

# FRONTEND

## Home Screen

KONE Equipment Manager

Search equipment

Equipment Number

363846

SEARCH EQUIPMENT 🔍

Equipment Number:

363846

Address:

Ysitie 20

Contract Start Date:

06.11.2017

Address:

27.03.2023

Status:

RUNNING

List of Equipments

Limit

5

FETCH 🔍

EqNumber	Address	Start Date	End Date	Status
363846	Ysitie 20	06.11.2017	27.03.2023	RUNNING ✓
363845	Kajaaninkatu 20	22.11.2022	27.03.2023	STOPPED !
363844	Snellmaninkatu 25	01.08.2022	28.03.2023	RUNNING ✓
363843	Kerkkolankatu 98	01.08.2022	06.03.2023	RUNNING ✓
363842	Rörgrunsvägen 33	23.08.2022	10.03.2026	RUNNING ✓

ADD NEW EQUIPMENT +

**React MUI(Material UI)** has been used to for each component. **TypeScript** used.

Components inside HomeScreen:

1. AppBar(on Top)
2. The Search Equipment component on the left, where you can search any equipment by it's number.
3. The List view on the right where by default it shows 5 equipments(latest added first). There is option to limit the number of equipments shown in the tabular view.
4. The "Add Equipment" button which opens up the Drawer from right side and allows to add new equipment to DB.

### List Equipments(Task 1)

Equipment details shown in MUI Tabular view. By default shows 5 equipment details(last added on top). There is an option to increase the limit(max 100 right now).

#### List of Equipments

Limit

5

FETCH 🔍

EqNumber	Address	Start Date	End Date	Status
363846	Ysitie 20	06.11.2017	27.03.2023	RUNNING ✓
363845	Kajaaninkatu 20	22.11.2022	27.03.2023	STOPPED !
363844	Snellmaninkatu 25	01.08.2022	28.03.2023	RUNNING ✓
363843	Kerkkolankatu 98	01.08.2022	06.03.2023	RUNNING ✓
363842	Rörgrunsvägen 33	23.08.2022	10.03.2026	RUNNING ✓

## Search Equipment(Task 2)

Takes an equipment number as input, shows the returned details in MUI card.

### Search equipment

Equipment Number

363846

SEARCH EQUIPMENT 🔍

Equipment Number:

363846

Address:

Ysitie 20

Contract Start Date:

06.11.2017

Address:

27.03.2023

Status:

RUNNING

## Add Equipment Drawer(Task 3)

The “Add new Equipment” drawer contains a form created using **Formik**. Basic Form validations on frontend are handled using **Yup**. I did not add complex validations on UI just to keep it simple and quick.

KONE Equipment Manager

Search equipment

Equipment Number

363846

SEARCH EQUIPMENT 🔍

Equipment Number:

363846

Address:

Ysitie 20

Contract Start Date:

06.11.2017

Address:

27.03.2023

Status:

RUNNING

List of Equipments

Limit

5

FETCH 🔍

EqNumber	Address	Start Date	End Date
363846	Ysitie 20	06.11.2017	27.03.2023
363845	Kajaaninkatu 20	22.11.2022	27.03.2023
363844	Snellmaninkatu 25	01.08.2022	28.03.2023
363843	Kerkkolankatu 98	01.08.2022	06.03.2023
363842	Röörgrunsvägen 33	23.08.2022	10.03.2026

Add new Equipment

Equipment Number

0

Address

Contract Start Date

07/01/2022

undefined

Contract End Date

07/01/2022

undefined

RUNNING

SUBMIT

### Add new Equipment

Equipment Number

Address

Contract Start Date

Contract End Date

July 2022

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

MUI Datepickers are used for dates.

### Basic Validations and Error Messages

**Axios interceptors** have been used to intercept request and response and avoid code duplication while sending requests to REST apis. **React toastify** has been used to show any errors that comes from API, no separate handling needs to be done inside any components. This screenshot shows the error on bottom left corner of screen when a Duplicate Equipment number is added.

### KONE Equipment Manager

#### Search equipment

Equipment Number

SEARCH EQUIPMENT

Equipment Number: 363846

Address: Ystie 20

Contract Start Date: 06.11.2017

Address: 27.03.2023

Status: RUNNING

#### List of Equipments

Limit

FETCH

EqNumber	Address	Start Date	End Date
363846	Ystie 20	06.11.2017	27.03.2023
363845	Kajaaninkatu 20	22.11.2022	27.03.2023
363844	Snelmaninkatu 25	01.08.2022	28.03.2023
363843	Kerikkolankatu 98	01.08.2022	06.03.2023
363842	Rörgrunsvägen 33	23.08.2022	10.03.2026

### Add new Equipment

Equipment Number

Address

Contract Start Date

Contract End Date

RUNNING

SUBMIT

Duplicate Equipment Number

## Test Cases

Test cases have been written for some components using RTL and jest.



# BACKEND

**Serverless framework** has been used just to speed up development and deployment. Amplify could also have been used but since I am using MongoDB Atlas as the No-Sql DB, it becomes little tricky to connect to it.

Amplify has good preference for DynamoDB.

With serverless, you just have to write a YML(Infra as a code) and behind the scene it uses CloudFormation to do the work and deploy it on AWS, just make the process easy.

```
! serverless.yml
1  org: pdhruv93
2  app: kone-backend
3  service: kone-backend-mongodb-service-2
4  frameworkVersion: "3"
5
6  provider:
7    name: aws      deploys to aws
8    stage: dev
9    runtime: nodejs16.x
10   httpApi:
11     cors: true    allows cors for frontend
12
13  functions:
14    listEquipments:
15      handler: handlers/listEquipments.listEquipments
16      events:
17        - httpApi:
18            path: /equipment/search
19            method: get
20
21    getEquipment:
22      handler: handlers/getEquipment.getEquipment
23      events:
24        - httpApi:
25            path: /equipment/{equipmentNumber}
26            method: get
27
28    addEquipment:
29      handler: handlers/addEquipment.addEquipment
30      events:
31        - httpApi:
```

Different endpoints and their corresponding Handlers

## Handler Functions

Each handler function has been put in their respective file. TypeScript has not been used. MongoDB connection code has been placed in separate file so that the connection is reused instead of creating new connection for each request. There are 3 handler functions created respective to each task.

## **SOME POINTS THAT I WOULD LIKE TO HIGHLIGHT ABOUT BACKEND**

1. I have not used any middlewares for validations or parsing requests or preparing responses. Most of the work has been done manually just to keep it simple and quick.
2. I had added IAM auth to secure the endpoints(it's quite simple with YML file). It was working nicely with Postman but somehow it was not working on the React frontend because there is a tricky way to send the credentials. I didn't want to consume too much time, so removed it.

EXPLORER

OPEN EDITORS

- JS listEquipments.js handlers
- KONE-BACKEND-...
  - .serverless
  - handlers
    - JS addEquipment.js
    - JS getEquipment.js
    - JS listEquipments.js
  - node\_modules
  - schema
  - .env
  - .gitignore
  - JS mongo-client.js
  - package-lock.json
  - package.json
  - serverless.yml

```
1 const dbConnector = require("../mongo-client");
2
3 module.exports.listEquipments = async (event) => {
4   const db = await dbConnector.getDBInstance();
5   const limit = event.queryStringParameters?.limit || 5;
6
7   if (isNaN(limit) || Number(limit) < 1) {
8     return {
9       statusCode: 400,
10      headers: {
11        "Access-Control-Allow-Origin": "*",
12        "Access-Control-Allow-Credentials": true,
13      },
14      body: JSON.stringify({
15        message: "Invalid Limit",
16      }),
17    };
18  }
19
20  const equipments = await db
21    .collection("equipments")
22    .find()
23    .sort({ equipmentNumber: -1 })
24    .limit(Number(limit))
25    .toArray();
```

## Serverless Dashboard



## AWS Dashboard

Services Search for services, features, blogs, docs, and more [Alt+S]

N. Virginia dhruv.pandey

### AWS Lambda

Dashboard Applications **Functions**

Additional resources: Code signing configurations, Layers, Replicas

Related AWS resources: Step Functions state machines

Lambda > Functions

Functions (3) Last fetched now Actions Create function

Filter by tags and attributes or search by keyword

	Function name	Description	Package type	Runtime	Last modified
<input type="checkbox"/>	kone-backend-mongodb-service-2-dev-getEquipment	-	Zip	Node.js 16.x	10 hours ago
<input type="checkbox"/>	kone-backend-mongodb-service-2-dev-listEquipments	-	Zip	Node.js 16.x	10 hours ago
<input type="checkbox"/>	kone-backend-mongodb-service-2-dev-addEquipment	-	Zip	Node.js 16.x	10 hours ago

## Mongo DB Atlas(Cloud) Dashboard

The screenshot displays the MongoDB Atlas web interface. On the left, a sidebar contains navigation links: Deployment, Database, Data Lake, Data Services, Triggers, Data API, Data Federation, Security, and Advanced. The 'Database' section is active, showing the 'equipments' collection. The main panel displays the 'equipments.equipments' collection with a filter bar and a query results section showing 10 documents. The documents are JSON objects containing fields like \_id, equipmentNumber, address, contractEndDate, contractStartDate, and status.