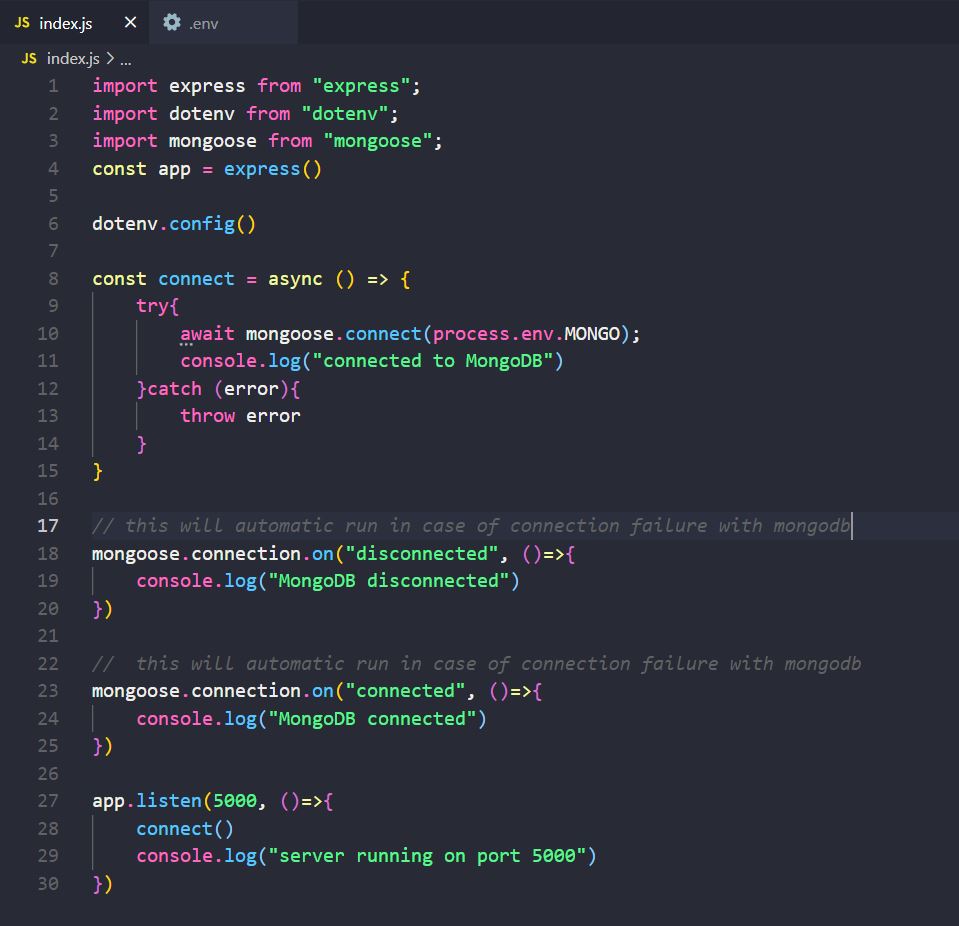
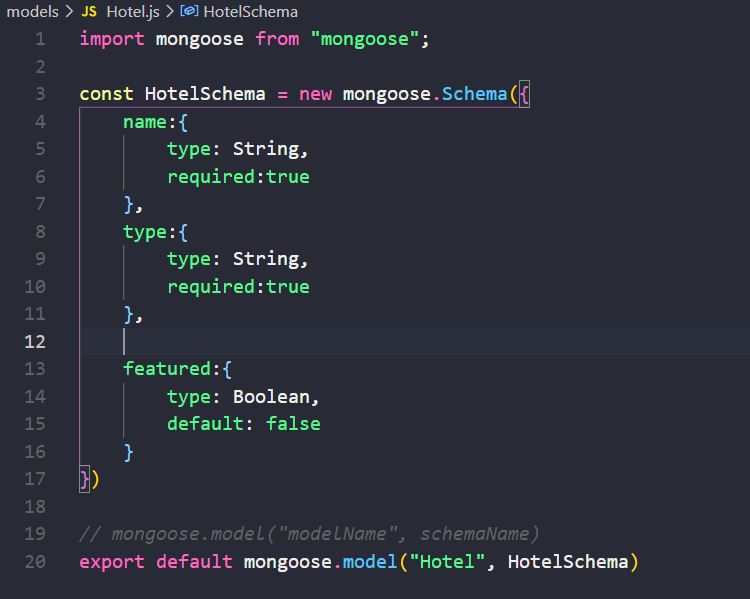
**Backend Development**

**Connection with mongodb:**

****

* dotenv is used to hide secrete data like id, password…

**Schema and model :**

****

**Send data to Mongodb using postman**

Json can’t be sent to express server, to send json data use middleware

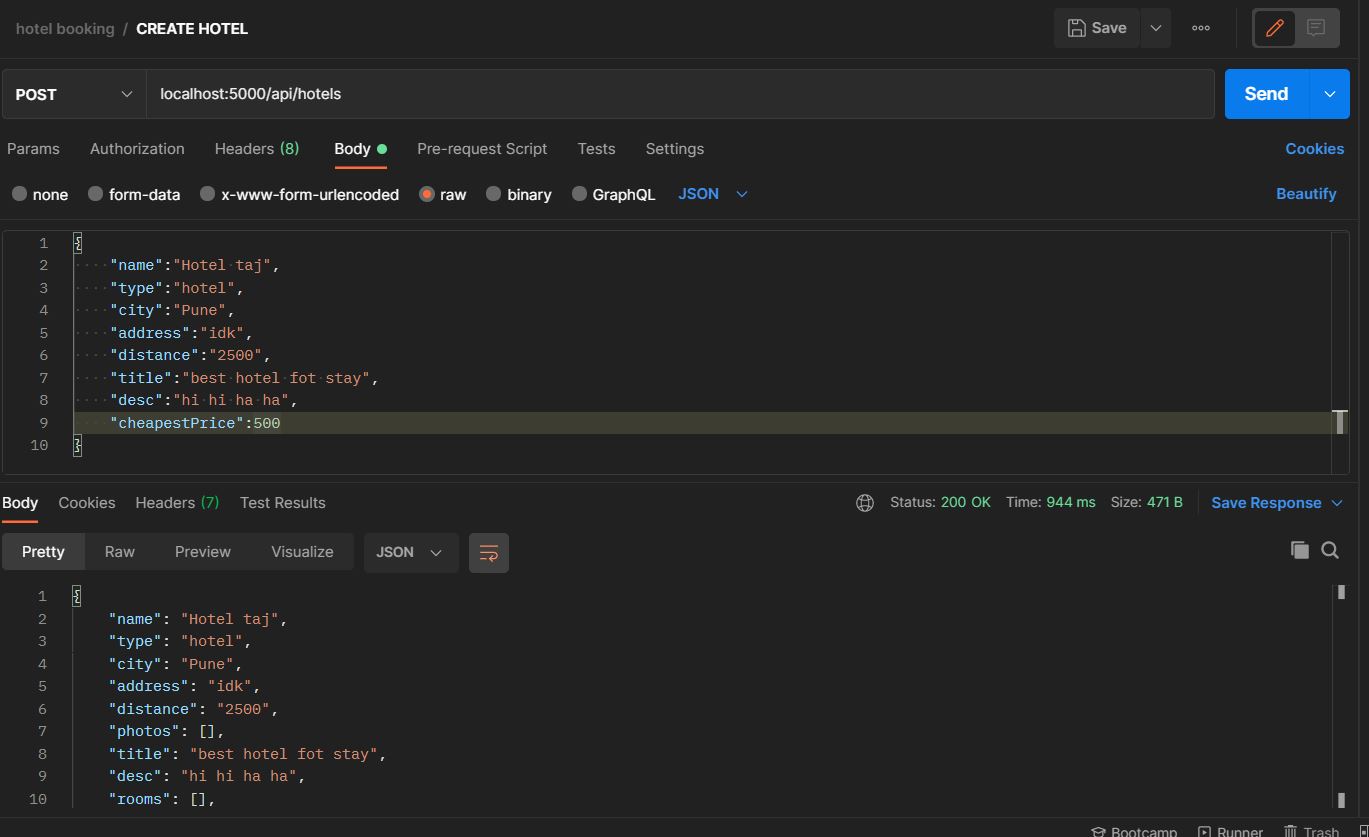
* app.use(express.json())



**CREATE in MongoDB or Send data**

****

**Sending through postman**

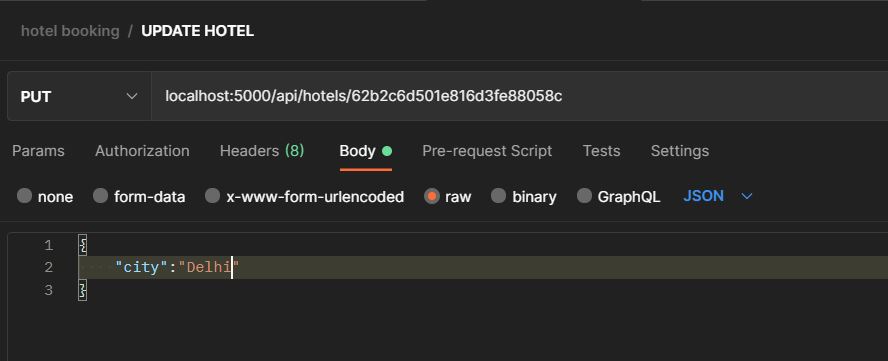


Update in mongodb

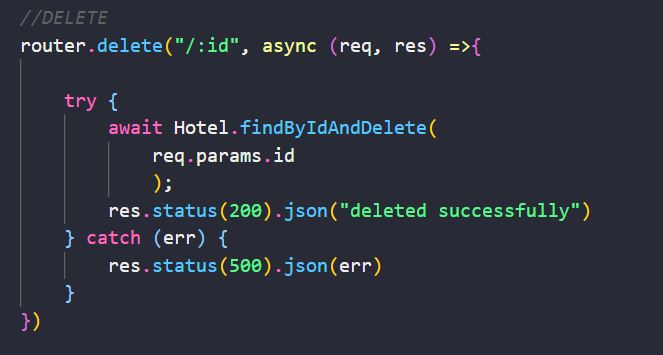


Note: if you don’t write {new:true} it will not affect database it is only return new copy to console

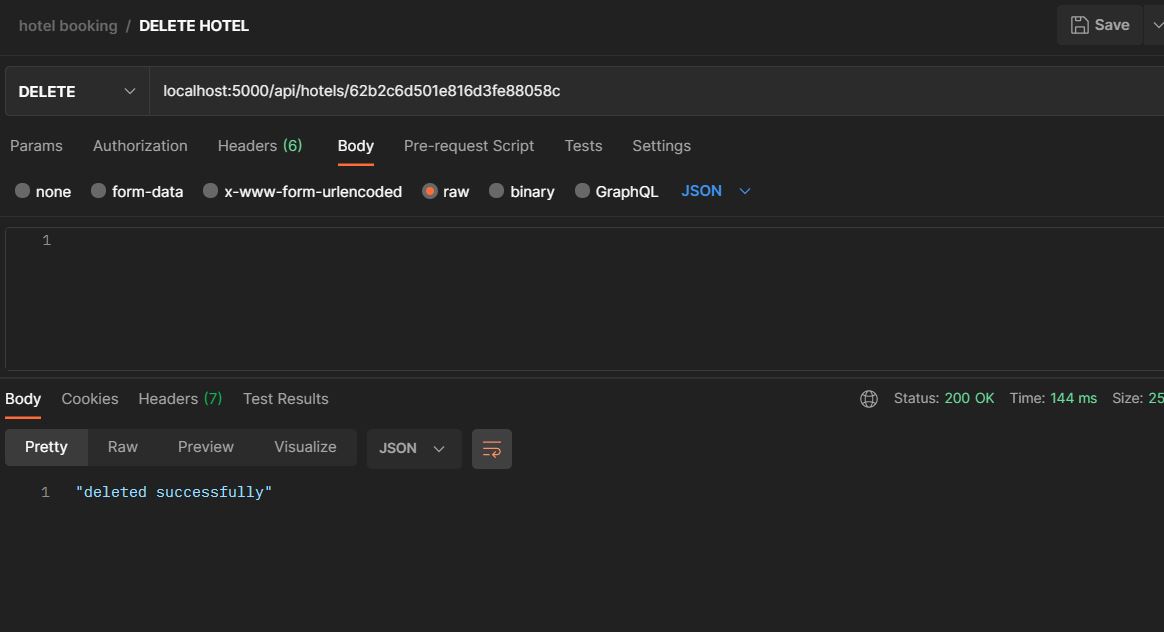
Updating through postman



Delete in mongodb



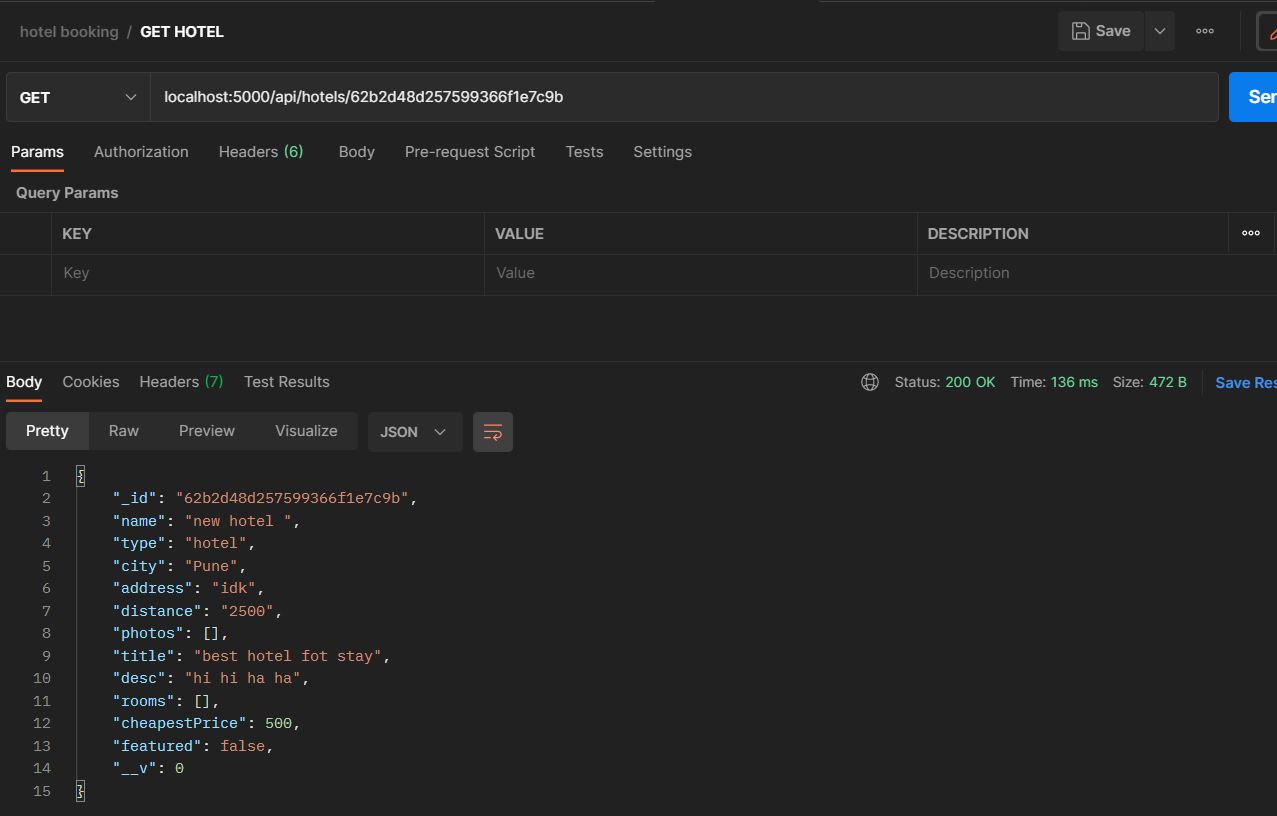
Deletion data through postman



GET or reading data in mongodb



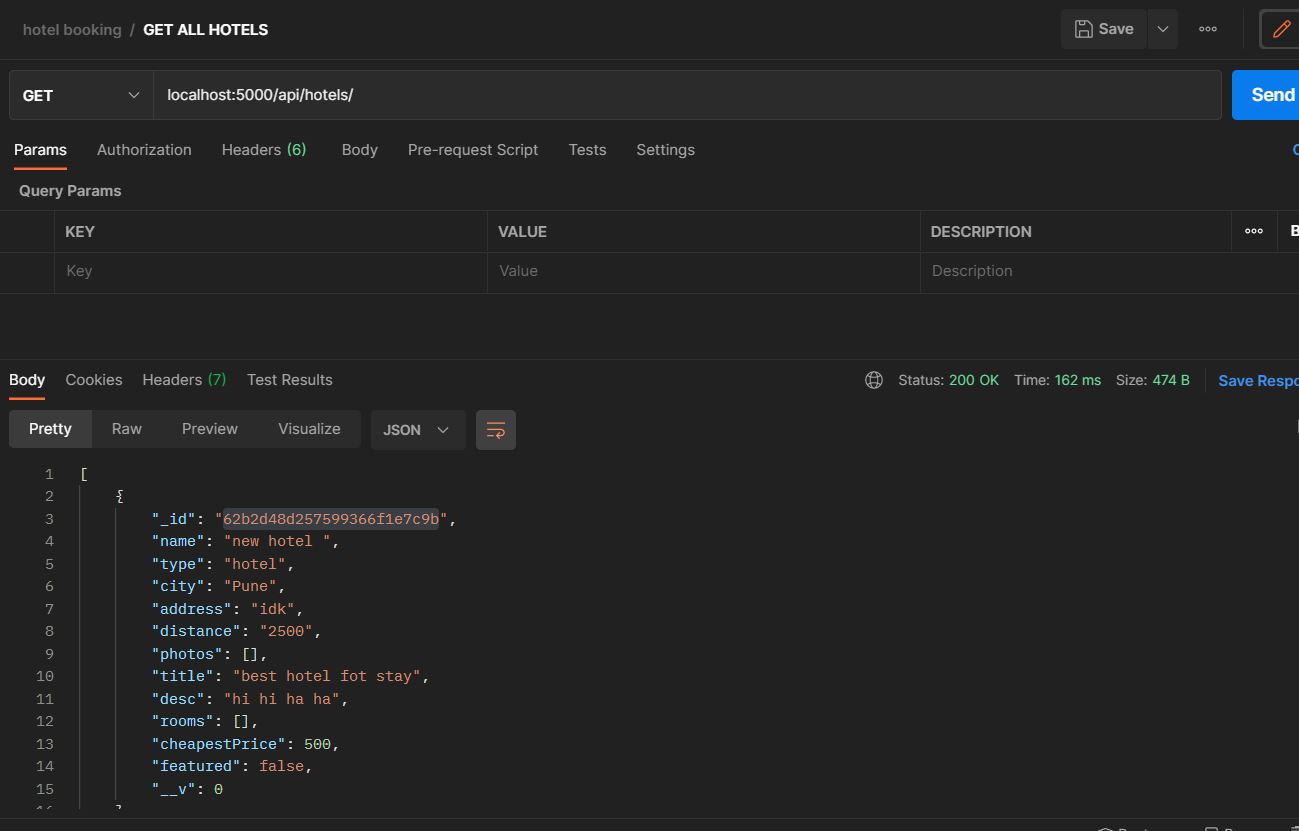
Reading data through postman



GET ALL or reading all data in mongo db:



GET ALL using postman



**Password Encoding:**

Install bcryptjs: yarn add bcryptjs or npm i bcryptjs

Import: import bcrypt from “bcryptjs”

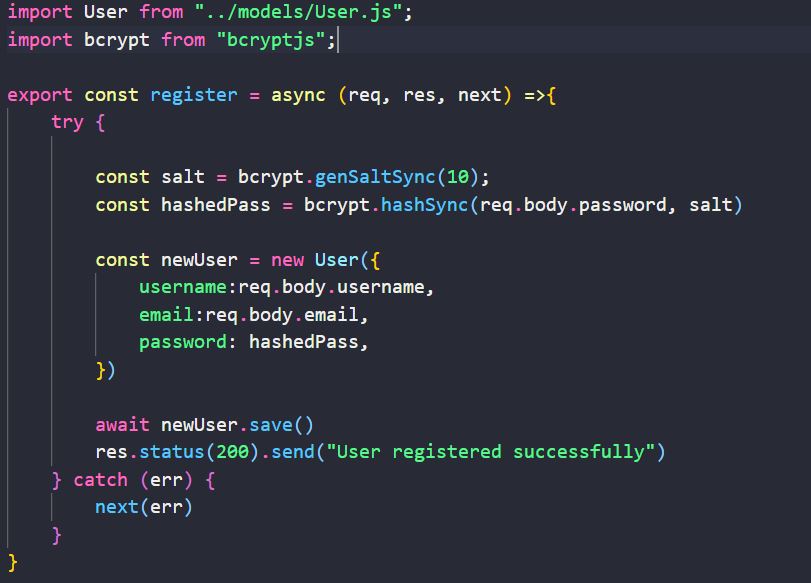
Use:

const salt = bcrypt.genSaltSync(10);

cons hashedPass = bcypt.hashSync(req.body.password, salt);

now **req.body.password** has ecrypted and become **hasedPass**

**Registration with hashed password:**



Check hashed password is same as user requested :



req.body.password ----- > password requested by user for sign-in

user.password --------- > password stored in db

**JWT with cookie:**

Install jwt – **yarn add jsonwebtoken**

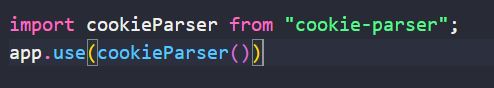
Create jwt token: 

Jwt.sign({data}, secrete\_key)

Use cookie:

Install cookie – **yarn add cookie-parser**

* Import cookie in main file(index.js) and use as middleware



* Send cookie with response



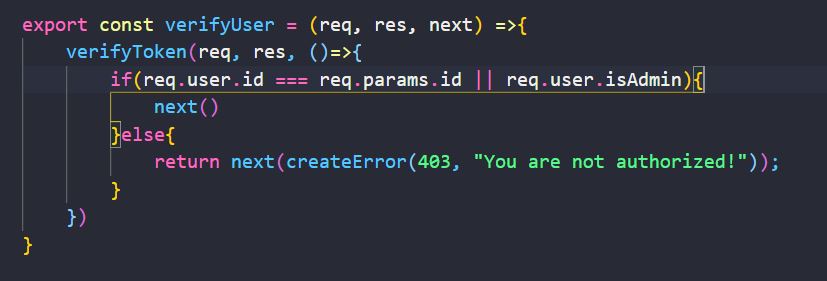
**Login with hashed password:**

****

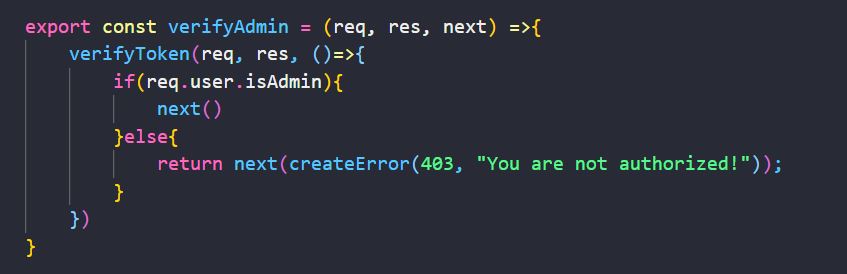
**Verify token :**

****

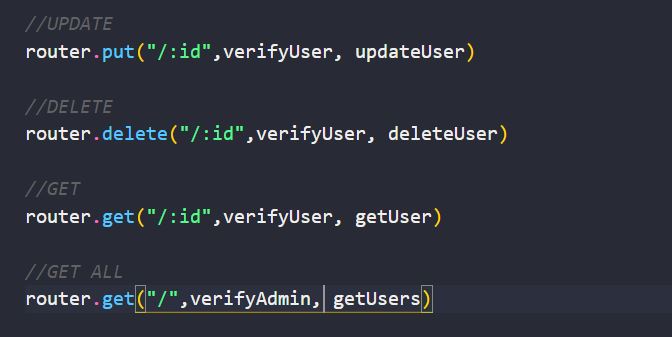
**Verify user to delete any account (they can delete only their account):**

****

**Verify admin (admin can delete any account) :**

****

**Use these verifications as middleware as below:**

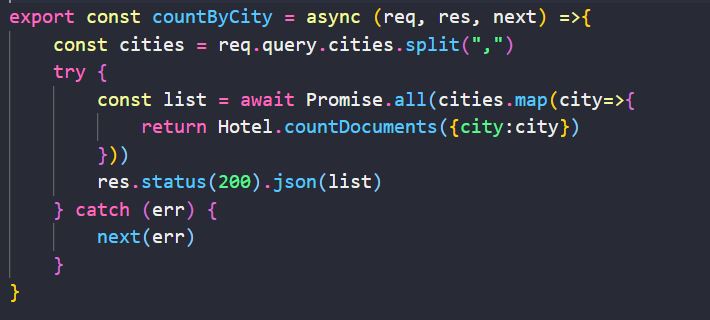
****

**Deal with query:**

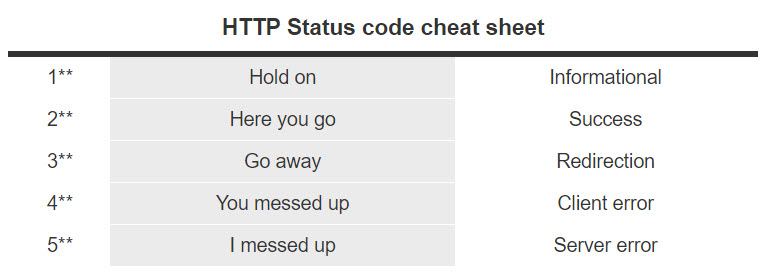
Query starts with question-mark “**?**”, then **key1=value1,value2&key2=val1,val2**

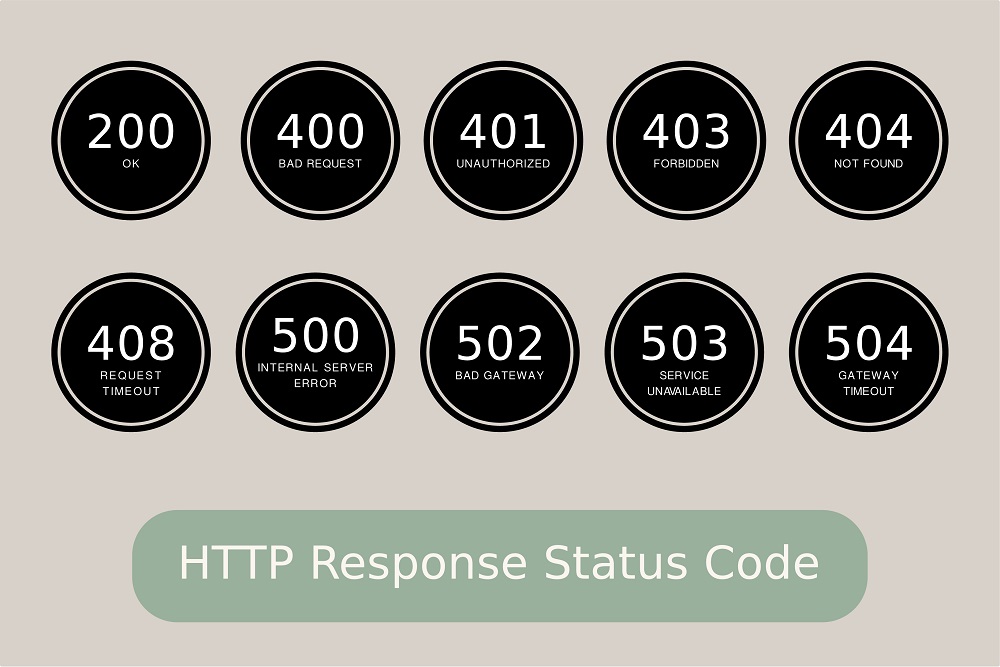
****

Accessing query:

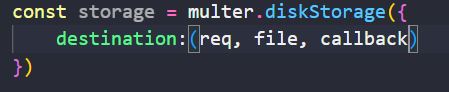


**HTTP Status Code –**





Multer





Important:

* findOne
* findByIdAndDelete vs findByIdAndRemove
* req.params vs req.query
* send vs json
* put vs fetch
* schema vs model

**MongoDB Important –**

* findOne({username: req.body.username})
* findById(req.body.id)
* findByIdAndUpdate(req.params.id, {$set: req.body}, {new: true})
* findByIdAndDelete(req.params.id)
* deleteMany({username: req.params.username})