Implementacija Node.JS REST sučelja i MySQL-a

## Korak 1 – Kreiranje nove aplikacije/projekta

Kreiramo novi direktorij za naš projekt i pozicioniramo se:

c:\>mkdir node-mysql-rest && cd node-mysql-rest

Zatim inicijaliziramo Node projekt:

npm init --yes

Nakon toga instaliramo sljedeće zavisnosti/dependency:

npm install express mysql body-parser cors –save

Sada package.json datoteka izgleda kao:

{

"name": "mysql-rest",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

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

},

"keywords": [],

"author": "",

"license": "ISC",

"dependencies": {

"body-parser": "^1.20.0",

"cors": "^2.8.5",

"express": "^4.17.3",

"mysql": "^2.18.1"

}

}

## Korak 2 – Spajanje aplikacije na MySQL bazu podataka

Konekcija na bazu :

// connection configurations

var dbConn = mysql.createConnection({

host: 'localhost',

user: 'root',

password: 'ikrasic',

database: 'fsre'

});

// connect to database

dbConn.connect();

(u vašem okruženju prilagodite/promijenite podatke potrebne za spajanje, ime baze/tablice(korisničko ime /lozikna)

## Korak 3 – Kreiranje REST sučelja

Za potrebe jednostavnog sučelja uradite ćemo sljedeći *slim* server app.js naredbom *node app.js*

*(kompletan kod je dostupan u app.zip datoteci, promijenite nazive zablice/baze, korisničko ime ili lozinku kako je već konfigurirano u vašem lokalnom okruženju )*

c:\node-mysql-rest>node app.js

Node MySQL REST API app is running on port 3000

Pokrenimo primjer i testirajmo (putem browsera-a ili POSTMAN-a ili sličnog alata)!

Sljedeća tablica prikazuje REST API sučelje koje je izloženo u našem slučaju:

|  |  |  |
| --- | --- | --- |
| **HTTP Metoda** | **URL** | **Primjer body-a i opis akcije** |
| GET | /employees | Dohvat svih employee zapisa |
| GET | /employee/:id | Dohvat pojedinačnog employee zapisa po id-u (kod nas u bazi je to emp\_id) |
| PUT | /employee | {         "employee": {          "emp\_no": 10041,          "birth\_date": "1959-08-26 23:00:00.000",          "first\_name": "Urikas",          "last\_name": "Lenartx",          "gender": "F",          "hire\_date": "1995-11-11 23:00:00.000"      }  } |
| POST | /employee | {         "employee": {          "birth\_date": "1959-08-26 23:00:00.000",          "first\_name": "Marko",          "last\_name": "Marković",          "gender": "M",          "hire\_date": "1995-11-11 23:00:00.000"      }  } |
| DELETE | /employee/:id | Brisanje pojedinačnog employee zapisa po id-u (kod nas u bazi je to emp\_id) |
| DELETE | /employees | Brisanje svih employees zapisa - nije implementirano |

PUT akciju možemo realizirati i kao akciju na resursu /employee/:id –što je ispravno/bolje ili razlike općenito?

