

Uvod u Mongoose ODM

Zlatko Hajdarevic

20. jun 2024.

Sadržaj

Uvod u Mongoose ODM

Kreiranje Mongoose modela

Izrada CRUD funkcionalnosti

Filtriranje podataka

Sortiranje podataka

Selekcija željenih polja iz BSON strukture

Ograničavanje broja vraćenih zapisa

Uvod u Mongoose ODM

- ▶ Mongoose je Object Data Modeling (ODM) biblioteka za MongoDB i Node.js.
- ▶ Pruža način za definisanje šema za dokumente u MongoDB kolekciji.
- ▶ Omogućava validaciju, transformaciju i poslovnu logiku direktno na modelima podataka.
- ▶ Podržava bogat API za kreiranje, čitanje, ažuriranje i brisanje (CRUD) operacija.
- ▶ Omogućava lakšu integraciju sa Node.js aplikacijama, pružajući strukturu i sigurnost u radu sa podacima.
- ▶ Pruža podršku za populaciju dokumenata, omogućavajući efikasno rukovanje referenciranim podacima.
- ▶ Uključuje mogućnosti za middleware, koji omogućavaju izvršavanje koda pre i posle određenih operacija na dokumentima.
- ▶ Može se integrisati sa drugim bibliotekama i framework-ovima kao što su Express i Koa za izradu kompletnih web aplikacija.

Kreiranje Mongoose modela

- ▶ Definisanje šeme:

- ▶ `const mongoose = require('mongoose');`
 - ▶ `const Schema = mongoose.Schema;`
 - ▶ `const userSchema = new Schema({ name: String, age: Number, email: String });`

- ▶ Kreiranje modela:

- ▶ `const User = mongoose.model('User', userSchema);`

Izrada CRUD funkcionalnosti

► Create: Dodavanje novog dokumenta

- `const user = new User({ name: 'John', age: 30, email: 'john@example.com' });`
- `user.save();`

► Read: Čitanje dokumenata

- `User.find({ age: 30 }, (err, users) => { console.log(users); });`

► Update: Ažuriranje dokumenta

- `User.updateOne({ name: 'John' }, { age: 31 }, (err, res) => { console.log(res); });`

► Delete: Brisanje dokumenta

- `User.deleteOne({ name: 'John' }, (err) => { console.log('Deleted'); });`

Filtriranje podataka

- ▶ Koristeći Mongoose, podaci se mogu filtrirati prema određenim kriterijumima.
- ▶ Primer: Pronaći sve korisnike starije od 25 godina.
 - ▶ `User.find({ age: { $gt: 25 } }, (err, users) => { console.log(users); });`

Sortiranje podataka

- ▶ Mongoose omogućava sortiranje podataka prema određenim poljima.
- ▶ Primer: Sortiranje korisnika po imenu u rastućem redosledu.
 - ▶ `User.find().sort({ name: 1 }).exec((err, users) => { console.log(users); });`

Ograničavanje broja vraćenih zapisa

- ▶ Mongoose omogućava ograničavanje broja rezultata koji će biti vraćeni iz upita.
- ▶ Primer: Vratiti samo prva tri korisnika.
- ▶ `User.find().limit(3).exec((err, users) => { console.log(users); });`