wa-final: Službeni šalabahter

Mongo CRUD operacije

```
collection = db.collection("naziv_kolekcije");
collection.findOne({_id : ObjectId("id")}); // traži po id-u
collection.find(filter).toArray();
collection.find(filter, projection).toArray();
collection.findOne(filter, projection)
collection.insertOne(document);
collection.insertMany([document1, document2, ...]);
collection.updateOne(filter, update)
collection.updateMany(filter, update)
collection.deleteOne(filter)
collection.deleteMany(filter)
```

Mongo operatori

```
{$set : {key: value}} // postavlja vrijednost ključa
{$unset: {key: value}} // briše ključ
{$inc: {key:value}} // inkrementira vrijednost ključa
{$mul: {key: value}} // množi vrijednost ključa
{$push: {key:value}} // ako je key polje, dodaje u njega vrijednost value
{$exists : {key: true}} // podudara elemente gdje postoji ključ
{key: {$eq: value}} // podudara elemente gdje je vrijednost ključa jednaka value
- varijante: $ne, $gt, $gte, $lt, $lte
{key : {$in: [value1, value2]}} // podudara elemente gdje je vrijednost ključa unutar
danog polja
- varijanta: $nin
{$and: [{key1: value1}, {key2: value2}]} // logičko i za spajanje uvjeta
{$or: [{key1: value1}, {key2: value2}]} // logičko ili za spajanje uvjeta
{$not: {key: value}} // negacija uvjeta
```

Mongo primjeri upita

```
collection.find({key: value}).toArray(); // svi elementi gdje je vrijednost ključa
jednaka value
collection.find({key: {$gt: value}}).toArray(); // svi elementi gdje su vrijednosti
ključa veće od value
collection.find({$and: [{ age: { $gte: 25 } }, { status: "active" }]}) // primjer 2
uvjeta
```

Middleware funkcije

```
const middleware_fn = (req, res, next) => {
  // kod koji se izvršava
  return next();
}
app.METHOD("/", [middleware_fn, middleware_fn_2, middleware_fn_n...], (req, res),
callback_fn); // na razini rute
app.use(middleware_fn); // na razini aplikacije app
```

Express-Validator

```
import { body, query, param, check, validationResult } from 'express-validator';
body, query, param ili check('key').METODA // sintaksa
query('key').isInt() // primjer
body('email').notEmpty().isEmail() // primjer lanca
validationResult(req) // vraća ukupni rezultat validacije
check('key').withMessage('Custom poruka greške')

// primjer upotrebe
const errors = validationResult(req);
  if (errors.isEmpty()) {} // ako nema greške...
```

Express-Validator česte provjere

```
notEmpty(), isEmpty(), exists()
isLength({min: value, max:value})
isInt(), isFloat(), isNumeric(), isAlpha(), isAlphanumeric()
contains(value), equals(value)
isIn([value1, value2, ...]), isNotIn([value1, value2, ...])
trim(), escape(), isArray(), isObject()
//primjer
app.METHOD("/", [check('key').isInt(), check('key2').isString()], (req, res),
callback_fn);
```

Bcrypt

```
let hash = await bcrypt.hash(plain_text, saltRounds); // hashiranje
let validate = await bcrypt.compare(plain_text, hashed_value); // usporedba
```

jsonwebtoken

```
secret = process.env.JWT_SECRET;
let token = jwt.sign(payload, secret, {expiresIn: '1h'}); // generiranje tokena
let decoded = jwt.verify(token, secret); // dekodiranje tokena
```