



**Programiranje internet aplikacija**

*Elektrotehnički fakultet, Univerzitet u Beogradu*

**2025/2026**

# Uvod

- Deo MEAN steka – *open-source* JavaScript softverski stek

- M – MongoDB
- E – Express
- A – Angular
- N – Node.js

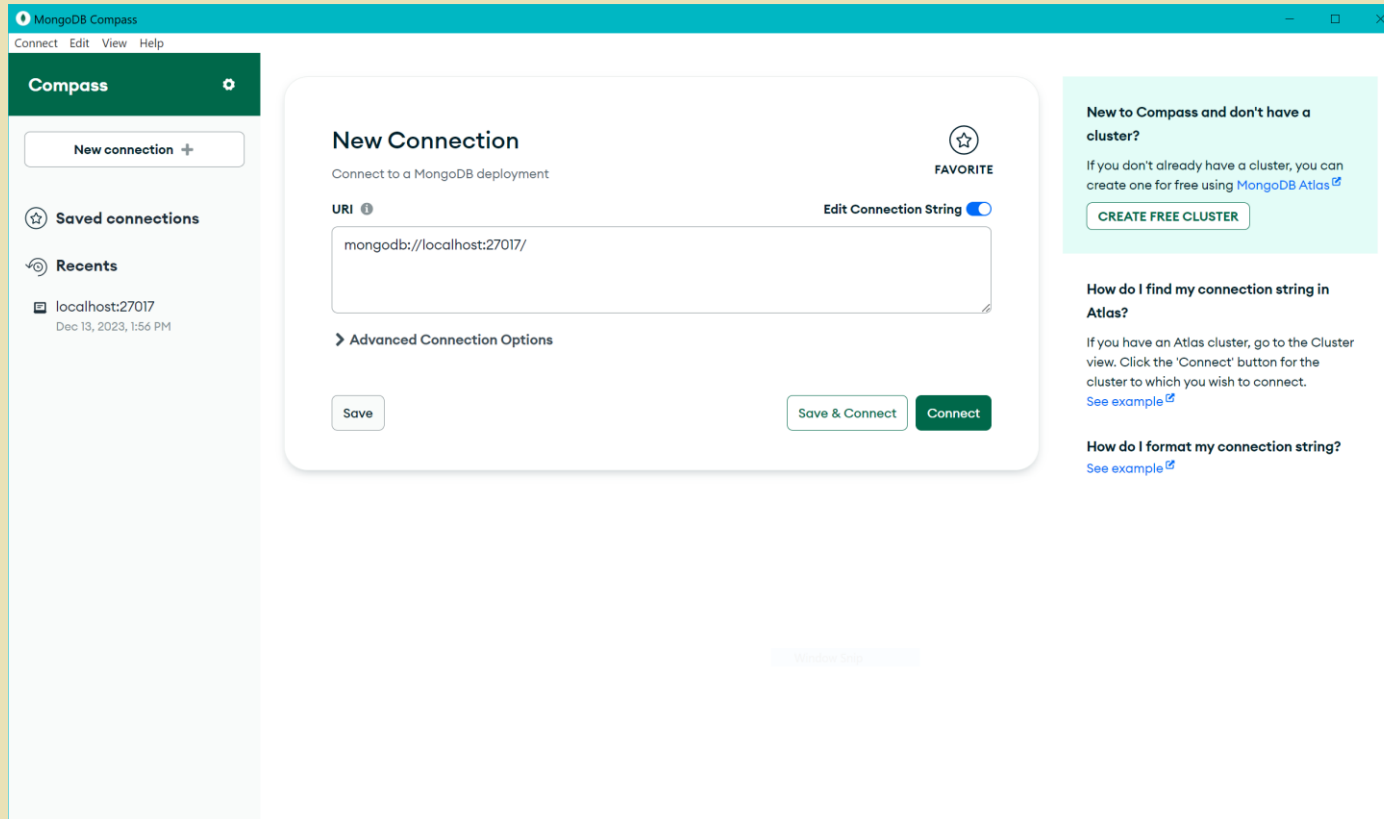




# MongoDB – 8.2.1

- <https://www.mongodb.com/try/download/community-edition/releases/archive>
  - Nerelaciona baza podataka – NoSQL
  - Bazirana na „dokumentima“ – *JSON-like* format
  - Glavne karakteristike
    - Indeksiranje
    - Agregacija (map-reduce)
    - Transakcije
    - Podržava kolekcije fiksne kapaciteta
- 
- 

# MongoDB Compass



- GUI alat za MongoDB
- MongoDB mora biti pokrenut u pozadini
- Podržava i pristup bazama i kolekcijama putem terminala
- Verzija u laboratoriji 1.47.1

# Mongoose

- „elegant mongodb object modeling for node.js“

```
import mongoose from 'mongoose'
```

```
mongoose.connect('mongodb://127.0.0.1:27017/baza')
```

```
const connection = mongoose.connection
connection.once('open', () => {
  console.log('db connection ok')
})
```

# Modeli - šeme

```
// user.ts
import mongoose from 'mongoose'
```

```
const Schema = mongoose.Schema;
```

```
let User = new Schema({
  username: {
    type: String
  },
  password: String,
  type: {
    type: Number, min: 0, max: 1
  }
})
```

```
export default mongoose.model('UserModel', User, 'korisnici')
```

naziv modela

šema

naziv kolekcije u bazi

# Upiti – pretraga

```
import UserModel from '...';
```

```
UserModel.findOne({ 'username': username, 'password': password }).then((user) => {  
    res.json(user)  
}).catch((err) => {  
    console.log(err);  
})
```

filter

update

```
UserModel.findOneAndUpdate({ 'username': username }, { 'type' : 0 } )  
    .then((user) => {  
        res.json({ 'message': 'ok' })  
    }).catch((err) => {  
        console.log(err);  
        res.json({ 'message': 'err' })  
    })
```

# Upiti – pretraga

```
UserModel.find( { 'type' : 0 } )  
  .then((users) => {  
    res.json(users)  
  }).catch((err) => {  
    console.log(err);  
  }) // nalazi sve dokumente koji odgovaraju filteru; vraća niz
```

```
UserModel.find({})  
  .then((all) => {  
    res.json(all)  
  }).catch((err) => {  
    console.log(err);  
  }) // vraća niz svih dokumenata iz kolekcije
```



# Upiti – unos

```
register = (req: express.Request, res: express.Response) => {  
  let user = new UserModel({  
    username: req.body.username,  
    password: req.body.password,  
    type: req.body.type  
  })  
  
  //let user = new UserModel(req.body)  
  
  user.save().then((ok) => {  
    res.json({ "message": "ok" })  
  }).catch((err) => {  
    console.log(err);  
    res.status(400).json({ "message": "error" })  
  })  
}
```

# Upiti – brisanje

```
UserModel.deleteOne({ 'username': username })  
  .then((ok) => {  
    res.json({ "message": "ok" })  
  }).catch((err) => {  
    console.log(err);  
    res.json({ "message": "error" })  
  })
```

# Upiti – ažuriranje

```
// neka šema sadrži polje nizovnog tipa - list: Array
```

```
let sth = ...; // može biti „prost“ tip ili objekat sa više polja
UserModel.updateOne({'username': username }, {$push: {'list': sth}})
  .then((ok) => {
    res.json({ "message": "ok" })
  }).catch((err) => {
    console.log(err);
    res.json({ "message": "error" })
  })
```

```
UserModel.updateOne({'username': username }, {'type': 'radnik'})
  .then((ok) => {
    res.json({ "message": "ok" })
  }).catch((err) => {
    console.log(err);
    res.json({ "message": "error" })
  })
```



HVALA NA PAŽNJI!

