**Setting up server, setting server, server setup, BE setup,** **back end setup,** **setting BE, setting up BE, setting back end, setting up back end**

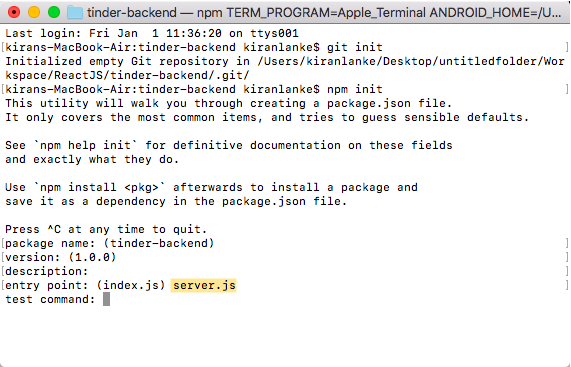
**Step1:**

Create a folder for back end

git init

**Step2:**

npm init



**Step3: Package.json**

{

"name": "tinder-backend",

"version": "1.0.0",

"description": "",

"main": "server.js",

"type": "module",

"scripts": {

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

"start": "node server.js"

},

"author": "",

"license": "ISC"

}

**Step4: Dependencies**

npm i express mongoose

sudo npm i -g nodemon

npm i cors // Will add headers for every request

**Step5: Setup mongoDB**

Login into mongodb.com | sign in

Create new project

**Step6: Starting server**

nodemon server.js

**Step7: Connect our database to MongoDB**

Go to mongodb.com | Database Access | Add New Database User | give user name and password (in my case admin/4oSrmY7SJQCrjzgt) | Add User

Network Access | Add IP Address | Allow Access From Anywhere | Confirm

Clusters | Connect | Connect your application | Copy | go to your code and configure the server by updating the password and dbname

**Step8: dbCards.js**

import mongoose from 'mongoose';

const cardSchema = mongoose.Schema({

name: String,

imgUrl: String

})

export default mongoose.model('cards', cardSchema);

**Step9: server.js**

import express from 'express';

import mongoose from 'mongoose';

import Cors from 'cors';

import Cards from './dbCards.js';

// App Config

const app = express();

const port = process.env.PORT || 8001;

const connection\_url = 'mongodb+srv://admin:4oSrmY7SJQCrjzgt@cluster0.2cdg7.mongodb.net/tinderdb?retryWrites=true&w=majority'

// Middleware

app.use(express.json());

app.use(Cors());

// DB Config (Connecting DB)

mongoose.connect(connection\_url, {

useNewUrlParser: true,

useCreateIndex: true,

useUnifiedTopology: true,

})

// API Endpoints

app.get('/', (req, res) => res.status(200).send("Hello World"));

app.post('/tinder/cards', (req, res) => {

const dbCard = req.body;

Cards.create(dbCard, (err, data) => {

if (err) {

res.status(500).send(err);

} else {

res.status(201).send(data);

}

});

});

app.get('/tinder/cards', (req, res) => {

Cards.find((err, data) => {

if (err) {

res.status(500).send(err);

} else {

res.status(200).send(data);

}

});

});

// Listener

app.listen(port, () => console.log(`Listening on host: ${port}`));

**Step10: Checking sample call in post man**

Get: <http://localhost:8001>

Output: Hello World

**Step11: Post cards**

Post: <http://localhost:8001/tinder/cards>

Body | Raw | JSON

[

{

"name": "Elon Musk",

"imgUrl":

"https://thumbor.forbes.com/thumbor/fit-in/416x416/filters%3Aformat%28jpg%29/https%3A%2F%2Fspecials-images.forbesimg.com%2Fimageserve%2F5f47d4de7637290765bce495%2F0x0.jpg%3Fbackground%3D000000%26cropX1%3D1398%26cropX2%3D3908%26cropY1%3D594%26cropY2%3D3102"

},

{

"name": "Jeff Bezos",

"imgUrl":

"https://thumbor.forbes.com/thumbor/fit-in/416x416/filters%3Aformat%28jpg%29/https%3A%2F%2Fspecials-images.forbesimg.com%2Fimageserve%2F5bb22ae84bbe6f67d2e82e05%2F0x0.jpg%3Fbackground%3D000000%26cropX1%3D560%26cropX2%3D1783%26cropY1%3D231%26cropY2%3D1455"

}

]

**Step12: Checking cards are available or not**

Get: <http://localhost:8001/tinder/cards>

Output:

[

{

"\_id": "5ff58e8b0902dc024bdc7f17",

"name": "Elon Musk",

"imgUrl": "https://thumbor.forbes.com/thumbor/fit-in/416x416/filters%3Aformat%28jpg%29/https%3A%2F%2Fspecials-images.forbesimg.com%2Fimageserve%2F5f47d4de7637290765bce495%2F0x0.jpg%3Fbackground%3D000000%26cropX1%3D1398%26cropX2%3D3908%26cropY1%3D594%26cropY2%3D3102",

"\_\_v": 0

},

{

"\_id": "5ff58e8b0902dc024bdc7f18",

"name": "Jeff Bezos",

"imgUrl": "https://thumbor.forbes.com/thumbor/fit-in/416x416/filters%3Aformat%28jpg%29/https%3A%2F%2Fspecials-images.forbesimg.com%2Fimageserve%2F5bb22ae84bbe6f67d2e82e05%2F0x0.jpg%3Fbackground%3D000000%26cropX1%3D560%26cropX2%3D1783%26cropY1%3D231%26cropY2%3D1455",

"\_\_v": 0

}

]