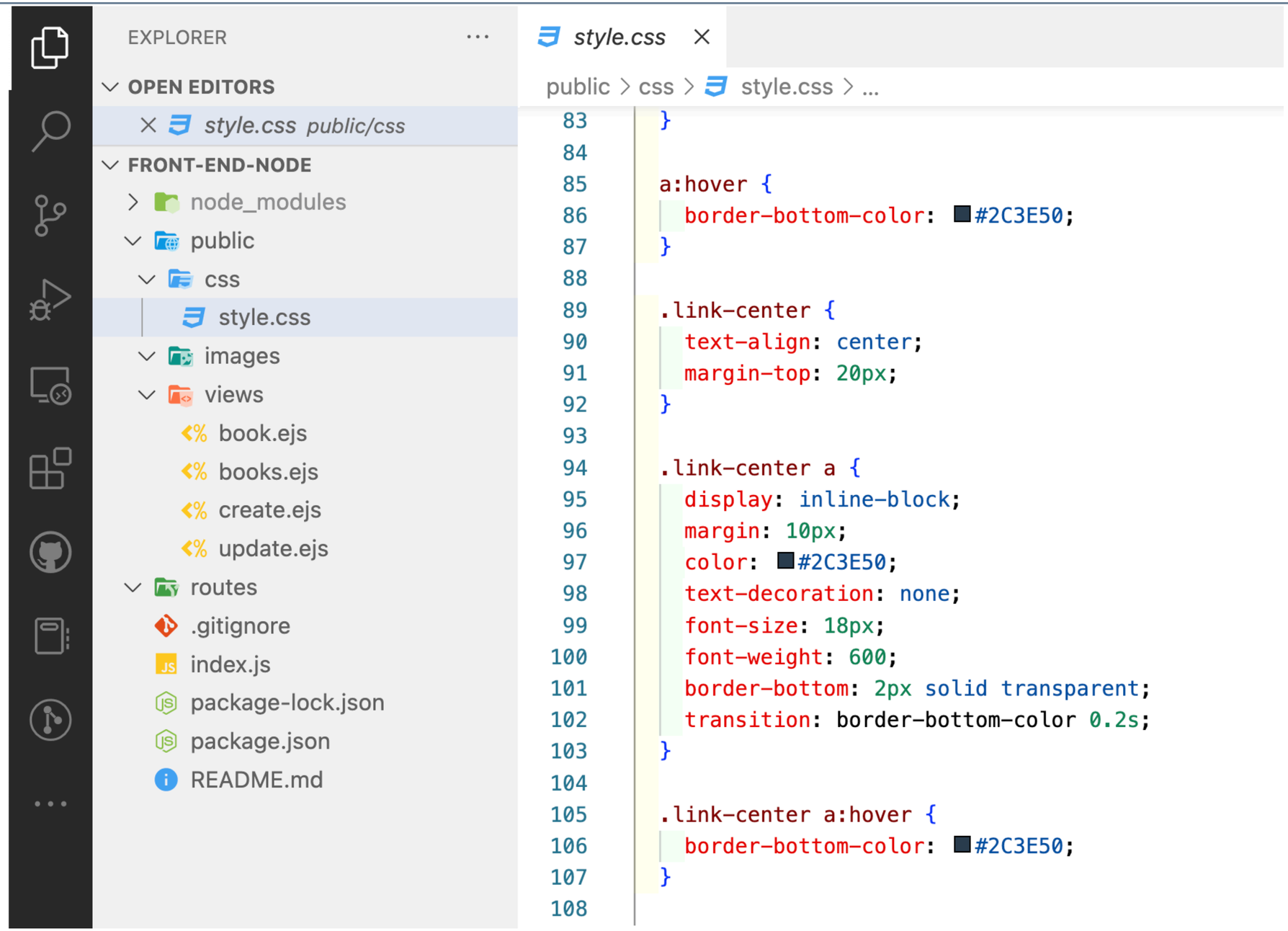
- The `views` directory contains EJS templates for each page of the application. In this example, there is only one template, `home.ejs`, but there could be additional templates for other pages of the application.
- The `public` directory contains static assets that will be served by the web server. In this example, there are subdirectories for `css` and `js`, which contain the CSS and JavaScript files for the application.
- The `node_modules` directory contains the installed Node.js modules that are used by the application.
- The `package.json` file contains information about the application and its dependencies.
- The `server.js` file is the main file of the application, which sets up the server and defines the routes and middleware.

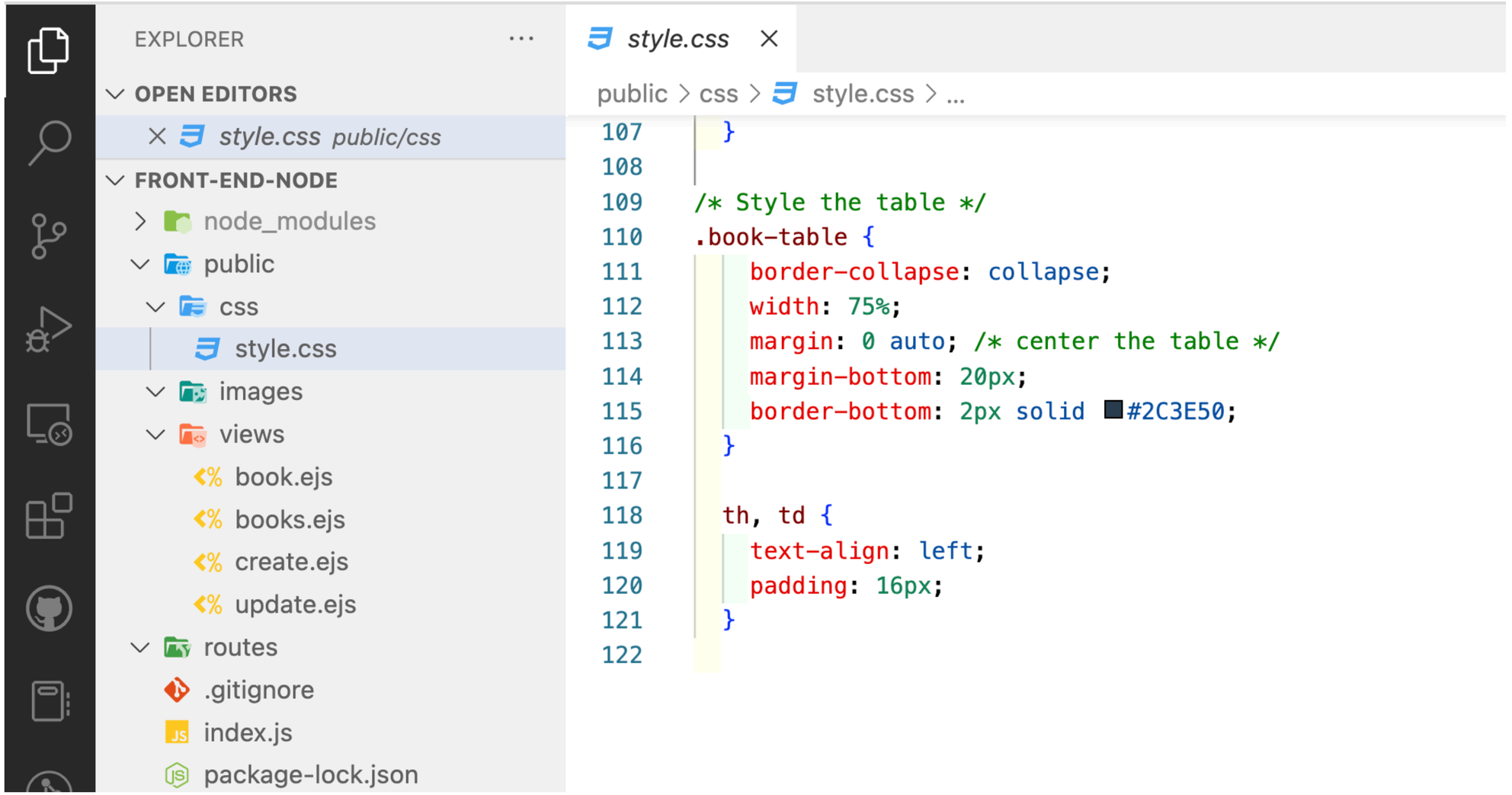

EXPLORER
OPEN EDITORS
package.json
FRONT-END-NODE
node_modules
public
css
style.css
images
views
book.ejs
books.ejs
create.ejs
update.ejs
routes
.gitignore
index.js
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
package.json
package.json > ...
You, 2 days ago | 1 author (You)
{ You, 4 days ago • DoneAndReady …
"name": "front-end-node",
"version": "1.0.0",
"description": "Front-End for the API Project Node.js",
"main": "index.js",
Debug
"scripts": {
"start": "node index.js"
},
"repository": {
"type": "git",
"url": "git+https://github.com/Anirach/Front-End-Node.git"
},
"keywords": [],
"author": "",
"license": "ISC",
"bugs": {
"url": "https://github.com/Anirach/Front-End-Node/issues"
},
"homepage": "https://github.com/Anirach/Front-End-Node#readme",
"dependencies": {
"axios": "^1.3.4",
"body-parser": "^1.20.2",
"ejs": "^3.1.8",
"express": "^4.18.2",
"http": "^0.0.1-security"
}
}

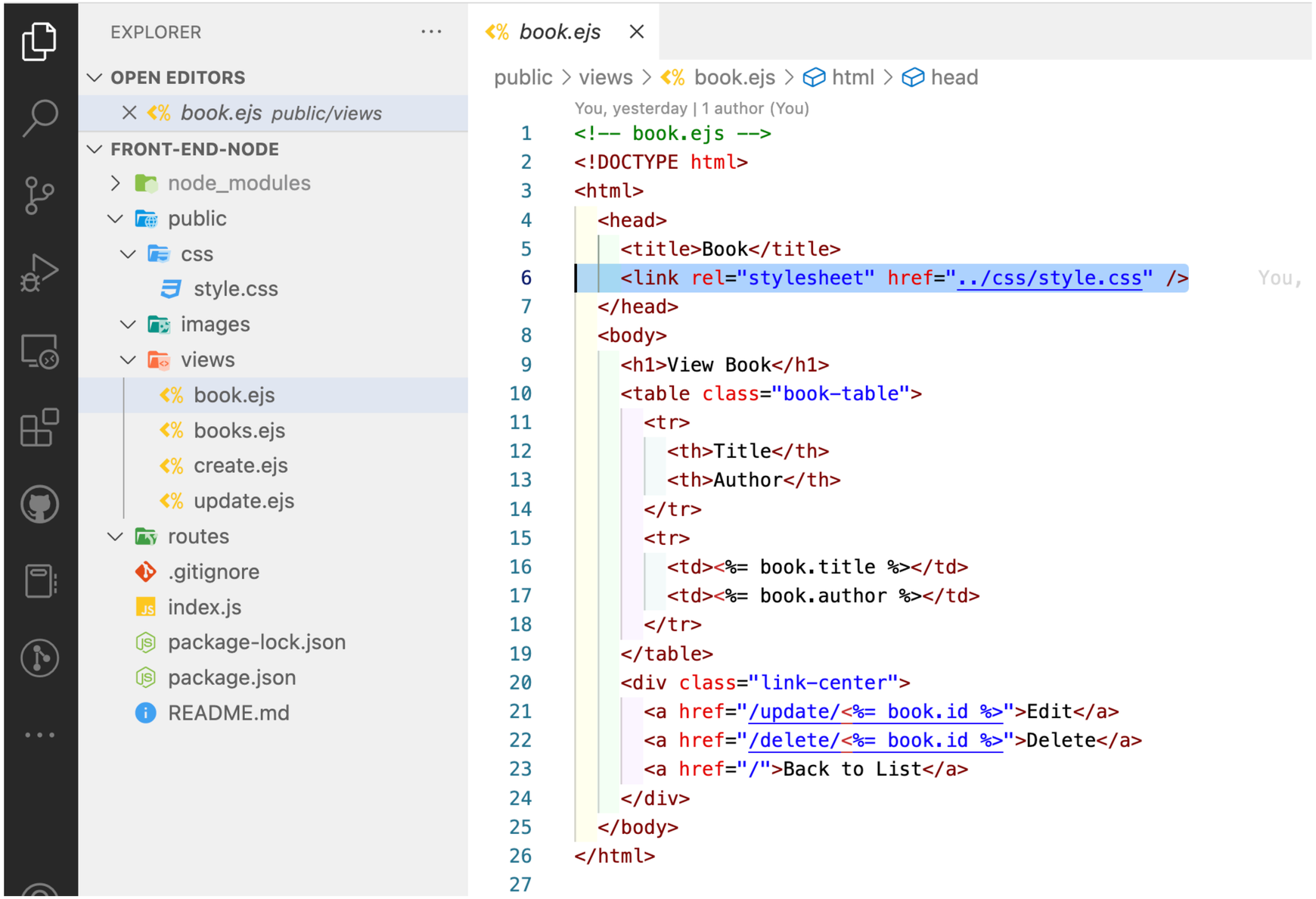

EXPLORER
OPEN EDITORS
style.css public/css
FRONT-END-NODE
node_modules
public
css
style.css
images
views
book.ejs
books.ejs
create.ejs
update.ejs
routes
.gitignore
index.js
package-lock.json
package.json
README.md
style.css
public > css > style.css > ...
You, yesterday | 1 author (You)
/* Set font family and size for the whole page */
body {
font-family: 'Open Sans', sans-serif;
font-size: 16px;
}
/* Center the main heading */
h1 {
text-align: center;
margin-top: 30px;
margin-bottom: 50px;
font-size: 36px;
color: #2C3E50;
}
/* Style the table */
table {
border-collapse: collapse;
width: 100%;
}
You, yesterday • add css to front-end
th, td {
text-align: left;
padding: 16px;
}


EXPLORER
OPEN EDITORS
style.css public/css
FRONT-END-NODE
node_modules
public
css
style.css
images
views
book.ejs
books.ejs
create.ejs
update.ejs
routes
.gitignore
index.js
package-lock.json
package.json
README.md
style.css
public > css > style.css > input[type="submit"]
25 }
26
27 th {
28 background-color: #2C3E50;
29 color: white;
30 font-weight: 600;
31 text-transform: uppercase;
32 }
33
34 tr:nth-child(even) {
35 background-color: #f5f5f5;
36 }
37
38 /* Style the form */
39 form {
40 width: 50%;
41 margin: auto;
42 border: 2px solid #2C3E50;
43 padding: 20px;
44 border-radius: 5px;
45 }
46
47 label {
48 display: block;
49 font-weight: bold;
50 margin-bottom: 10px;
51 }
52


EXPLORER
OPEN EDITORS
style.css public/css
FRONT-END-NODE
node_modules
public
css
style.css
images
views
book.ejs
books.ejs
create.ejs
update.ejs
routes
.gitignore
index.js
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
NPM SCRIPTS
style.css
public > css > style.css > ...
52
53 input[type="text"] {
54 width: 100%;
55 padding: 8px;
56 box-sizing: border-box;
57 border-radius: 5px;
58 border: none;
59 background-color: #f2f2f2;
60 }
61
62 input[type="submit"] {
63 background-color: #2C3E50;
64 color: white;
65 border: none;
66 padding: 10px 16px;
67 font-size: 16px;
68 cursor: pointer;
69 border-radius: 5px;
70 margin-top: 10px;
71 }
72 You, yesterday • add css to front-end
73 /* Style the links */
74 a {
75 display: inline-block;
76 margin: 10px;
77 color: #2C3E50;
78 text-decoration: none;
79 font-size: 18px;
80 font-weight: 600;
81 border-bottom: 2px solid transparent;
82 transition: border-bottom-color 0.2s;
83 }

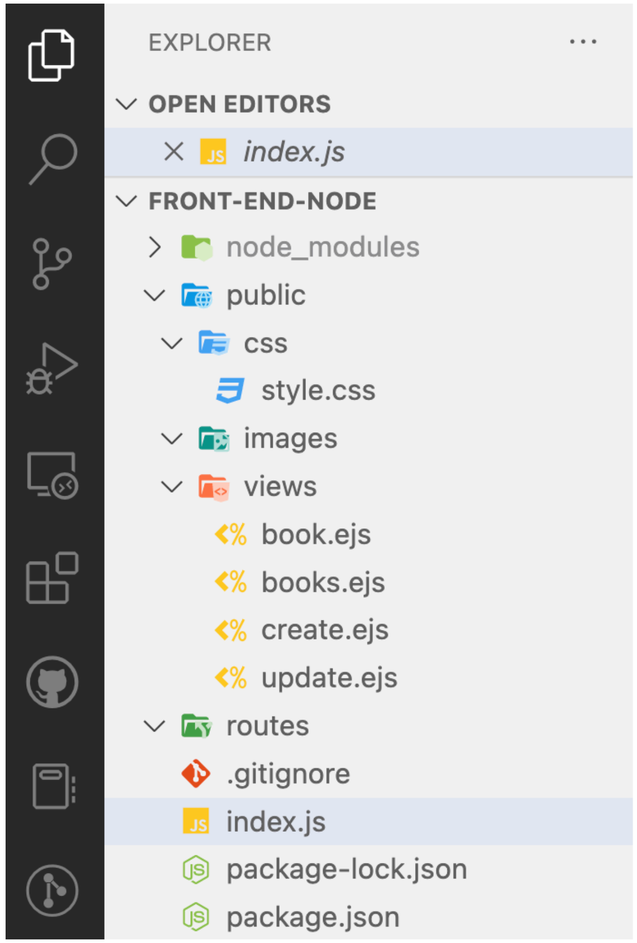

EXPLORER
OPEN EDITORS
style.css public/css
FRONT-END-NODE
node_modules
public
css
style.css
images
views
book.ejs
books.ejs
create.ejs
update.ejs
routes
.gitignore
index.js
package-lock.json
package.json
README.md
style.css
public > css > style.css > ...
}
a:hover {
border-bottom-color: #2C3E50;
}
.link-center {
text-align: center;
margin-top: 20px;
}
.link-center a {
display: inline-block;
margin: 10px;
color: #2C3E50;
text-decoration: none;
font-size: 18px;
font-weight: 600;
border-bottom: 2px solid transparent;
transition: border-bottom-color 0.2s;
}
.link-center a:hover {
border-bottom-color: #2C3E50;
}


EXPLORER
OPEN EDITORS
style.css public/css
FRONT-END-NODE
node_modules
public
css
style.css
images
views
book.ejs
books.ejs
create.ejs
update.ejs
routes
.gitignore
index.js
package-lock.json
style.css
public > css > style.css > ...
107 }
108
109 /* Style the table */
110 .book-table {
111 border-collapse: collapse;
112 width: 75%;
113 margin: 0 auto; /* center the table */
114 margin-bottom: 20px;
115 border-bottom: 2px solid #2C3E50;
116 }
117
118 th, td {
119 text-align: left;
120 padding: 16px;
121 }
122


EXPLORER
OPEN EDITORS
book.ejs public/views
FRONT-END-NODE
node_modules
public
css
style.css
images
views
book.ejs
books.ejs
create.ejs
update.ejs
routes
.gitignore
index.js
package-lock.json
package.json
README.md
book.ejs
public > views > book.ejs > html > head
You, yesterday | 1 author (You)
<!-- book.ejs -->
<!DOCTYPE html>
<html>
<head>
<title>Book</title>
<link rel="stylesheet" href="../css/style.css" />
You,
</head>
<body>
<h1>View Book</h1>
<table class="book-table">
<tr>
<th>Title</th>
<th>Author</th>
</tr>
<tr>
<td><%= book.title %></td>
<td><%= book.author %></td>
</tr>
</table>

<a href="/update/<%= book.id %>">Edit
<a href="/delete/<%= book.id %>">Delete
Back to List

</body>
</html>



index.js > app.get("/") callback

You, 23 hours ago | 1 author (You)

```
1  // index.js
2  // Description: Node.js HTML client
3  // requires: npm install express ejs axios body-parser
4
5  const express = require("express");
6  const axios = require("axios");
7  var bodyParser = require("body-parser");
8  const path = require("path");
9  const app = express();
10
11 // Base URL for the API
12 //const base_url = "https://api.example.com";
13 const base_url = "http://node41091-noderest.proen.app.ruk-com.cloud";
14
15 // Set the template engine
16 app.set("views", path.join(__dirname, "/public/views"));
17 app.set("view engine", "ejs");
18 app.use(bodyParser.json());
19 app.use(bodyParser.urlencoded({ extended: false }));
20
```

API WITH PGSQL

src > JS PGSqlCRUD.js > Book > author

```
1   const express = require('express');
2   const Sequelize = require('sequelize');
3   const app = express();
4   // parse incoming requests
5   app.use(express.json());
6
7   // set db url
8   const dbUrl = 'postgres://webadmin:MAIgsf81141@node40729-noderest.proen.app.ruk-com.cloud:11478/Books'
9
10  // create a connection to the database
11  const sequelize = new Sequelize(dbUrl);
```

pg DT

8.10.0 • Public • Published 5 days ago

Readme | Code Beta | 7 Dependencies | 8,224 Dependents | 215 Versions

node-postgres

build passing npm v8.10.0 downloads 18M/month

Non-blocking PostgreSQL client for Node.js. Pure JavaScript and optional native libpq bindings.

Install

```
$ npm install pg
```

⭐ Documentation ⭐

Features

- Pure JavaScript client and native libpq bindings share *the same API*

Install

```
> npm i pg
```

Repository

github.com/brianc/node-postgres

Homepage

github.com/brianc/node-postgres

Weekly Downloads

4,274,097

Version

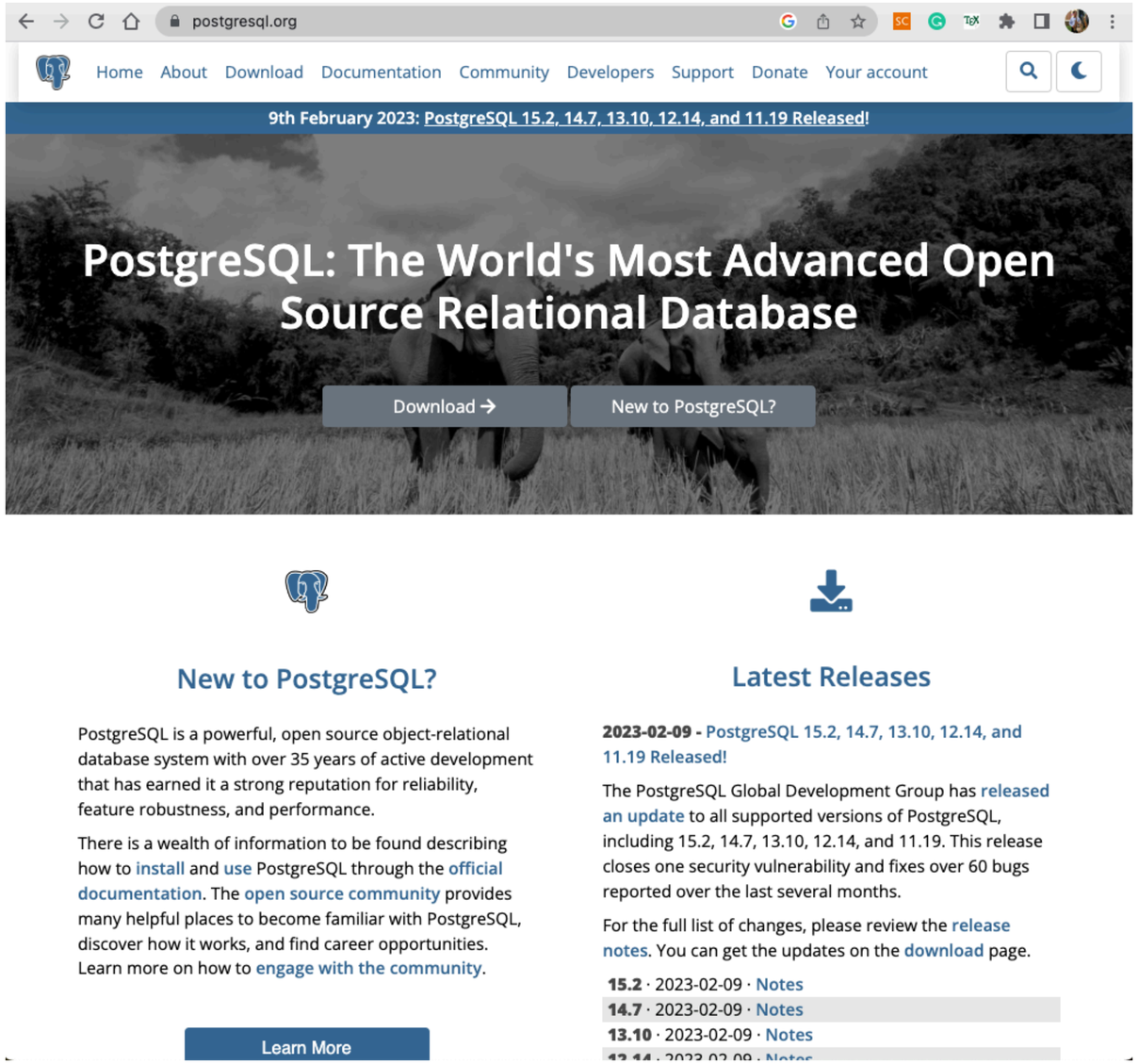
8.10.0

License

MIT

https://www.npmjs.com/package/pg

PGSQL-DB



https://www.postgresql.org/

PostgreSQL Tutorial

Welcome to the PostgreSQLTutorial.com website! This **PostgreSQL tutorial** helps you understand PostgreSQL quickly. You'll master PostgreSQL very fast through many practical examples and apply the knowledge in developing applications using PostgreSQL.

If you are...

- Looking for learning PostgreSQL fast.
- Developing applications using PostgreSQL as the back-end database management system.
- Migrating from other database management systems such as MySQL, Oracle, and Microsoft SQL Server to PostgreSQL.

You'll find all you need to know to get started with PostgreSQL quickly and effectively here on this website.

PostgreSQL tutorial demonstrates many unique features of PostgreSQL that make it the most advanced open-source database management system.

Getting Started with PostgreSQL

This section helps you get started with PostgreSQL by showing you how to install PostgreSQL on Windows, Linux, and macOS. You also learn how to connect to PostgreSQL using the psql tool as well as how to load a sample database into the PostgreSQL for practicing.

Basic PostgreSQL Tutorial

Search ...

POSTGRESQL QUICK START

What is PostgreSQL?

PostgreSQL Sample Database

Install PostgreSQL on Windows

Connect to Database

Load Sample Database

Install PostgreSQL on macOS

Install PostgreSQL on Linux

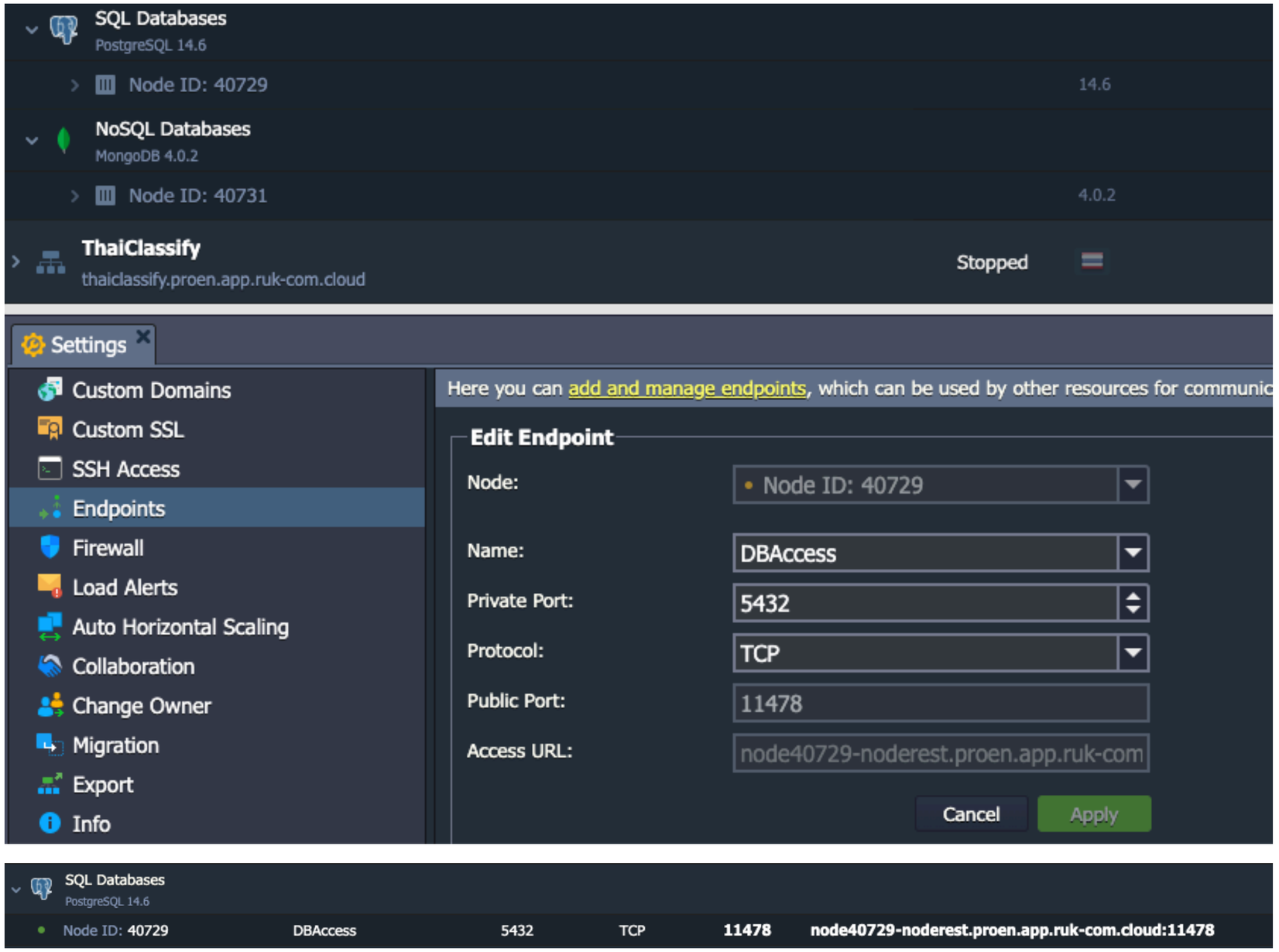
POSTGRESQL FUNDAMENTALS

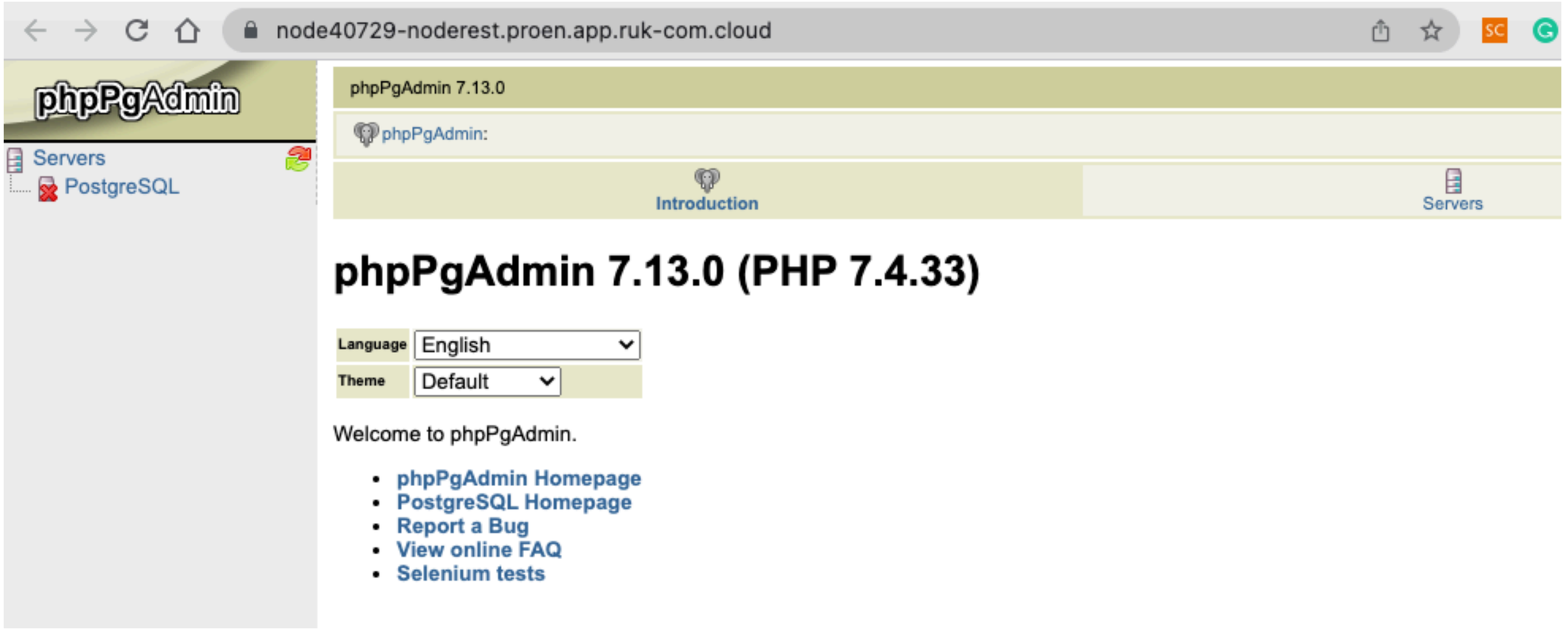
https://www.postgresqltutorial.com/

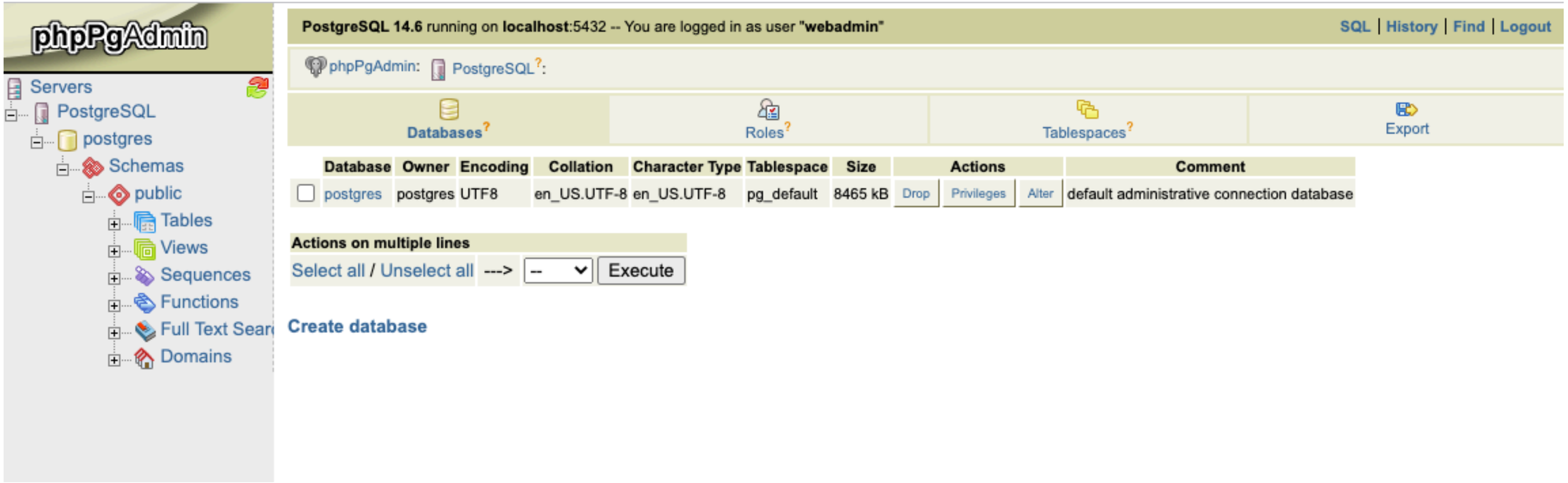
A new PostgreSQL server was successfully added to your NodeREST environment in Ruk-Com.

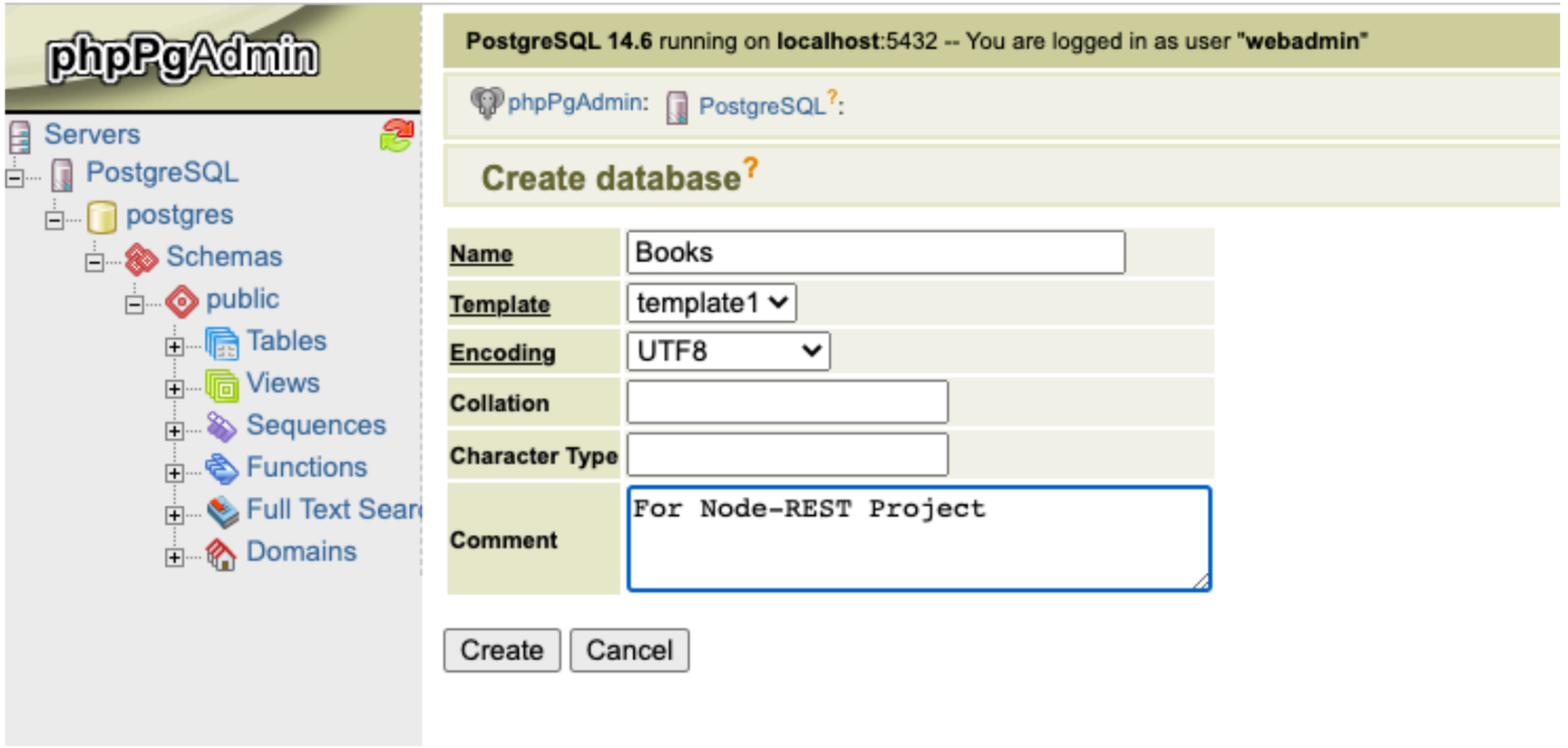
Please use the following data to access and manage the added node:

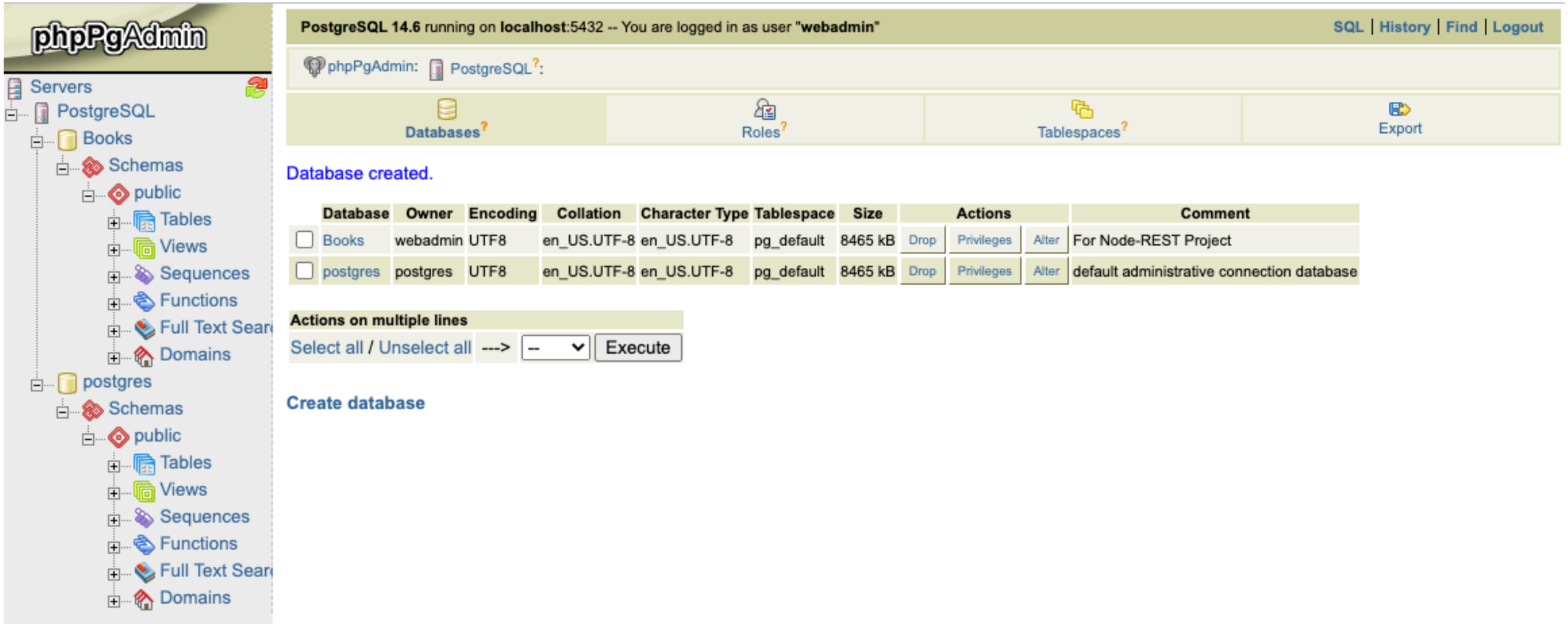
phpPgAdmin: https://node40729-noderest.proen.app.ruk-com.cloud
Login: webadmin
Password: MAlgsf81141

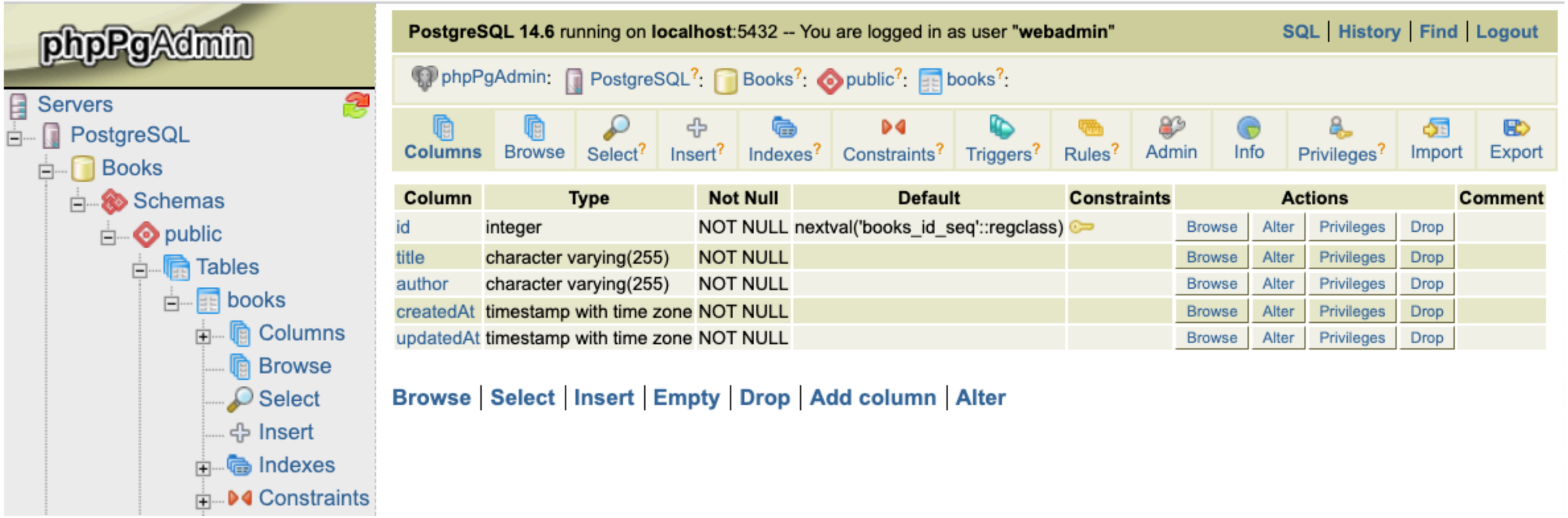

SQL Databases
PostgreSQL 14.6
Node ID: 40729
14.6
NoSQL Databases
MongoDB 4.0.2
Node ID: 40731
4.0.2
ThaiClassify
thaiclassify.proen.app.ruk-com.cloud
Stopped
Settings
Custom Domains
Custom SSL
SSH Access
Endpoints
Firewall
Load Alerts
Auto Horizontal Scaling
Collaboration
Change Owner
Migration
Export
Info
Here you can add and manage endpoints, which can be used by other resources for communic
Edit Endpoint
Node:
Node ID: 40729
Name:
DBAccess
Private Port:
5432
Protocol:
TCP
Public Port:
11478
Access URL:
node40729-noderest.proen.app.ruk-com
Cancel
Apply
SQL Databases
PostgreSQL 14.6
Node ID: 40729
DBAccess
5432
TCP
11478
node40729-noderest.proen.app.ruk-com.cloud:11478

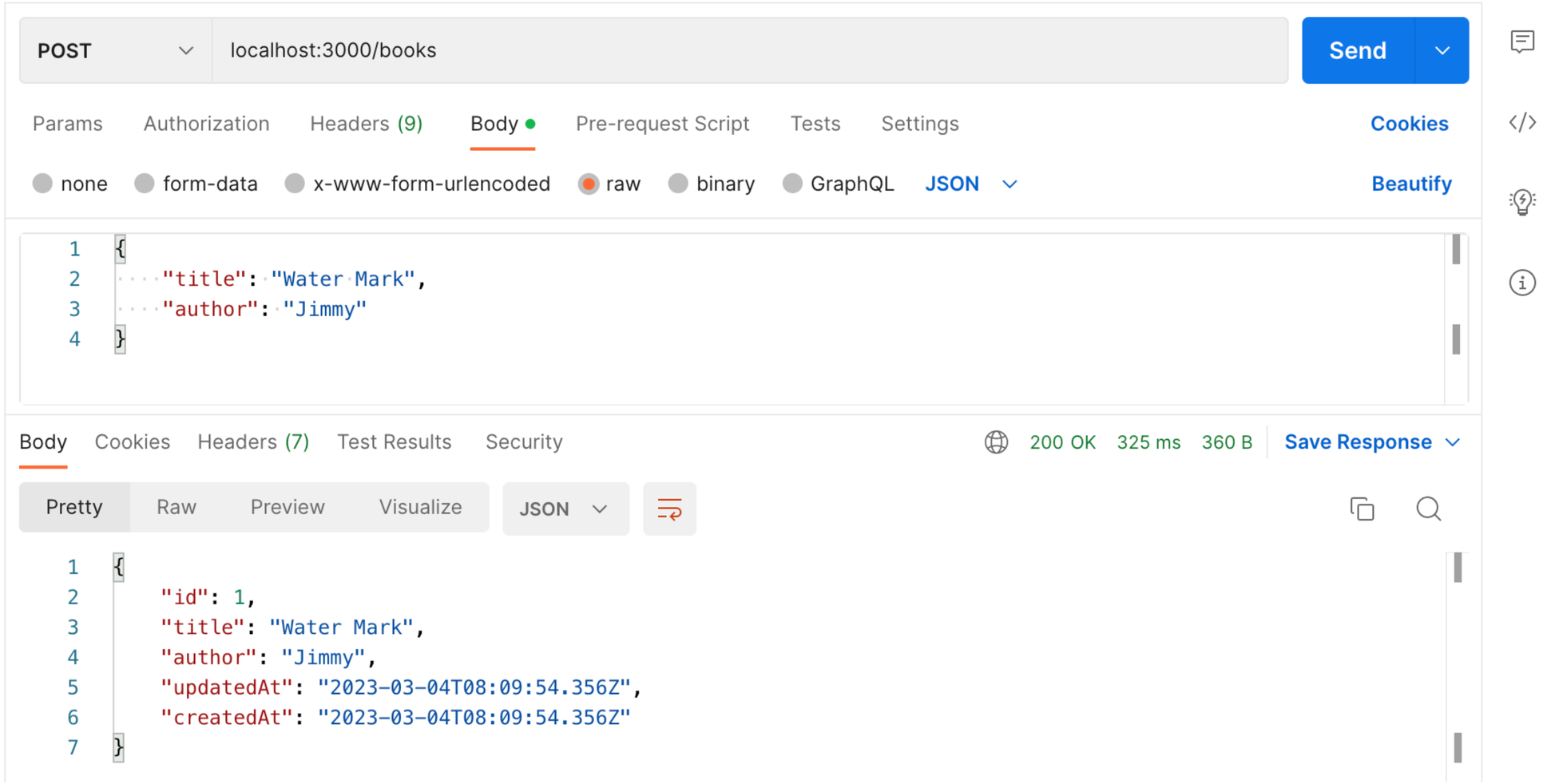

node40729-noderest.proen.app.ruk-com.cloud
phpPgAdmin
Servers
PostgreSQL
phpPgAdmin 7.13.0
phpPgAdmin:
Introduction
Servers
phpPgAdmin 7.13.0 (PHP 7.4.33)
Language
English
Theme
Default
Welcome to phpPgAdmin.
phpPgAdmin Homepage
PostgreSQL Homepage
Report a Bug
View online FAQ
Selenium tests

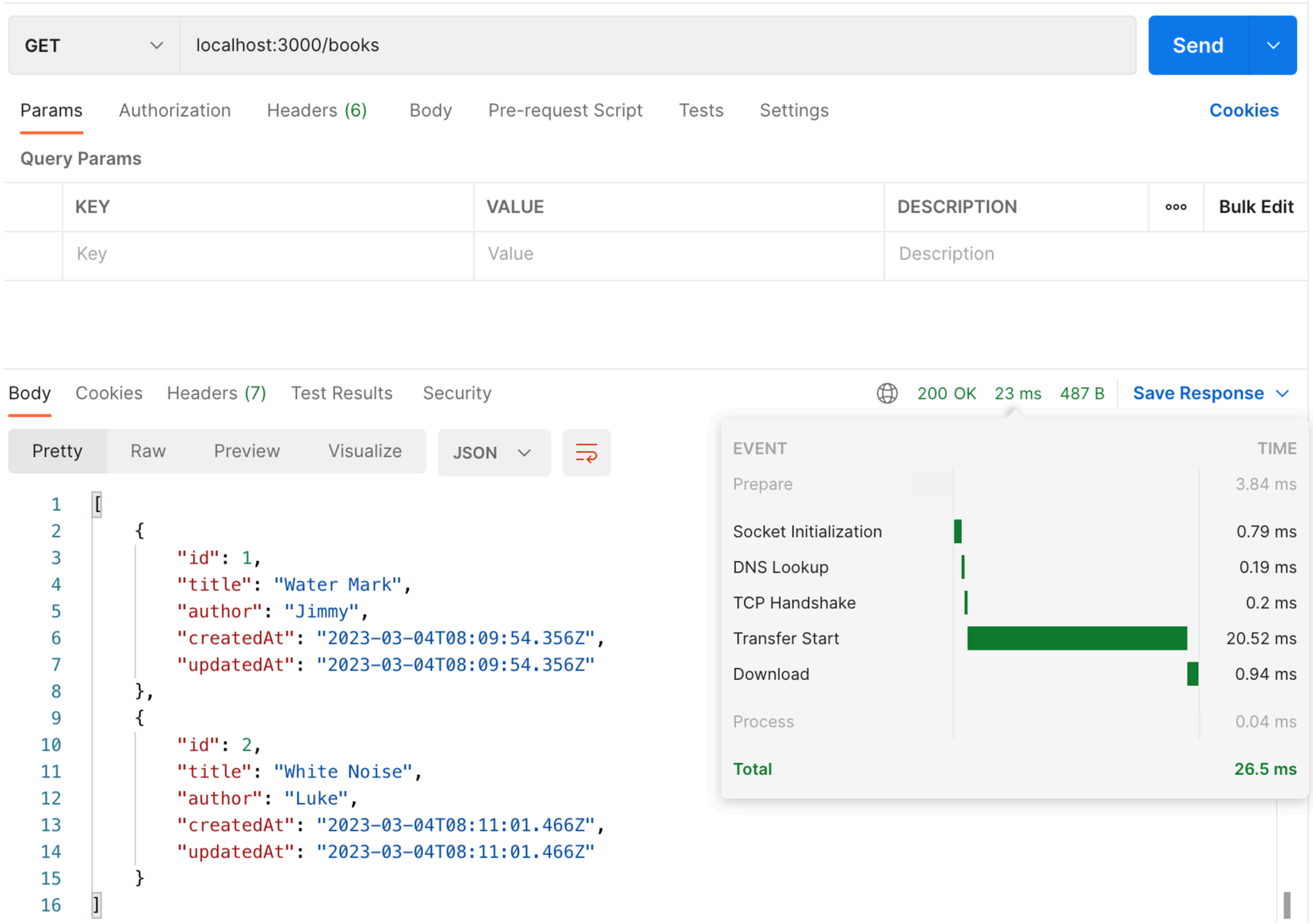

phpPgAdmin
Servers
PostgreSQL
postgres
Schemas
public
Tables
Views
Sequences
Functions
Full Text Sear
Domains
PostgreSQL 14.6 running on localhost:5432 -- You are logged in as user "webadmin"
SQL | History | Find | Logout
phpPgAdmin: PostgreSQL?:
Databases?
Roles?
Tablespaces?
Export
Database Owner Encoding Collation Character Type Tablespace Size Actions Comment
postgres postgres UTF8 en_US.UTF-8 en_US.UTF-8 pg_default 8465 kB Drop Privileges Alter default administrative connection database
Actions on multiple lines
Select all / Unselect all ---> -- Execute
Create database

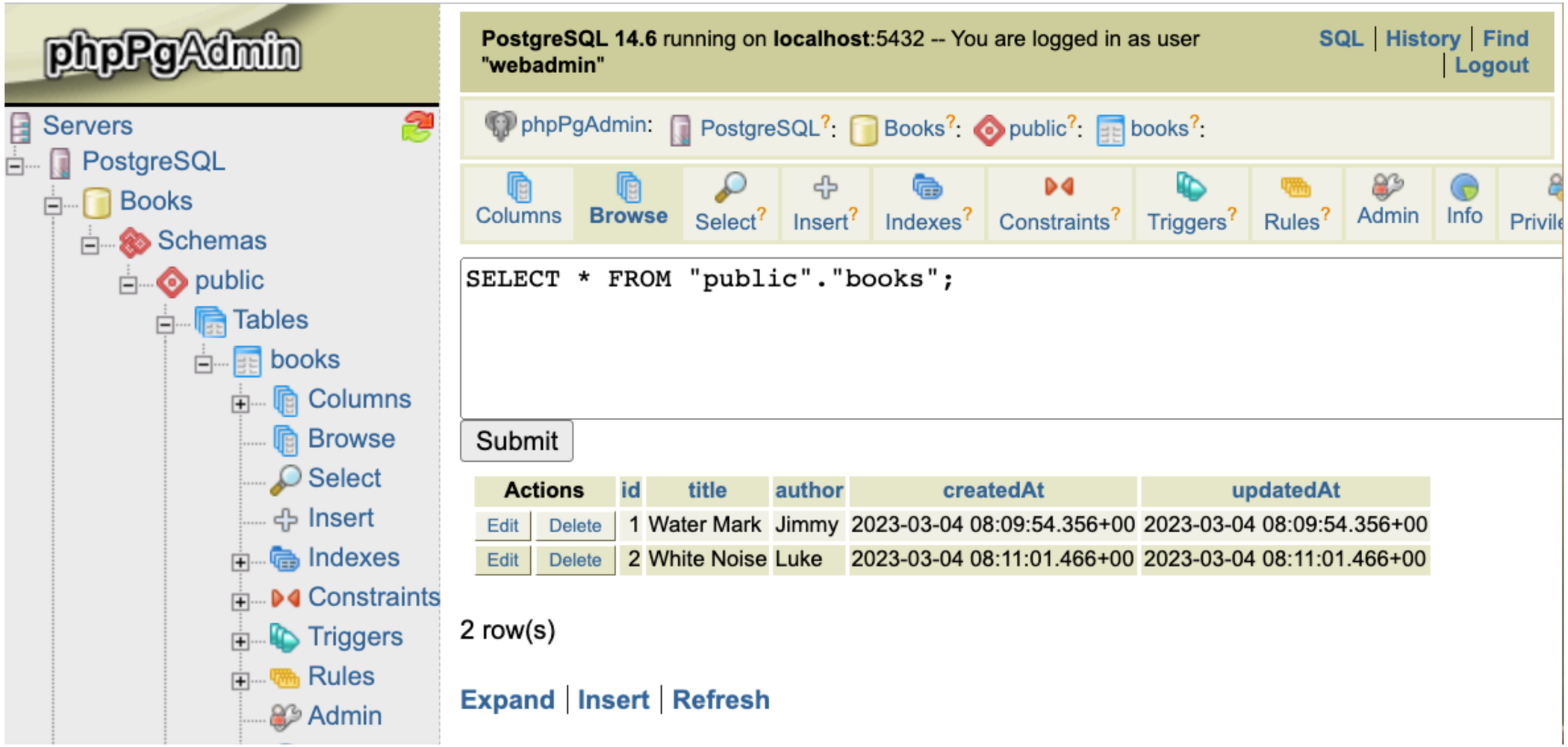

phpPgAdmin
Servers
PostgreSQL
postgres
Schemas
public
Tables
Views
Sequences
Functions
Full Text Sear
Domains
PostgreSQL 14.6 running on localhost:5432 -- You are logged in as user "webadmin"
phpPgAdmin: PostgreSQL?:
Create database?
Name
Books
Template
template1
Encoding
UTF8
Collation
Character Type
Comment
For Node-REST Project
Create
Cancel


phpPgAdmin
Servers
PostgreSQL
Books
Schemas
public
Tables
Views
Sequences
Functions
Full Text Sear
Domains
postgres
Schemas
public
Tables
Views
Sequences
Functions
Full Text Sear
Domains
PostgreSQL 14.6 running on localhost:5432 -- You are logged in as user "webadmin"
SQL | History | Find | Logout
phpPgAdmin: PostgreSQL?:
Databases?
Roles?
Tablespaces?
Export
Database created.
Database Owner Encoding Collation Character Type Tablespace Size Actions Comment
Books webadmin UTF8 en_US.UTF-8 en_US.UTF-8 pg_default 8465 kB Drop Privileges Alter For Node-REST Project
postgres postgres UTF8 en_US.UTF-8 en_US.UTF-8 pg_default 8465 kB Drop Privileges Alter default administrative connection database
Actions on multiple lines
Select all / Unselect all ---> -- Execute
Create database

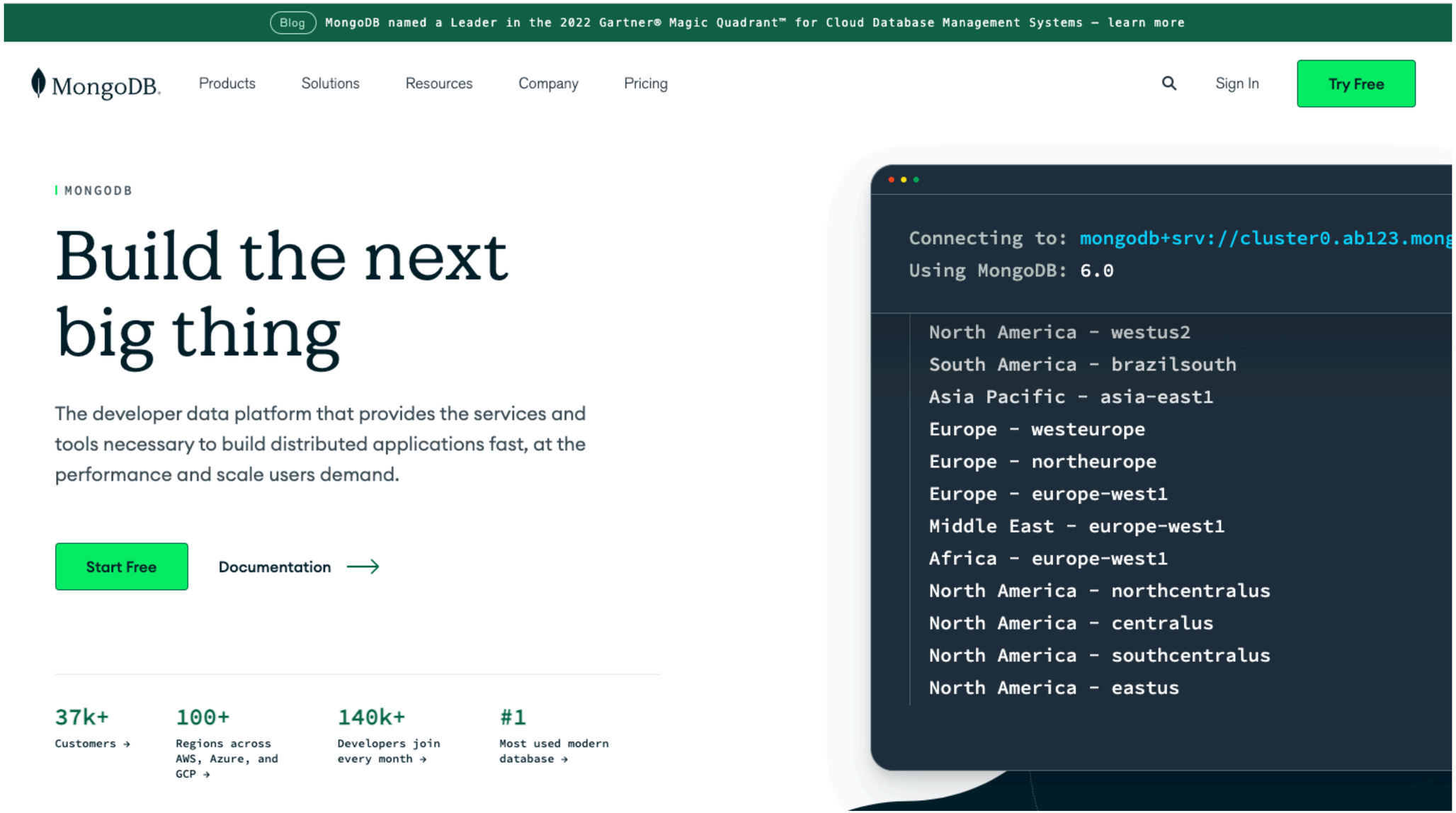

phpPgAdmin
Servers
PostgreSQL
Books
Schemas
public
Tables
books
Columns
Browse
Select
Insert
Indexes
Constraints
PostgreSQL 14.6 running on localhost:5432 -- You are logged in as user "webadmin"
SQL | History | Find | Logout
phpPgAdmin: PostgreSQL?: Books?: public?: books?:
Columns Browse Select? Insert? Indexes? Constraints? Triggers? Rules? Admin Info Privileges? Import Export
Column Type Not Null Default Constraints Actions Comment
id integer NOT NULL nextval('books_id_seq'::regclass) Browse Alter Privileges Drop
title character varying(255) NOT NULL Browse Alter Privileges Drop
author character varying(255) NOT NULL Browse Alter Privileges Drop
createdAt timestamp with time zone NOT NULL Browse Alter Privileges Drop
updatedAt timestamp with time zone NOT NULL Browse Alter Privileges Drop
Browse | Select | Insert | Empty | Drop | Add column | Alter


POST
localhost:3000/books
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"title": "Water Mark",
"author": "Jimmy"
}
Body
Cookies
Headers (7)
Test Results
Security
200 OK
325 ms
360 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"id": 1,
"title": "Water Mark",
"author": "Jimmy",
"updatedAt": "2023-03-04T08:09:54.356Z",
"createdAt": "2023-03-04T08:09:54.356Z"
}


GET
localhost:3000/books
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (7)
Test Results
Security
200 OK
23 ms
487 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"title": "Water Mark",
"author": "Jimmy",
"createdAt": "2023-03-04T08:09:54.356Z",
"updatedAt": "2023-03-04T08:09:54.356Z"
},
{
"id": 2,
"title": "White Noise",
"author": "Luke",
"createdAt": "2023-03-04T08:11:01.466Z",
"updatedAt": "2023-03-04T08:11:01.466Z"
}
]
EVENT
TIME
Prepare
3.84 ms
Socket Initialization
0.79 ms
DNS Lookup
0.19 ms
TCP Handshake
0.2 ms
Transfer Start
20.52 ms
Download
0.94 ms
Process
0.04 ms
Total
26.5 ms


phpPgAdmin
Servers
PostgreSQL
Books
Schemas
public
Tables
books
Columns
Browse
Select
Insert
Indexes
Constraints
Triggers
Rules
Admin
PostgreSQL 14.6 running on localhost:5432 -- You are logged in as user "webadmin"
SQL | History | Find | Logout
phpPgAdmin: PostgreSQL?: Books?: public?: books?:
Columns
Browse
Select?
Insert?
Indexes?
Constraints?
Triggers?
Rules?
Admin
Info
SELECT * FROM "public"."books";
Submit
Actions
id
title
author
createdAt
updatedAt
Edit
Delete
1
Water Mark
Jimmy
2023-03-04 08:09:54.356+00
2023-03-04 08:09:54.356+00
Edit
Delete
2
White Noise
Luke
2023-03-04 08:11:01.466+00
2023-03-04 08:11:01.466+00
2 row(s)
Expand | Insert | Refresh

MONGO-DB



https://www.mongodb.com/

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

Database

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collection

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

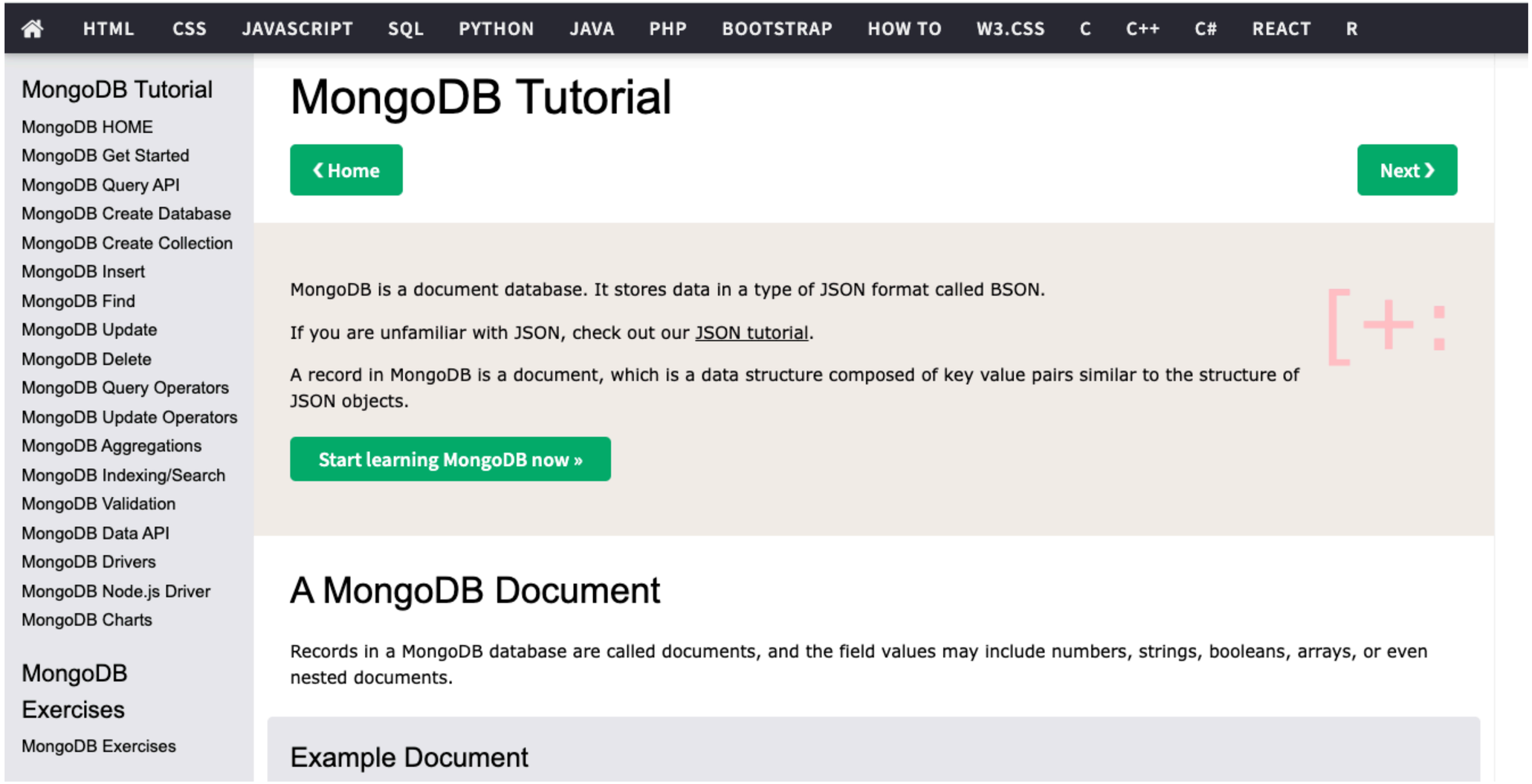
Document

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

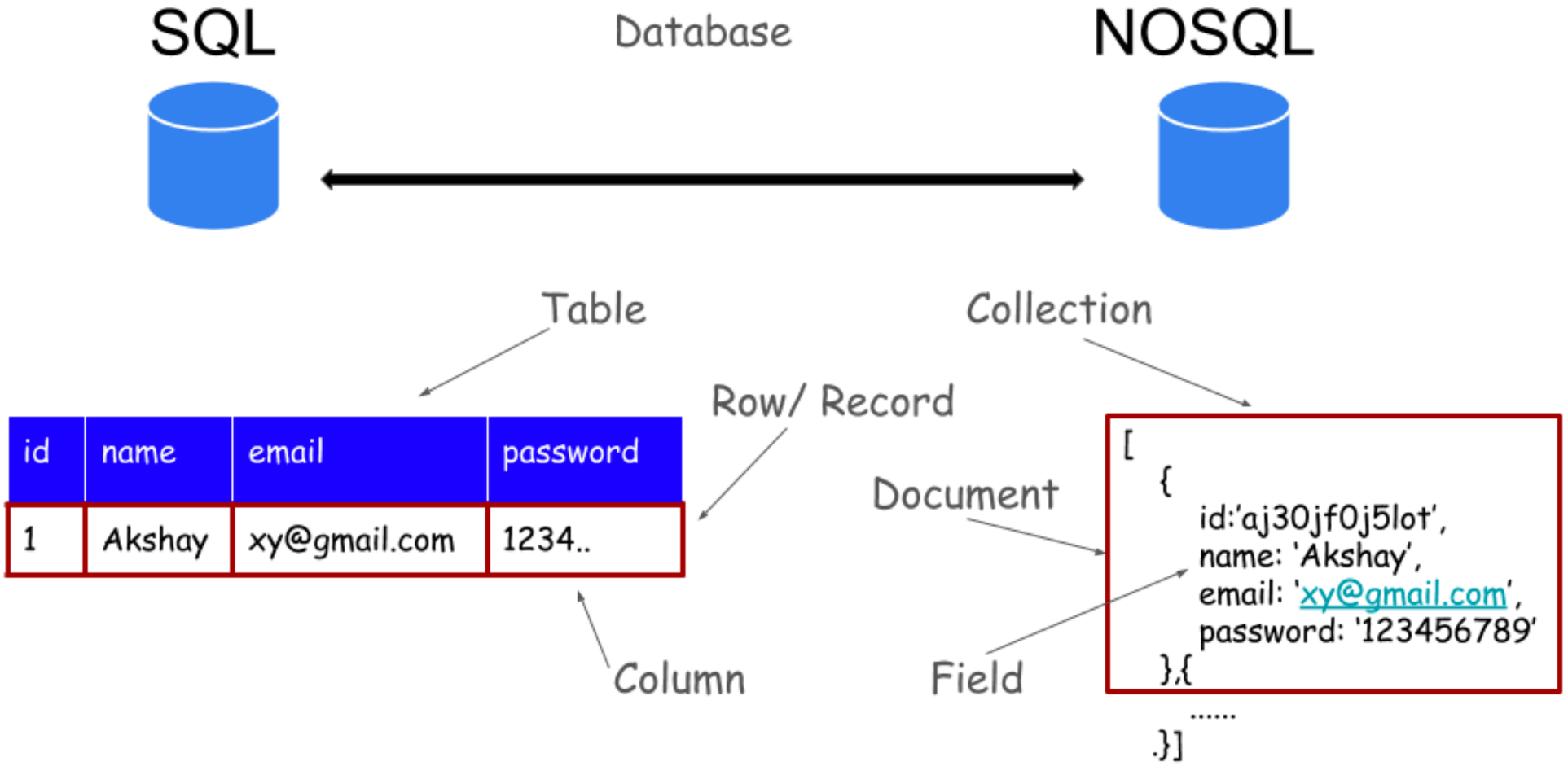
The following table shows the relationship of RDBMS terminology with MongoDB.

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by MongoDB itself)
Database Server and Client	
mysqld/Oracle	mongod

https://www.tutorialspoint.com/mongodb/mongodb_overview.h



https://www.w3schools.com/mongodb/

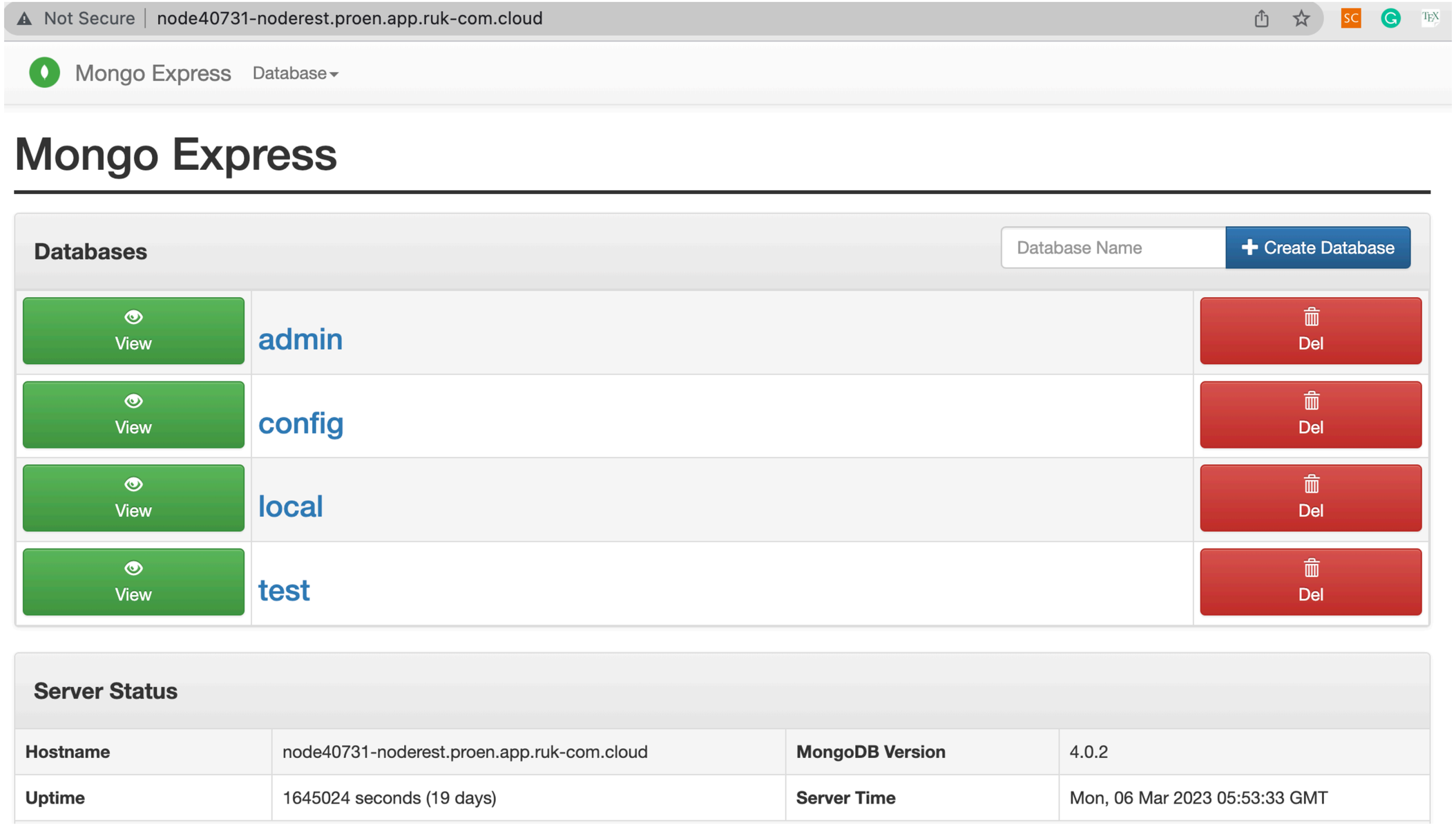

SQL
Database
NOSQL
Table
Collection
Row/ Record
id
name
email
password
1
Akshay
xy@gmail.com
1234..
Document
Column
Field
[
{
id:'aj30jf0j5lot',
name: 'Akshay',
email: 'xy@gmail.com',
password: '123456789'
},{
.....
.}]

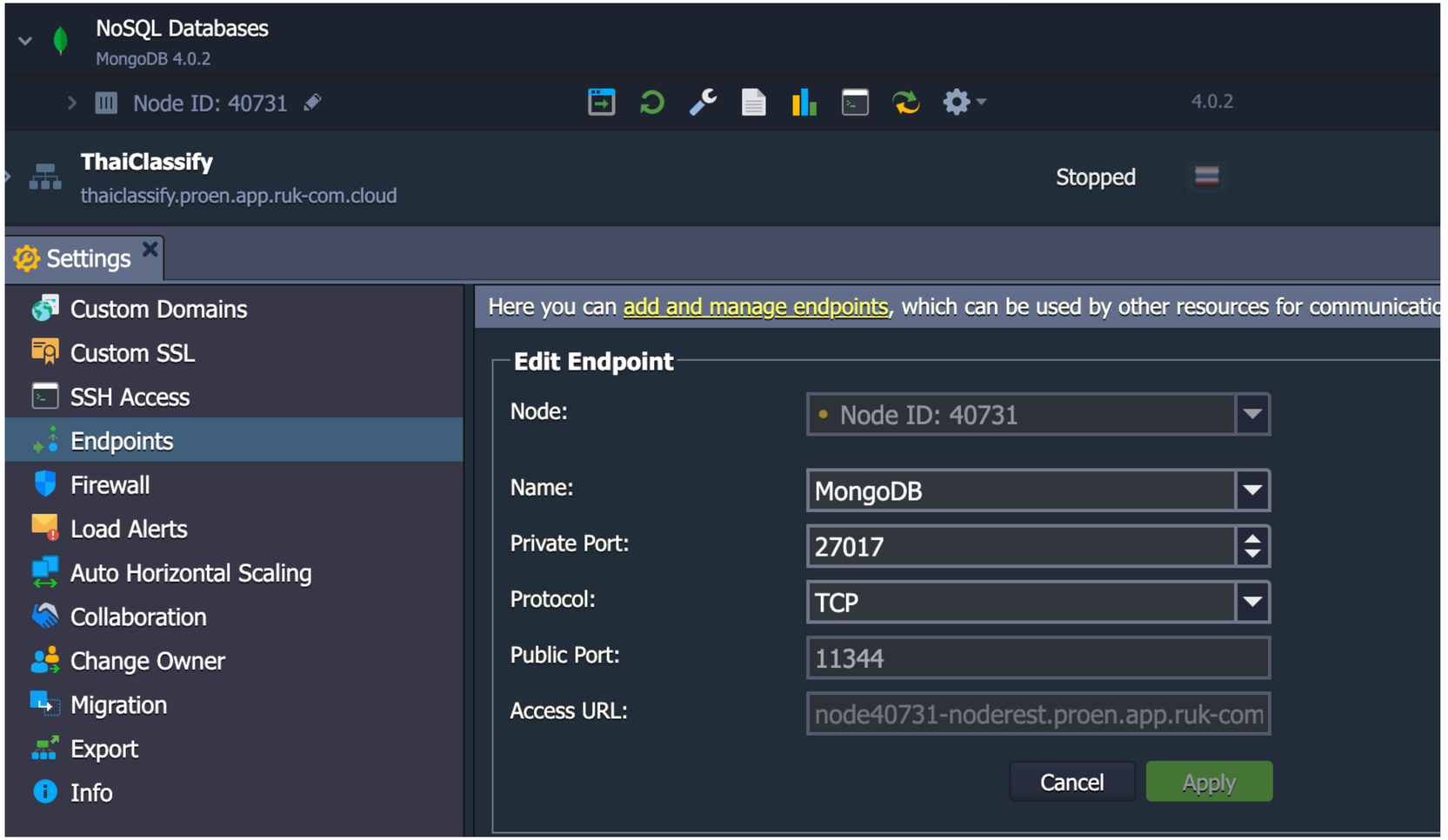
A new MongoDB server was successfully added to your NodeREST environment in Ruk-Com.

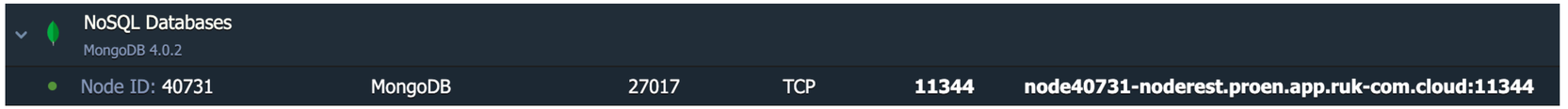
Please use the following data to access and manage the added node:

Admin Panel: http://node40731-noderest.proen.app.ruk-com.cloud/
Login: admin
Password: SBFsqa14913

Connect to MongoDB from your application code using the linked instruction and the following details:


Not Secure | node40731-noderest.proen.app.ruk-com.cloud
Mongo Express
Database
Mongo Express
Databases
Database Name
Create Database
View
admin
Del
View
config
Del
View
local
Del
View
test
Del
Server Status
Hostname
node40731-noderest.proen.app.ruk-com.cloud
MongoDB Version
4.0.2
Uptime
1645024 seconds (19 days)
Server Time
Mon, 06 Mar 2023 05:53:33 GMT


NoSQL Databases
MongoDB 4.0.2
Node ID: 40731
4.0.2
ThaiClassify
thaiclassify.proen.app.ruk-com.cloud
Stopped
Settings
Custom Domains
Custom SSL
SSH Access
Endpoints
Firewall
Load Alerts
Auto Horizontal Scaling
Collaboration
Change Owner
Migration
Export
Info
Here you can add and manage endpoints, which can be used by other resources for communicatio
Edit Endpoint
Node:
Node ID: 40731
Name:
MongoDB
Private Port:
27017
Protocol:
TCP
Public Port:
11344
Access URL:
node40731-noderest.proen.app.ruk-com
Cancel
Apply


NoSQL Databases
MongoDB 4.0.2
Node ID: 40731
MongoDB
27017
TCP
11344
node40731-noderest.proen.app.ruk-com.cloud:11344

mongoose

7.0.1 • Public • Published 5 days ago

Readme | Code Beta | 7 Dependencies | 13,580 Dependents | 772 Versions

Mongoose

Mongoose is a MongoDB object modeling tool designed to work in an asynchronous environment. Mongoose supports Node.js and Deno (alpha).

Documentation

The official documentation website is mongoosejs.com.

Mongoose 7.0.0 was released on February 27, 2023. You can find more details on backwards breaking changes in 7.0.0 on our docs site.

Support

Install

```
npm i mongoose
```

Repository

github.com/Automattic/mongoose

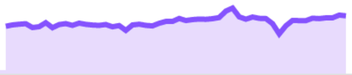
Homepage

mongoosejs.com

❤Fund this package

Weekly Downloads

2,136,278



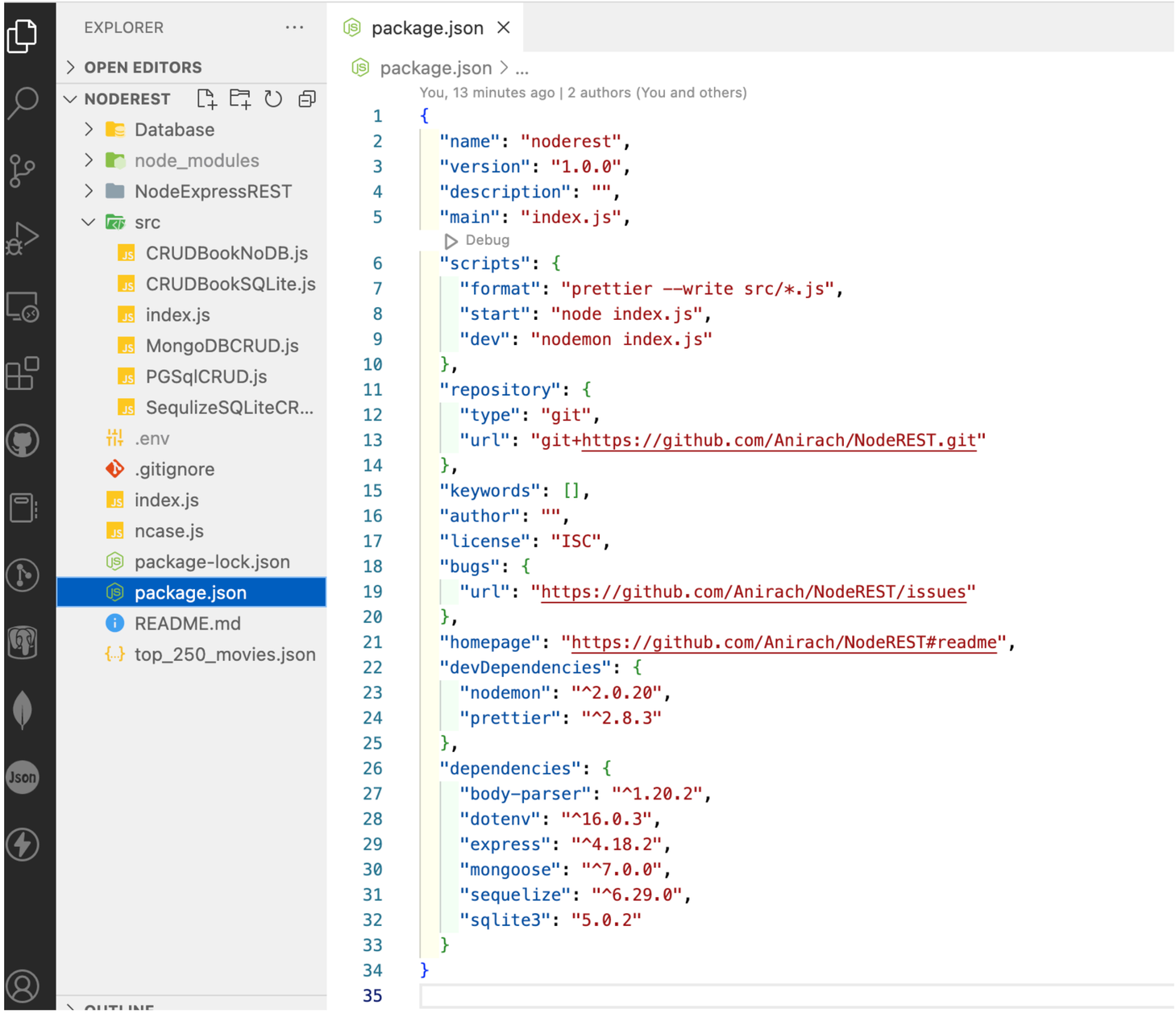
Version

7.0.1

License

MIT

https://www.npmjs.com/package/mongoose


EXPLORER
OPEN EDITORS
NODEREST
Database
node_modules
NodeExpressREST
src
CRUDBookNoDB.js
CRUDBookSQLite.js
index.js
MongoDBCRUD.js
PGSqlCRUD.js
SequlizeSQLiteCR...
.env
.gitignore
index.js
ncase.js
package-lock.json
package.json
README.md
top_250_movies.json
package.json
package.json > ...
You, 13 minutes ago | 2 authors (You and others)
{
"name": "noderest",
"version": "1.0.0",
"description": "",
"main": "index.js",
Debug
"scripts": {
"format": "prettier --write src/*.js",
"start": "node index.js",
"dev": "nodemon index.js"
},
"repository": {
"type": "git",
"url": "git+https://github.com/Anirach/NodeREST.git"
},
"keywords": [],
"author": "",
"license": "ISC",
"bugs": {
"url": "https://github.com/Anirach/NodeREST/issues"
},
"homepage": "https://github.com/Anirach/NodeREST#readme",
"devDependencies": {
"nodemon": "^2.0.20",
"prettier": "^2.8.3"
},
"dependencies": {
"body-parser": "^1.20.2",
"dotenv": "^16.0.3",
"express": "^4.18.2",
"mongoose": "^7.0.0",
"sequelize": "^6.29.0",
"sqlite3": "5.0.2"
}
}

src > MongoDBCRUD.js > ...

Anirach, 6 minutes ago | 2 authors (You and others)

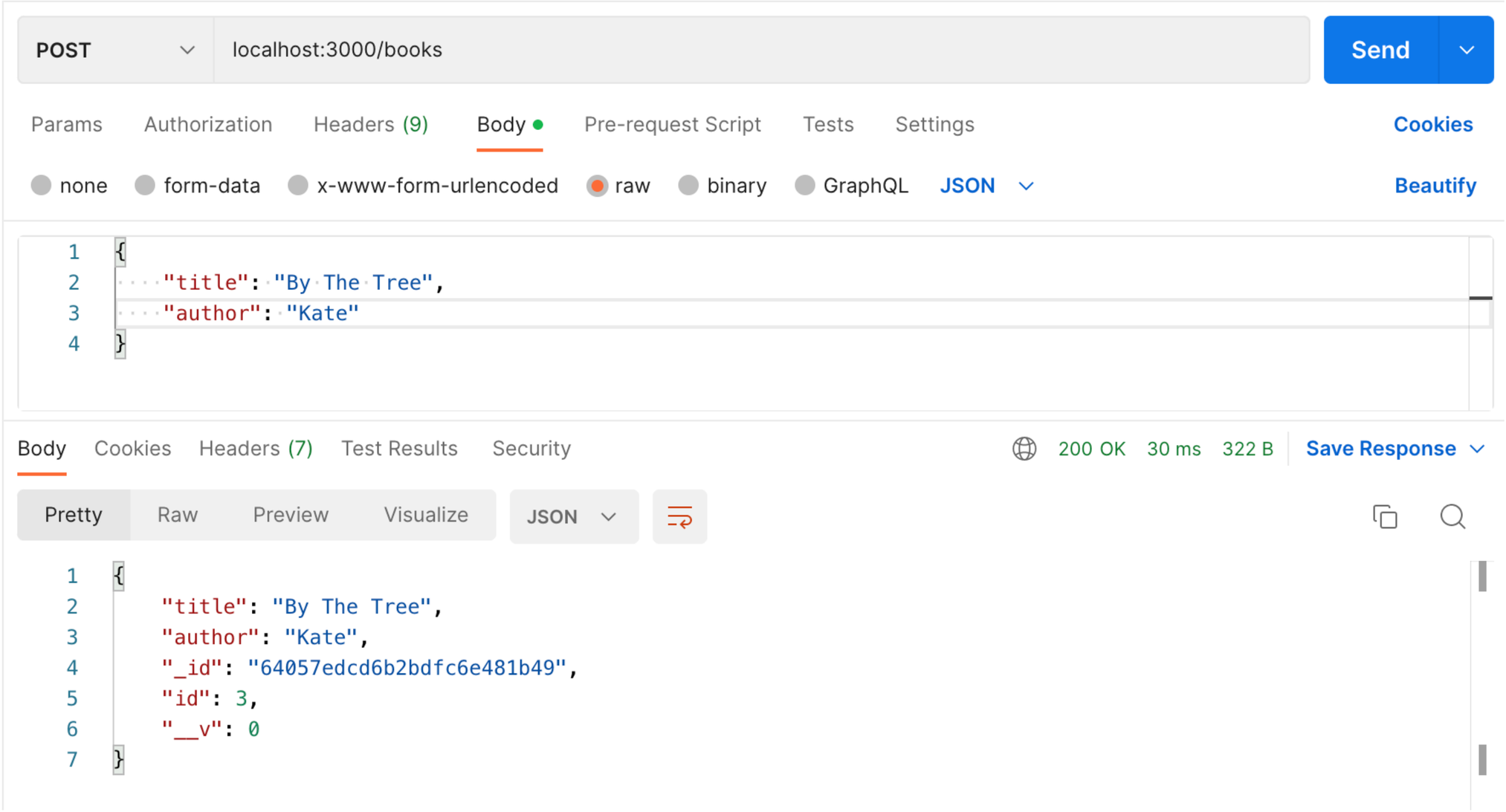
```
1  // Description: REST API with MongoDB
2  // npm install express mongoose  body-parser
3  // Run this file with node MongoDBREST.js
4  // Test with Postman
5
6  const express = require("express");
7  const mongoose = require("mongoose");
8  const bodyParser = require("body-parser");
9
10 // Database connection
11 mongoose.connect(
12   "mongodb://admin:SBFsqa14913@node40731-noderest.proen.app.ruk-com.cloud:11344",
13   {
14     useNewUrlParser: true,
15     useUnifiedTopology: true,
16   }
17 );
18
19 const Book = mongoose.model("Book", {
20   id: {
21     type: Number,
22     unique: true, // Ensures uniqueness of the "id" field
23     required: true, // If you want "id" to be required
24   },
25   title: String,
26   author: String,
27 });
28
```

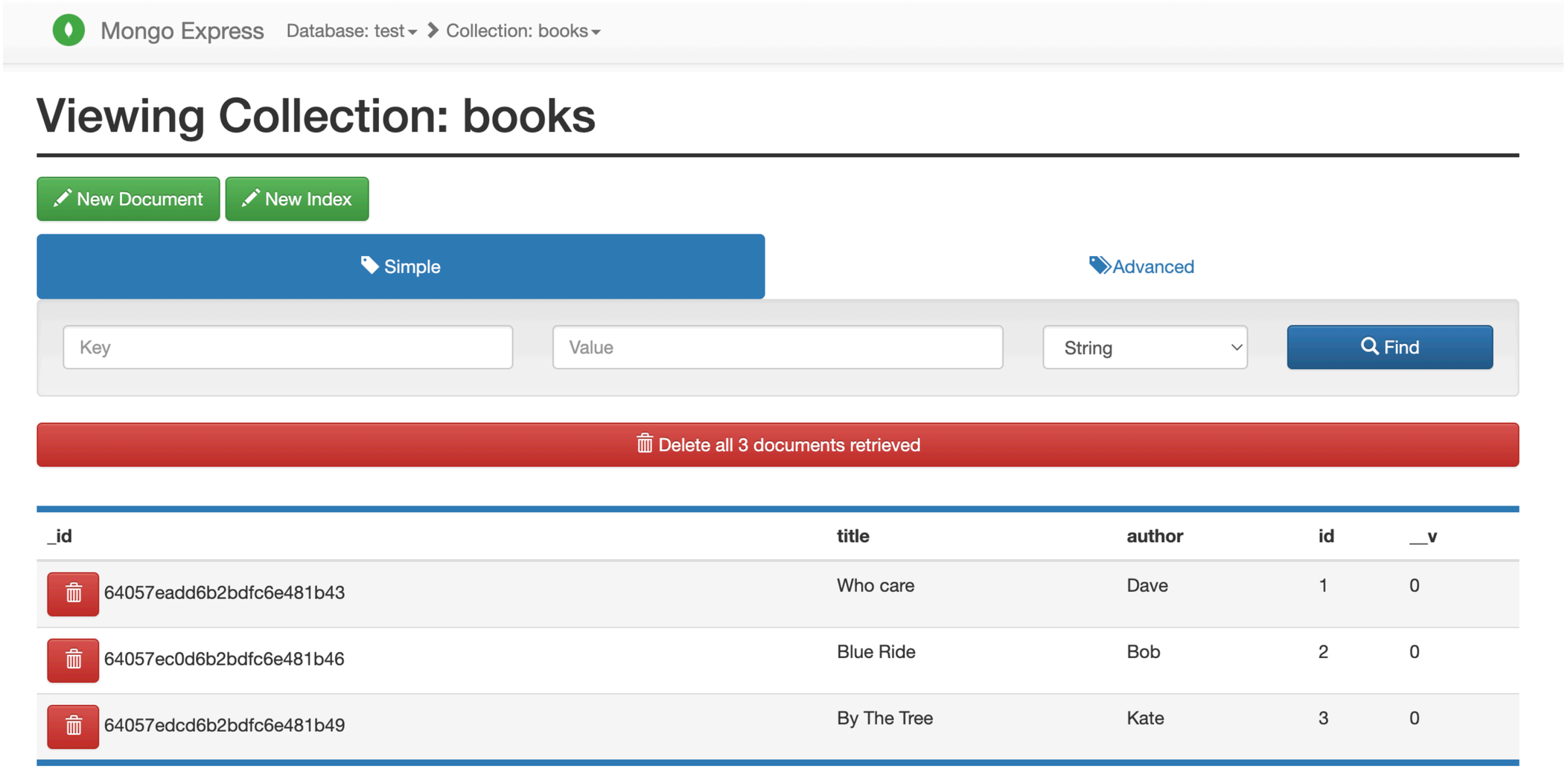
```
28
29  const app = express();
30  app.use(bodyParser.json());
31
32  // Create
33  app.post("/books", async (req, res) => {
34    try {
35      // Get the last book record to determine the next ID
36      const lastBook = await Book.findOne().sort({ id: -1 });
37      const nextId = lastBook ? lastBook.id + 1 : 1;
38
39      // Create a new book with the next ID
40      const book = new Book({
41        id: nextId, // Set the custom "id" field
42        ...req.body, // Include other book data from the request body
43      });
44
45      await book.save();
46      res.send(book);
47    } catch (error) {
48      res.status(500).send(error);
49    }
50  });
51
```

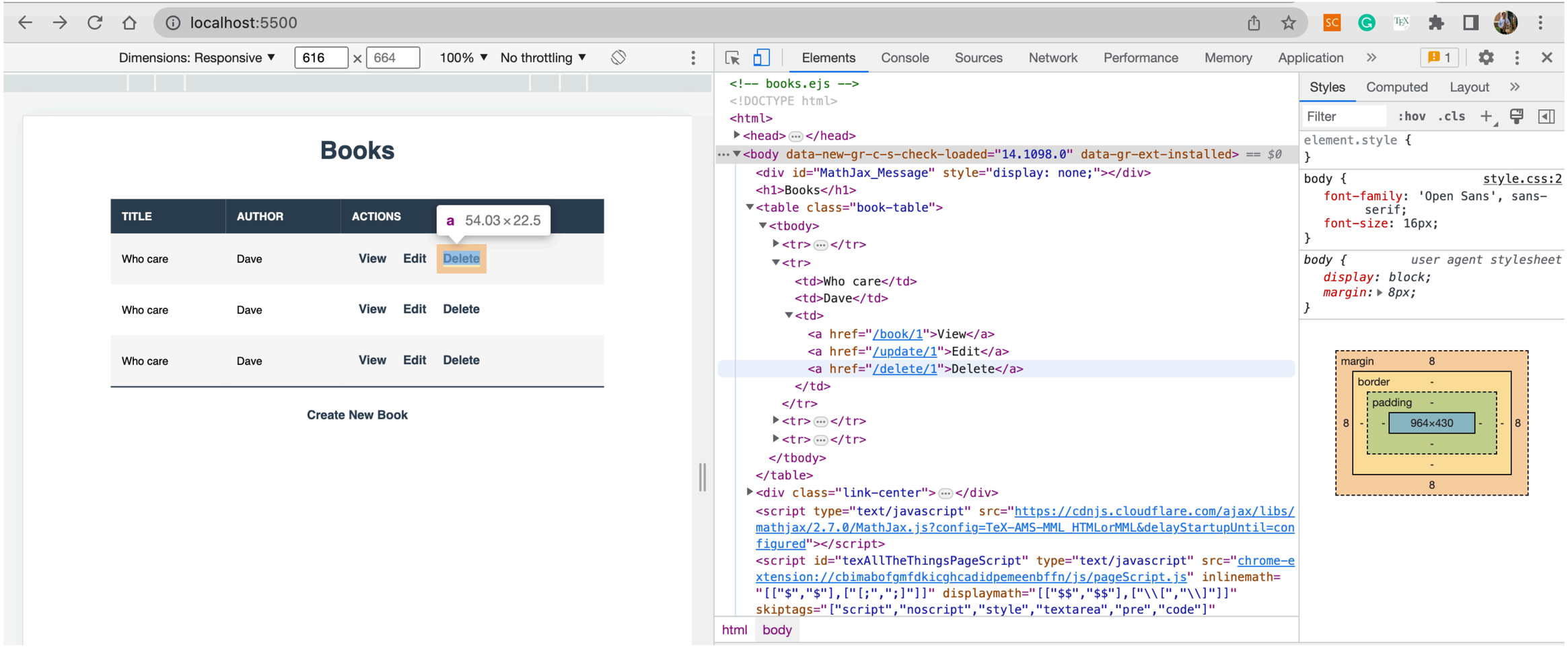
```
52  // Read all
53  app.get("/books", async (req, res) => {
54    try {
55      const books = await Book.find();
56      res.send(books);
57    } catch (error) {
58      res.status(500).send(error);
59    }
60  });
61
62  // Read one
63  app.get("/books/:id", async (req, res) => {
64    try {
65      const book = await Book.findOne({id:req.params.id});
66      res.send(book);
67    } catch (error) {
68      res.status(500).send(error);
69    }
70  });
71
```

```
71
72  // Update
73  app.put("/books/:id", async (req, res) => {
74    try {
75      const book = await Book.findOneAndUpdate({id:req.params.id}, req.body, {
76        new: true,
77      });
78      res.send(book);
79    } catch (error) {
80      res.status(500).send(error);
81    }
82  });
83
84  // Delete
85  app.delete("/books/:id", async (req, res) => {
86    try {
87      const book = await Book.findOneAndDelete({id:req.params.id});
88      res.send(book);
89    } catch (error) {
90      res.status(500).send(error);
91    }
92  });
```

```
93
94  // Start the server
95  const PORT = process.env.PORT || 3000;
96  app.listen(PORT, () => {
97    console.log(`Server started at http://localhost:${PORT}`);
98  });
99
```


POST
localhost:3000/books
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
1 {
2 "title": "By The Tree",
3 "author": "Kate"
4 }
Body
Cookies
Headers (7)
Test Results
Security
200 OK
30 ms
322 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "title": "By The Tree",
3 "author": "Kate",
4 "_id": "64057edcd6b2bdfc6e481b49",
5 "id": 3,
6 "__v": 0
7 }


Mongo Express
Database: test
Collection: books
Viewing Collection: books
New Document
New Index
Simple
Advanced
Key
Value
String
Find
Delete all 3 documents retrieved
_id
title
author
id
__v
64057eadd6b2bdfc6e481b43
Who care
Dave
1
0
64057ec0d6b2bdfc6e481b46
Blue Ride
Bob
2
0
64057edcd6b2bdfc6e481b49
By The Tree
Kate
3
0


localhost:5500
Dimensions: Responsive
616
664
100%
No throttling
Elements
Console
Sources
Network
Performance
Memory
Application
Books
TITLE
AUTHOR
ACTIONS
a 54.03 × 22.5
Who care
Dave
View
Edit
Delete
Who care
Dave
View
Edit
Delete
Who care
Dave
View
Edit
Delete
Create New Book
<!-- books.ejs -->
<!DOCTYPE html>
<html>
<head> </head>
<body data-new-gr-c-s-check-loaded="14.1098.0" data-gr-ext-installed> == $0

<h1>Books</h1>
<table class="book-table">
<tbody>
<tr> </tr>
<tr>
<td>Who care</td>
<td>Dave</td>
<td>
View
Edit
Delete
</td>
</tr>
<tr> </tr>
<tr> </tr>
</tbody>
</table>

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?config=TeX-AMS-MML_HTMLorMML&delayStartupUntil=configured"></script>
<script id="texAllTheThingsPageScript" type="text/javascript" src="chrome-extension://cbimabofgmfdkicghcadidpemeenbffn/js/pageScript.js" inlinemath="[["$","$"],["[;",";]"]]" displaymath="[["$$","$$"],["\\[","\\]"]]" skiptags="["script","noscript","style","textarea","pre","code"]"
html
body
Styles
Computed
Layout
Filter
:hov .cls
element.style {
}
body {
style.css:2
font-family: 'Open Sans', sans-serif;
font-size: 16px;
}
body {
user agent stylesheet
display: block;
margin: 8px;
}
margin
8
border
padding
964×430

THANK YOU

- CSS-Front-End
- PostgreSQL
- MongoDB
- Postman Test
- Front-End connect with Back-End