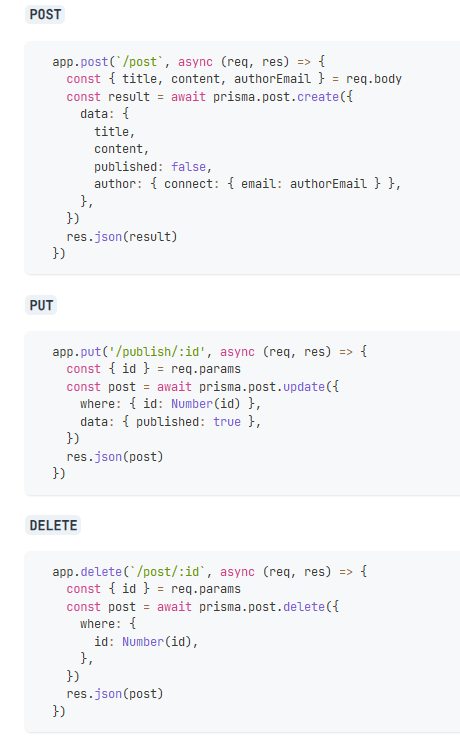
GET request

* localhost:3000/api/users

<https://www.prisma.io/docs/orm/overview/prisma-in-your-stack/rest>

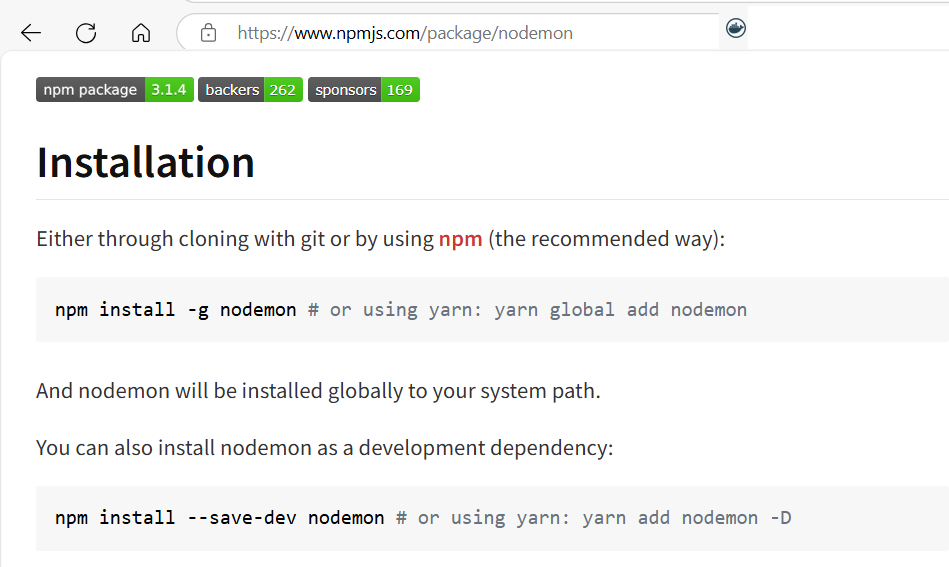
**REST API server example**



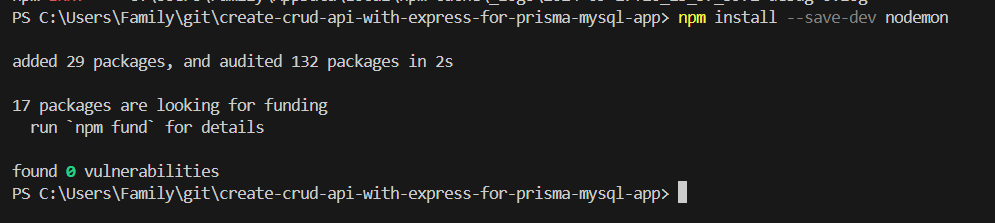


* Install nodemon dependency

<https://www.npmjs.com/package/nodemon>



C:\Users\Family\git\create-crud-api-with-express-for-prisma-mysql-app> npm install --save-dev nodemon



* Nodemon dependency was installed – OK

On package.json add:        
"dev": "nodemon --watch --exec node src/server.js",



package.json

{

  "name": "hello-prisma",

  "type": "module",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

     "dev": "nodemon --watch --exec node src/server.js",

    "test": "echo \"Error: no test specified\" && exit 1"

  },

Server.js

import { PrismaClient } from '@prisma/client';

import express from 'express';

const app = express();

const prisma = new PrismaClient();

// custom middleware

app.use((req, res, next) => {

    console.log(`${req.url} ${new Date()}`);

    next(); // call the next middleware in the stack

})

app.get('/',(req,res) => {

    res.send('HELLO WROLD')

})

// GET api/course

app.get('/api/users', async (req,res) => {

    // fetch all the users from db

    const allUsers = await prisma.user.findMany({

        include: {

          posts: true,

          profile: true,

        },

      })

    // get the prisma

    // get the course from the prisma object

    // call the many method

    // send back to user as a json

    res.status(200);

    res.json(allUsers);

});

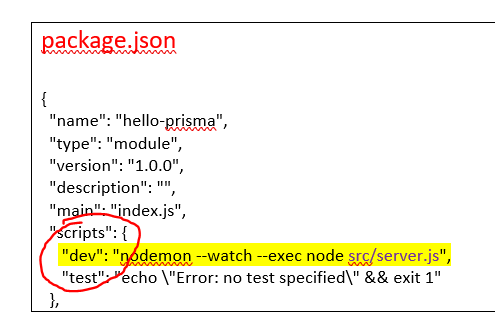
app.listen(3000, () => {

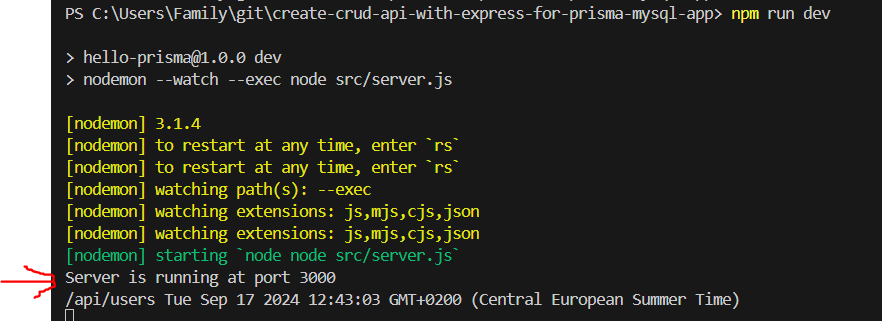
    console.log('Server is running at port 3000');

})

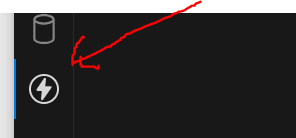
export default app;

* RUN SERVER…
* npm run dev





Thnder Client | http Client

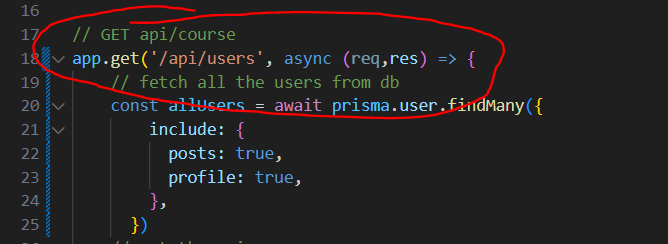


<https://www.prisma.io/docs/orm/overview/prisma-in-your-stack/rest>

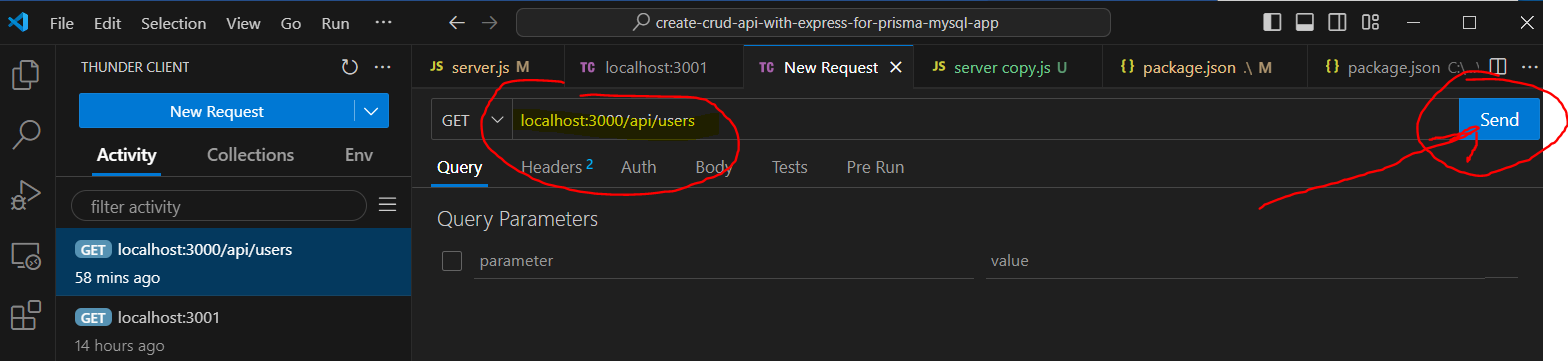
**REST API server example**

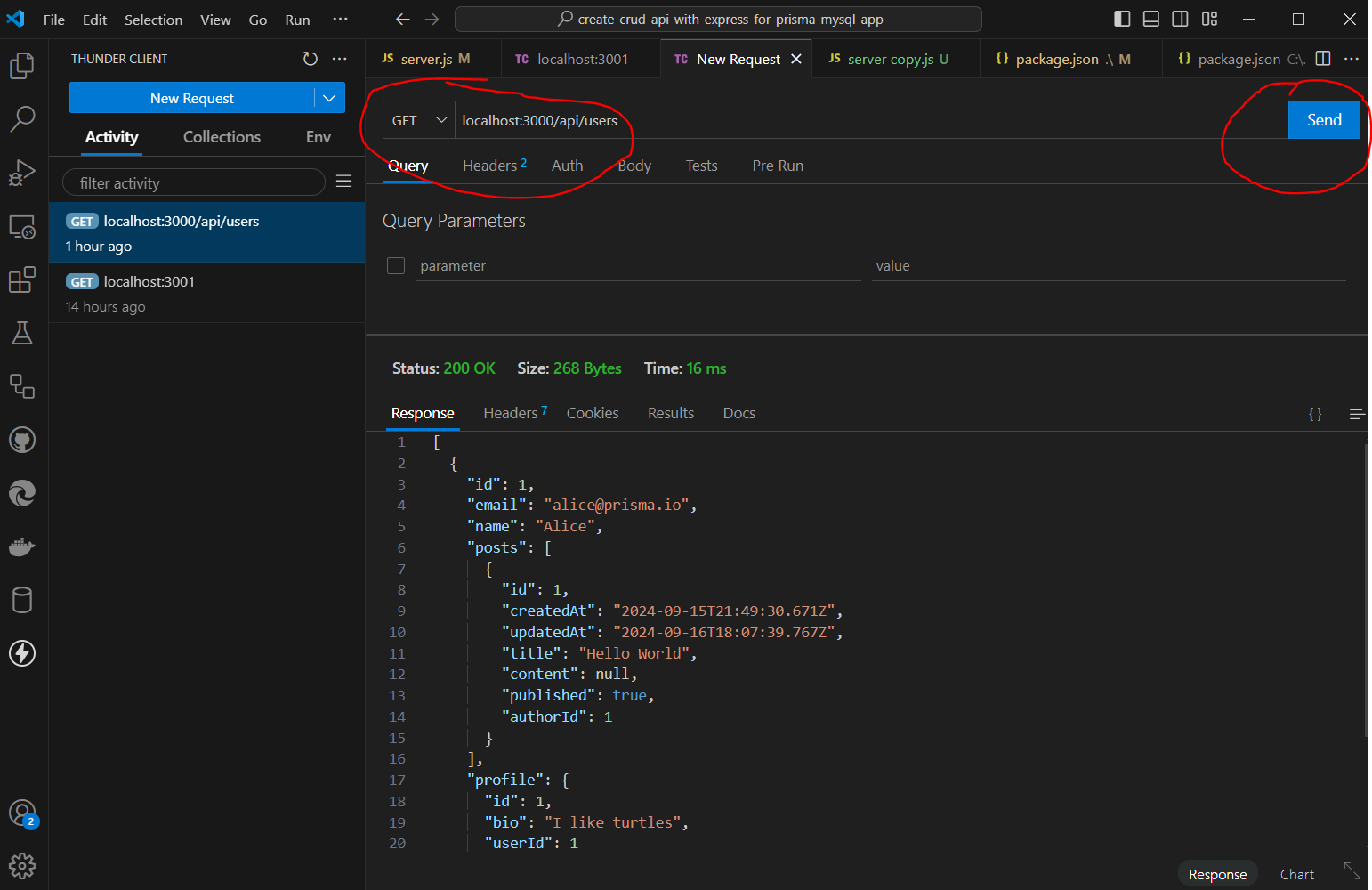


Server.js

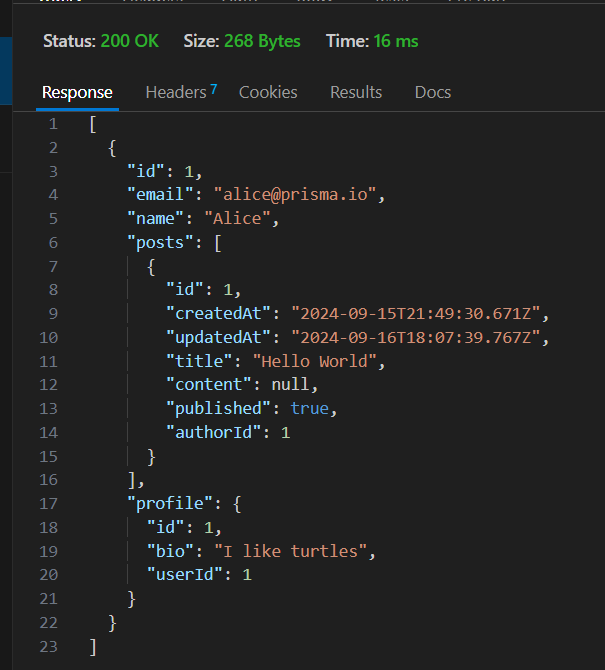


GET: localhost:3000/api/users 🡪 Click send button





Staus 200 OK 🡪 😊



[

{

"id": 1,

"email": "alice@prisma.io",

"name": "Alice",

"posts": [

{

"id": 1,

"createdAt": "2024-09-15T21:49:30.671Z",

"updatedAt": "2024-09-16T18:07:39.767Z",

"title": "Hello World",

"content": null,

"published": true,

"authorId": 1

}

],

"profile": {

"id": 1,

"bio": "I like turtles",

"userId": 1

}

}

]

GET: localhost:3000/api/users

Result ok 😊

GET Request OK 😊

POST request

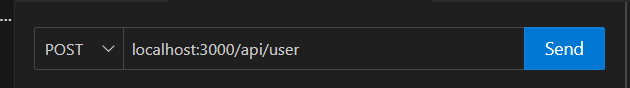
* localhost:3000/api/user

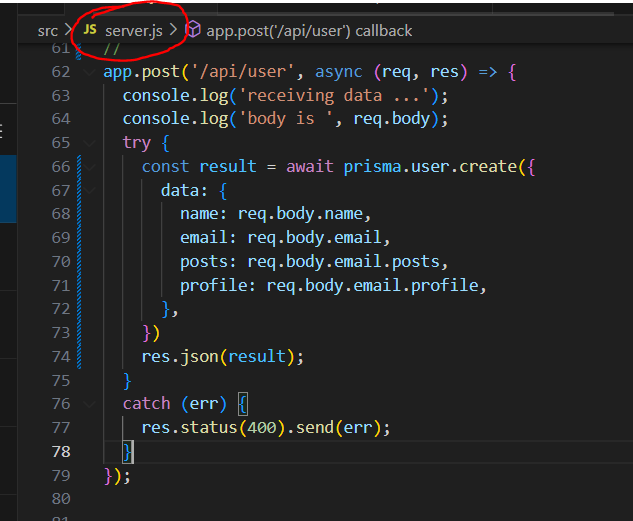
<https://www.prisma.io/docs/orm/overview/prisma-in-your-stack/rest>

**REST API server example**



Add new POST request to the http request and server.js as below





Server.js

app.post('/api/user', async (req, res) => {

  console.log('receiving data ...');

  console.log('body is ', req.body);

  try {

    const result = await prisma.user.create({

      data: {

        name: req.body.name,

        email: req.body.email,

        posts: req.body.email.posts,

        profile: req.body.email.profile,

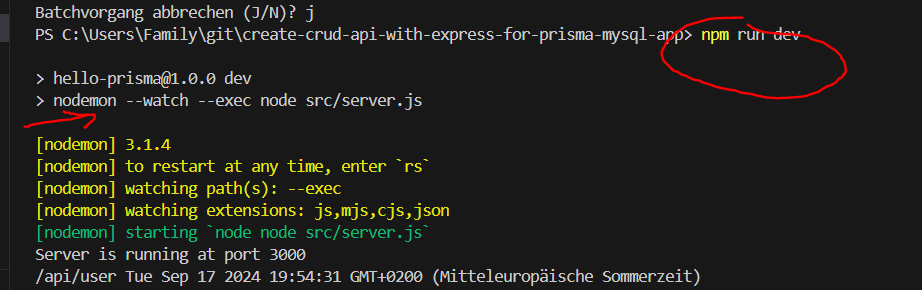
      },

    })

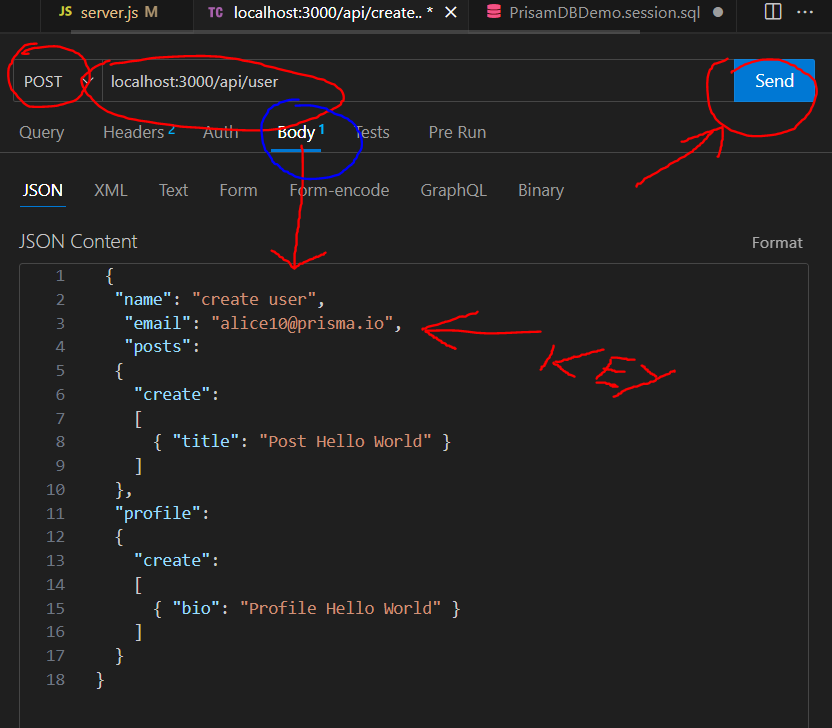
    res.json(result);

  }

RUN server… | npm run dev



Assign Body data to the new POST request as below:



Body - JSON Content   
  
{

"name": "create user",

"email": "alice10@prisma.io",

"posts":

{

"create":

[

{ "title": "Post Hello World" }

]

},

"profile":

{

"create":

[

{ "bio": "Profile Hello World" }

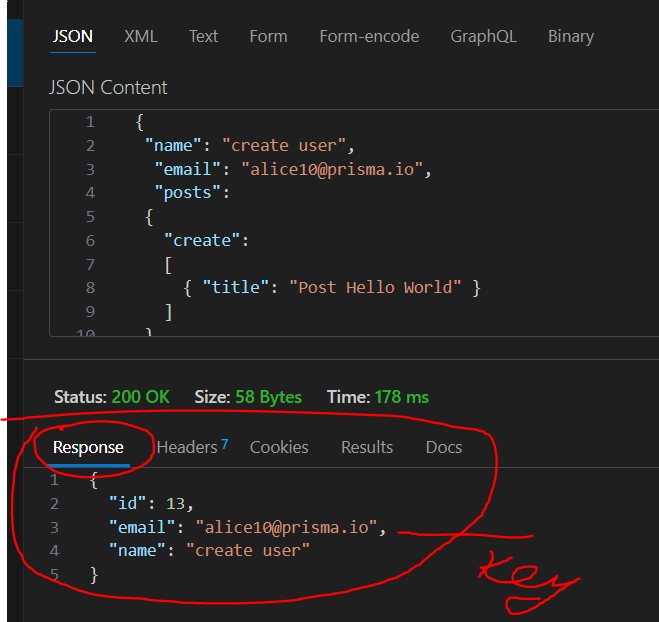
]

}

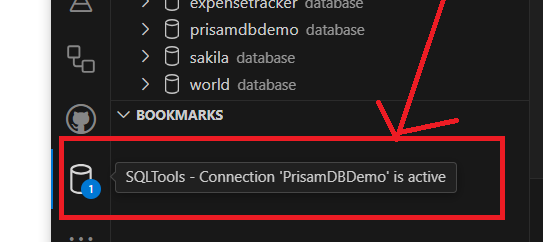
}

Click 🡪 Send button 🡪 Response 🡪 Status 200 OK 😊

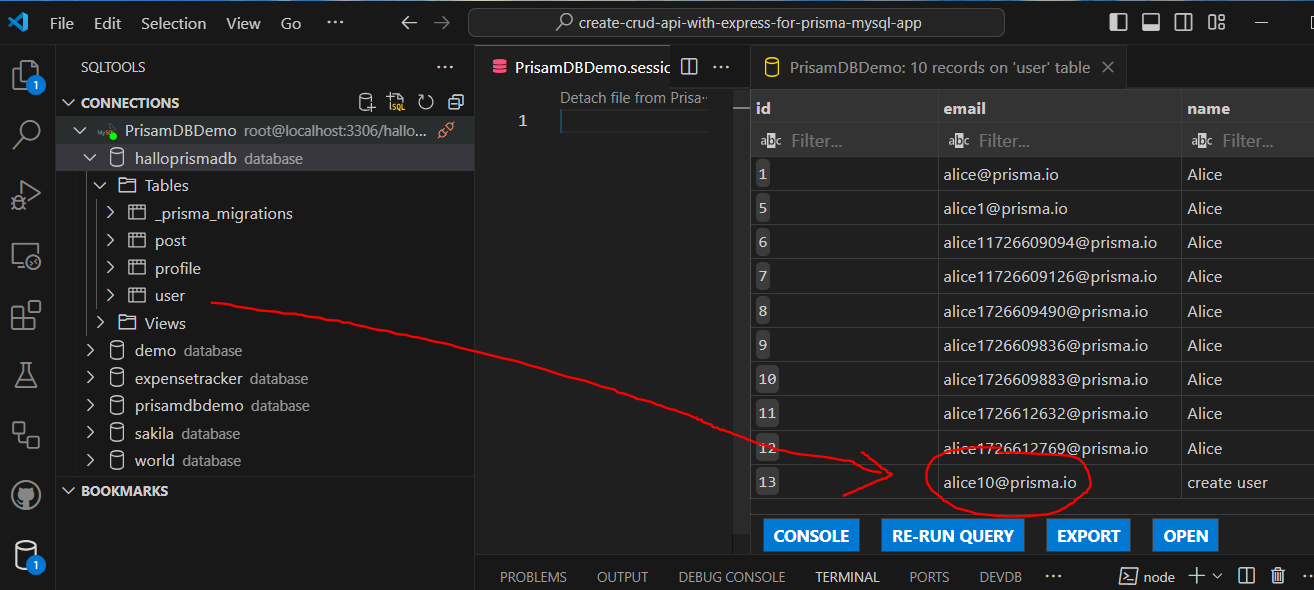
GET Response Result 🡪 OK



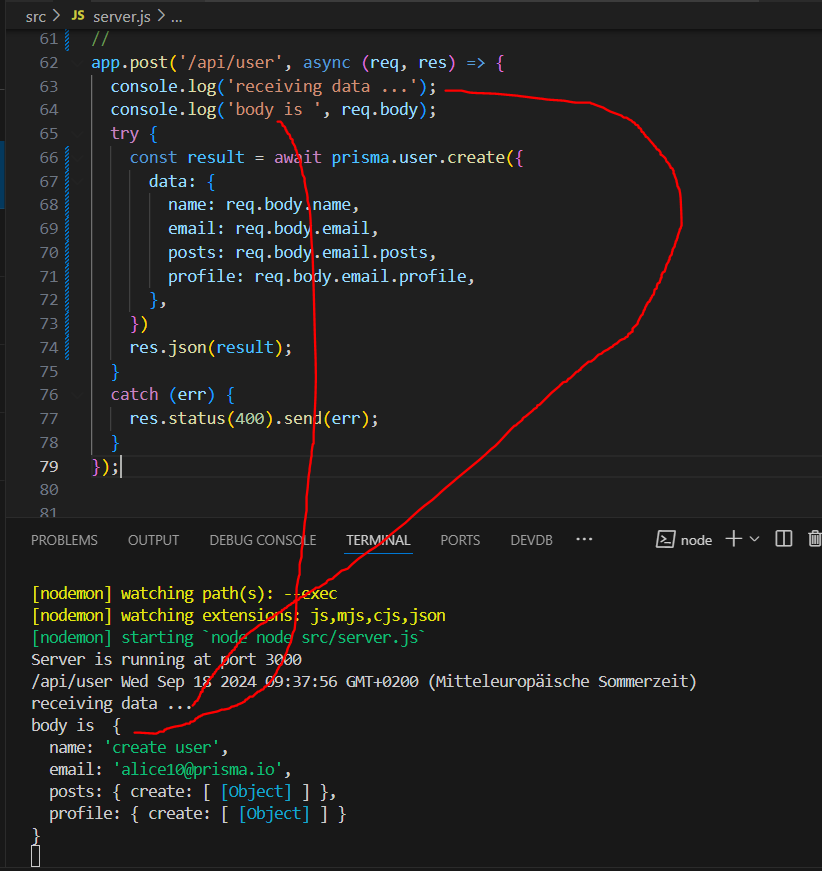
* Check MySQL database



Request data was inserted into database OK 😊



LOG http Request Body data OK 😊



POST Request OK 😊

<https://www.prisma.io/docs/orm/overview/prisma-in-your-stack/rest>

<https://www.prisma.io/docs/orm/overview/prisma-in-your-stack/rest>

