

LU11c - CRUD Server and Postman

Learning Objectives

1. How to connect the server to the db
2. How to fetch data from the db and display it in POSTMAN
3. How to perform the CRUD operations on the server
4. How to perform the CRUD operations by using PoSTMAN

Sources

- SampleServer that can perform CRUD

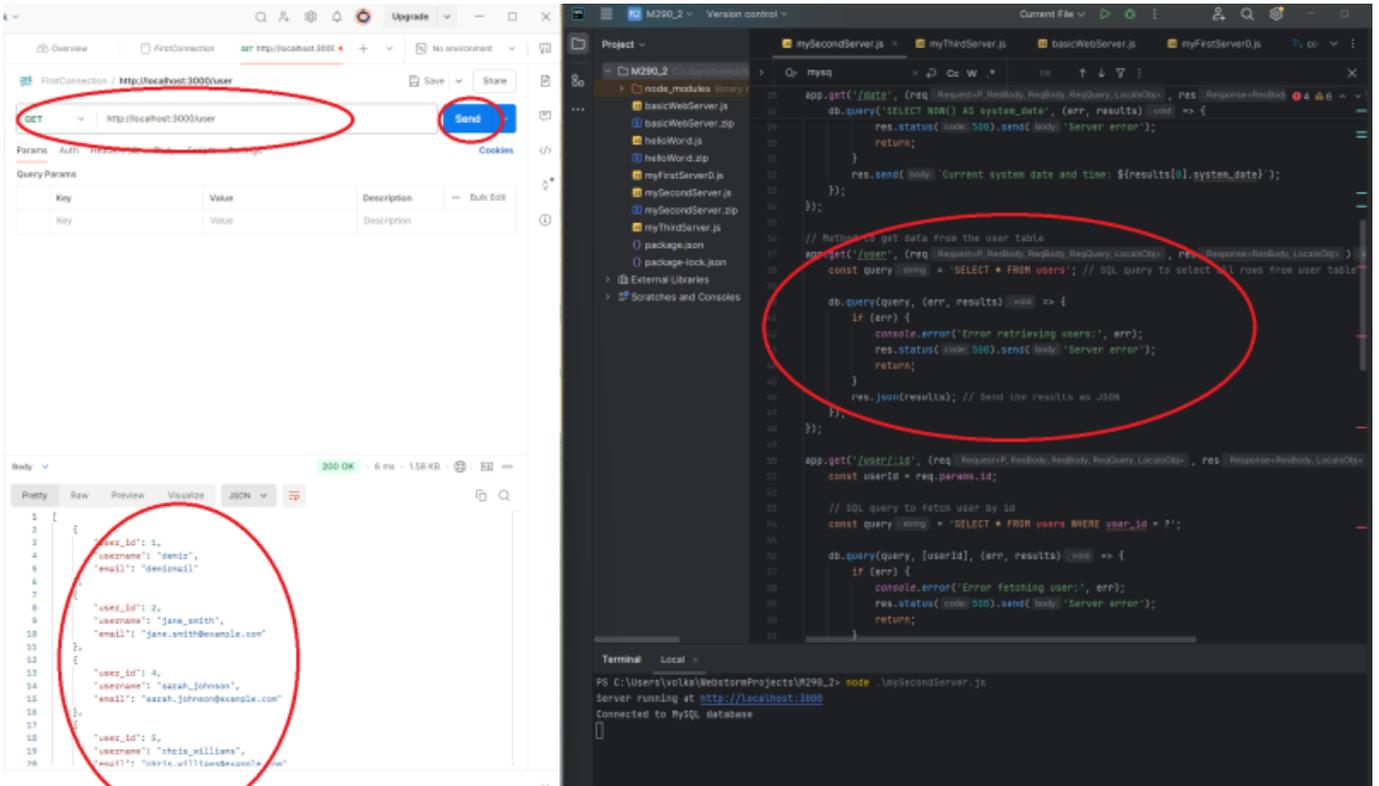
Introduction

At last chapter of our database journey we want to perform all CRUD operations within the server AND the client (POSTMAN). For reasons of efficiency we are going to start with the R = READ of CRUD.

R - READ: Fetching list of data

First of all, we want to display the list of users, which are store in the table users. The following code show how to get that list from the database.

```
// Method to get data from the user table
app.get('/user', (req, res) => {
  const query = 'SELECT * FROM users'; // SQL query to select all rows
  from user table
  db.query(query, (err, results) => {
    if (err) {
      console.error('Error retrieving users:', err);
      res.status(500).send('Server error');
      return;
    }
    res.json(results); // Send the results as JSON
  });
});
```



R - READ: Getting one specific row of data

If we want one specific individual to be displayed, we need to make some changes, such as filtering a `user_id`, as shown in the following code below.

```
app.get('/user/:id', (req, res) => {
  const userId = req.params.id;
  // SQL query to fetch user by id
  const query = 'SELECT * FROM users WHERE user_id = ?';
  db.query(query, [userId], (err, results) => {
    if (err) {
      console.error('Error fetching user:', err);
      res.status(500).send('Server error');
      return;
    }
    if (results.length === 0) {
      res.status(404).send('User not found');
    } else {
      res.status(200).json(results[0]);
    }
  });
});
```

The image shows a Postman REST client interface on the left and a VS Code editor on the right. In Postman, a GET request to `http://localhost:3000/user/4` is shown, with the response body containing a JSON object: `{ "user_id": 4, "username": "sarah_johnson", "email": "sarah.johnson@example.com" }`. In VS Code, the `mySecondServer.js` file contains the following code:

```
app.get('/user', (req, res) => {
  db.query(query, (err, results) => {
  });
});

app.get('/user/:id', (req, res) => {
  const userId = req.params.id;

  // SQL query to fetch user by id
  const query = `SELECT * FROM users WHERE user_id = ?`;

  db.query(query, [userId], (err, results) => {
    if (err) {
      console.error('Error fetching user:', err);
      res.status(500).send({ body: 'Server error' });
      return;
    }

    if (results.length === 0) {
      res.status(404).send({ body: 'User not found' });
    } else {
      res.status(200).json(results[0]);
    }
  });
});

// DELETE user method
app.delete('/user/:id', (req, res) => {
  const userId = req.params.id;

  const deleteQuery = `DELETE FROM users WHERE user_id = ?`;

  db.query(deleteQuery, [userId], (err, result) => {
    if (err) {
      res.status(500).send({ body: 'Error deleting user' });
    }
  });
});
```



From: <https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link: <https://wiki.bzz.ch/modul/m290/learningunits/lu09/theorie/03?rev=1730810599>

Last update: **2024/11/05 13:43**

