# Backend

## Kreiranje API-ja

Instalirati node pakete:

npm install express cors body-parser mysql

Kreirati početnu datoteku:

indeks.js

const express = require("express");

const cors = require("cors");

const bodyParser = require("body-parser");

const mysql = require("mysql");

const app = express();

const port = 3000;

// Parser za JSON podatke

app.use(bodyParser.json());

// Parser za podatke iz formi

app.use(bodyParser.urlencoded({ extended: true }));

const connection = mysql.createConnection({

    host: host

    user: user,

    password: password,

    database: database

  });

app.use(express.urlencoded({ extended: true }));

connection.connect(function(err) {

    if (err) throw err;

    console.log("Connected!");

  });

  app.listen(port, () => {

    console.log("Server running at port: " + port);

});

## Popis API-ja:

|  |  |  |  |
| --- | --- | --- | --- |
| **URL** | **HTTP** | **Result** | **SQL** |
| /api/knjige | GET | Lista svih knjiga | SELECT \* FROM knjiga |
| /api/knjige/naslov/:naslov | GET | Lista knjiga po naslovu | SELECT \* FROM knjiga WHERE naslov like "%naziv%" |
| /api/knjige/autor/:autor | GET | Lista knjiga po autoru | SELECT \* FROM knjiga WHERE autor like "%autor%" |
| /api/slob\_knjige/ | GET | Lista slobodnih knjiga | SELECT (knjiga.stanje - count(rezervacija.knjiga)) as slobodne, knjiga.id, knjiga.naslov, knjiga.stanje FROM `knjiga` left join rezervacija on knjiga.id=rezervacija.knjiga group by knjiga.id |
| /api/slob\_knjige/:id\_knjige | GET | Provjera ako je knjiga slobodna |  |
| /api/rezerv\_knjige | GET | Lista rezerviranih knjiga | SELECT \* FROM knjiga, rezervacija WHERE knjiga.id=rezervacija.knjiga |
| /api/rezerv\_knjige\_korisnici | GET | Lista rezerviranih knjiga s korisnicima | SELECT \* FROM knjiga, rezervacija, korisnik WHERE knjiga.id=rezervacija.knjiga and korisnik.id=rezervacija.korisnik |
| /api/rezerv\_knjige/korisnik/:id\_korisnik | GET | Lista rezerviranih knjiga za korisnika id\_korisnik | SELECT \* FROM knjiga, rezervacija, korisnik WHERE knjiga.id=rezervacija.knjiga and korisnik.id=rezervacija.korisnik AND korisnik.id=id\_korisnik |
| /api/rezerv\_knjige/knjiga/:id\_knjiga | GET | Lista rezervacija za knjigu id\_knjiga | SELECT \* FROM knjiga, rezervacija, korisnik WHERE knjiga.id=rezervacija.knjiga and korisnik.id=rezervacija.korisnik AND knjiga.id=id\_knjiga |
| /api/korisnici | GET | Lista svih korisnika |  |
| /api/korisnici/:id\_korisnik | GET | Dohvati jednog korisnika po id\_korisnik |  |
| /api/korisnici/:id\_korisnik | PUT | Ažuriraj podatke o jednom korisniku | UPDATE… |
| /api/rezerv\_knjige/:id\_knjige | POST | Unesi rezervaciju knjige za korisnika | INSERT INTO rezervacija (datum, knjiga, korisnik) VALUES ('2024-10-31', 3, 2) |
| /api/rezerv\_knjige/:id\_knjiga | DELETE | Brisanje rezervacije za knjigu | DELETE… |
|  |  |  |  |

U indeks.js dodati prikladne CRUD operacije:

app.get("/api/knjige", (req, res) => {

  connection.query("SELECT \* FROM knjige", (error, results) => {

    if (error) throw error;

    res.send(results);

  });

});

app.post("/api/rezerv\_knjige", (req, res) => {

  const data = req.body;

  rezervacija = [[date.today, data.id\_knjiga, data.id\_korisnik]]

  connection.query("INSERT INTO rezervacija (datum, knjiga, korisnik) VALUES ?", [rezervacija], (error, results) => {

    if (error) throw error;

    res.send(results);

  });

});

## ZADACI:

1. Napraviti SQL naredbu kojom se provjerava koliko je knjiga kod odabranog korisnika (id\_korisnik) – vraća se ukupan broj knjiga kod korisnika

[SELECT](http://student.veleri.hr/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [count](http://student.veleri.hr/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(\*) FROM `rezervacija` WHERE korisnik=2

1. Napraviti SQL naredbu kojom se provjerava ima li koji slobodan primjerak knjige za id\_knjige – vratiti broj slobodnih primjeraka knjige za npr. id knjige=5

SELECT knjiga.stanje AS stanje,count(rezervacija.id) AS rezervirano FROM knjiga, rezervacija WHERE knjiga.id=rezervacija.knjiga AND knjiga=11

1. Napraviti SQL naredbu kojom se vraća broj rezerviranih primjeraka knjige za id\_knjige – vratiti broj rezerviranih primjeraka knjige za npr. id\_knjige=5
2. Napraviti SQL naredbu za dohvat svih korisnika kod kojih se nalazi knjiga s id\_knjige (npr.id\_knjige=5)
3. Napraviti SQL naredbu kojom se može dobiti ukupan broj svih primjeraka svih knjiga u knjižnici
4. Napraviti SQL naredbu kojom se može dobiti ukupan broj svih rezerviranih primjeraka u knjižnici
5. Napraviti SQL naredbu kojom se može dobiti ukupan broj svih slobodnih primjeraka u knjižnici (zanima nas prostorno zauzeće knjiga)
6. Napraviti SQL naredbu za ispis knjiga koje imaju manje od 3 primjerka.
7. Napraviti SQL naredbu koja vraća sve korisnike koji imaju rezervirane knjige duže od mjesec dana. I vratiti popis tih primjeraka knjiga.
8. U bazu dodati uz korisnika podatke: email i broj telefona
9. Napraviti SQL naredbu koja vraća e-mail, broj telefona, naslov knjige i datum rezervacije – za korisnike koji imaju rezervirane knjige duže od mjesec dana, kako bi se korisniku poslao mail i SMS poruka s napomenom da mora vratiti knjigu k koja je preuzeta datuma d.
10. Napraviti SQL naredbu kojom se provjerava ima li korisnik rezervirana dva ili više primjeraka iste knjige.
11. Napraviti SQL naredbu za izmjenu podataka o korisniku.
12. Na koji način bismo mogli nadopuniti bazu tako da možemo pratiti rezervacije kroz određeno vrijeme? Odnosno trebamo mogućnost da nam se ne brišu podaci o rezervacijama, kako bismo pratili statistiku zauzetosti pojedinih knjiga i eventualno naručili više primjeraka.