**IOT Assignment code implementation document**

**PURPOSE OF THIS DOCUMENT:**

Main motive of this document is to provide in-sights on how the design of code is made, main components involved in structuring the project, functionality of the code, deployment in IBM cloud infrastructure etc.

**Software development tools/technology stack used:**1. Spring boot(Java based frame-work to develop micro-services)

2. Maven(For maintaining various dependencies and build purposes)

3. HTML(Used to provide UI for easy usage of the project, internally made ajax calls to retrieve json reponse from backend apis and display them to the user in tabular format also to receive equipment details as input and store them to database)

4.IBM Cloudant(IBM Cloud service provides a fully managed, distributed JSON document database. Instantly deploy an instance, create databases and independently scale thoughput capacity and data storage to meet user requirements. Used Java libraries to contact IBM cloudant service in IBM cloud to store and fetch data related to project)

5.Postman(During local testing of API’s,it is used)

🡪Followed code libraries of java 1.8 version in code development.

**Internal Components used and their description:**

🡪As per the standards, the packages are named after the company name ‘Kone’.

🡪Used MVC pattern for easy maintainance of code.

🡪For three use cases, created three rest api’s and have implemented corresponding business logic in corresponding service methods.

**User Interface:**🡪Provided a view controller in the code to route the initial request to index.html file.

🡪Ajax calls are designed to call rest api’s to retrieve json data and display to the user in two of the use cases which are to fetch details of particular equipment and to fetch details regarding ‘n’ equipments.

🡪UI also has the capability to take different attributes specified in the requirement document as input and store the same in IBM cloudant database.

🡪If data provided by user already exist then UI will throw a message that the data provided already exist in the database

🡪User can retrieve the details from database and will be displayed the same in a tabular format.

**Internal code details along with a description of their implementation:**

**“Add” rest method implementation:**🡪To implement use-case for storing the data related to equipment in IBM cloudant service.

🡪Internally makes a service call to method “addDetails” in which connection is made to establish between IBM cloudant service and data will be sent to store.

🡪This service also makes use of internal logic implemented to verify whether the data that the user submitted is already present in IBM cloudant service or not.

**🡪“getSpecificEquipmentDetails” rest method implementation:**

🡪To implement one of the use case to fetch equipment details specific to a equipment number.

🡪Internally calls a service method named “getDetails” to fetch the details of equipment specific to a particular equipment number.

🡪Internally the service establishes a communication with IBM cloudant service and retrieves the data required.

**“getEquipmentRecords” rest method implementation:**🡪To implement one of the use case to fetch records of ‘n’ number at once.

🡪Internally this rest call makes a call to a service named “getEquipmentrecordDetails”.

🡪Internally this service establishes a connection with IBM cloudant service and fetches data related to ‘n’ records to the user.

**Flow Diagram:**

RestController Service Interface

API (Post request)to add equipment details(Throw an error in case of duplicate equipment details addition)

API(Get request) to get equipment details with respect to specific equipment number.

API(Get request) to fetch ‘n’ number of equipment details as requested by user.

Service Interface

Receives the request from controller to add equipment details. Will contact IBM cloudant service and add corresponding details.

Receives the request from controller to fetch the details of specific equipment and will retrieve the same by querying IBM cloudant service.

Receives the request to fetch ‘n’ number of equipment details, this service will query the IBM cloudant service and responds with required data.

User Interface(Can be called by view controller)

IBM Cloudant service database

**Deployments:**

Used IBM cloud for deployment. Have linked IBM cloudant service to the spring boot service deployed. Application in cloud environment is deployed as a docker image and intended to run as a container.

With Kubernetes which is a container orchestration service , the deployed containers can be scaled automatically depending upon the number of requests the application receives.

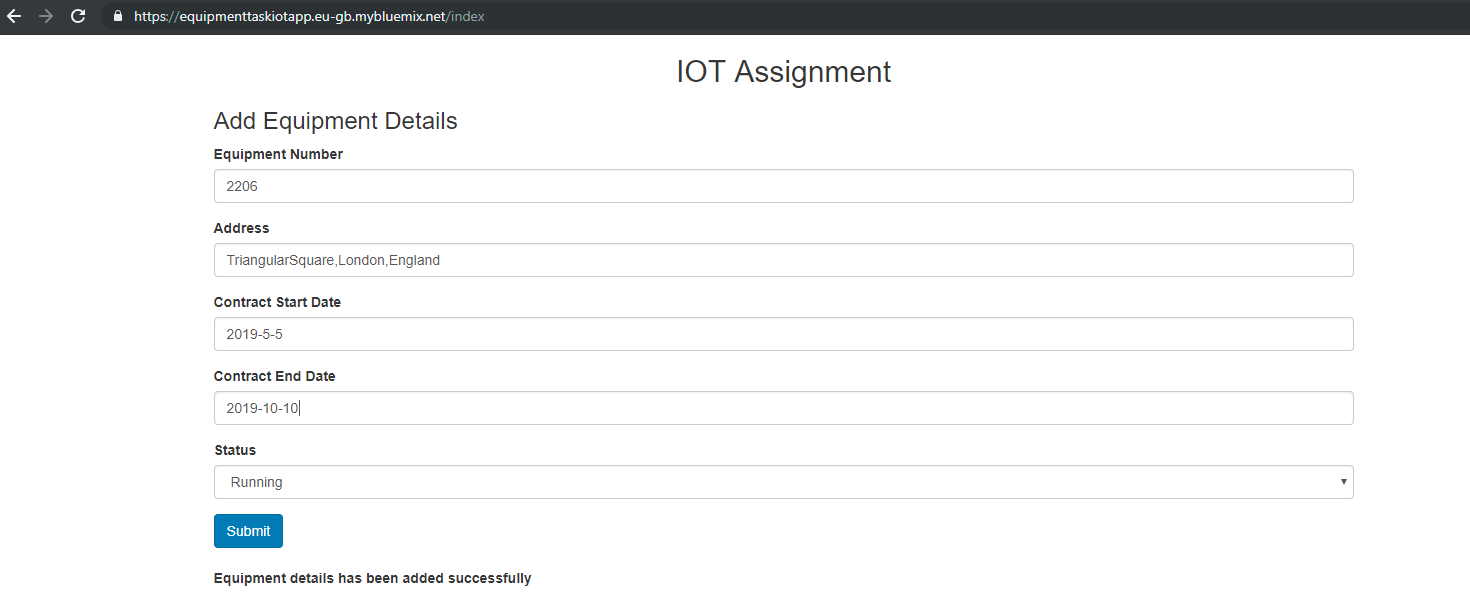
**Functionality Explain with Screen-shots:**

**URL to hit:**

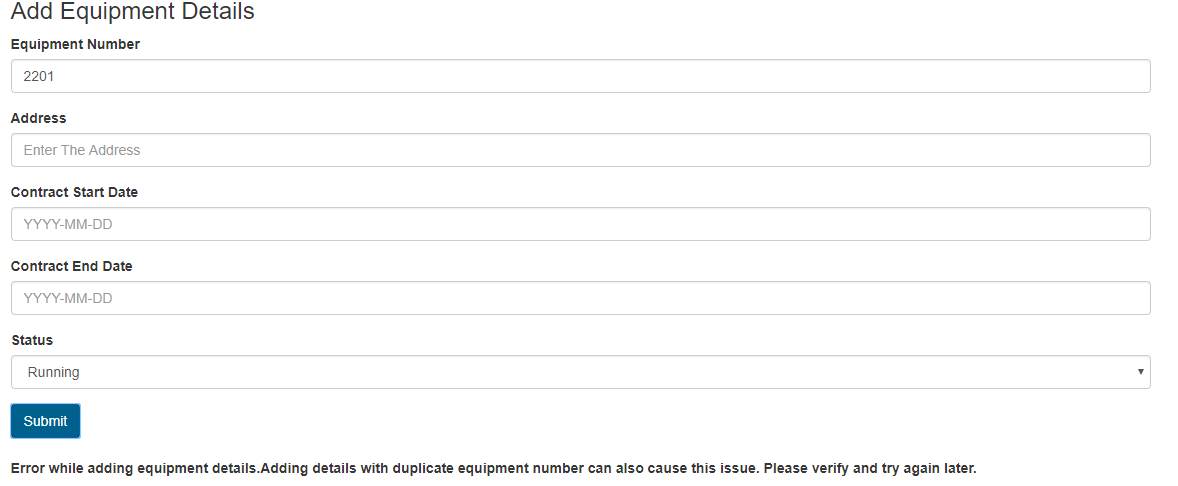
[**https://equipmenttaskiotapp.eu-gb.mybluemix.net/index**](https://equipmenttaskiotapp.eu-gb.mybluemix.net/index)

When we hit the url mentioned above,we get a UI screen with all the required attributes listed to be filled with data

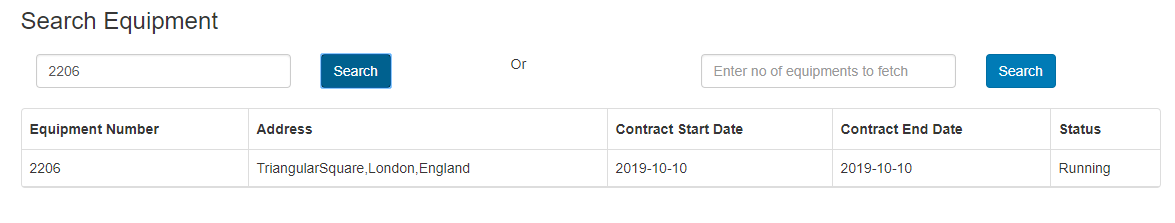
**Functionality to store the information:**



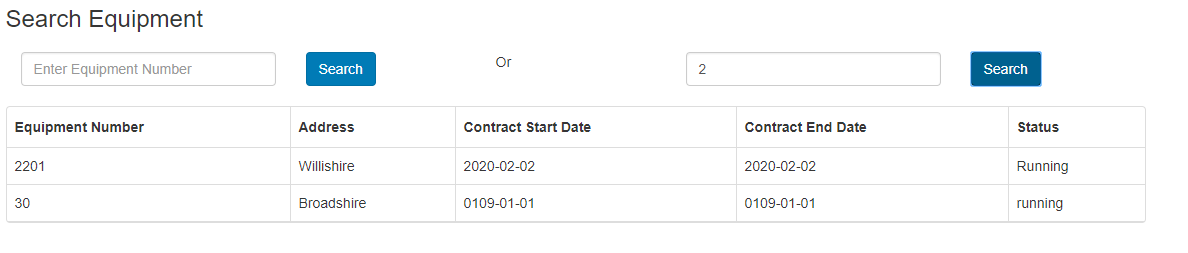
When we click on ‘submit’ then all the details will be sent to IBM cloudant service and stored.



When details that are already existing in IBM cloudant service database then error message will be invoked when trying to store the data as shown in the screen shot above.

**Functionality to get information requested of a particular service:**

When we enter a equipment number which is stored in the IBM cloudant database then it will fetch the information regarding it as shown in screen shot above in tabular format.

**Functionality to get information requested for ‘n’ number of equipments:**

When we enter the number of equipments to fetch in the field mentioned then UI will fetch information regarding and will display in the tabular format as shown above.

**Internal API Details:**

**API Details to post equipment details:**

**POST Request:**

<https://equipmenttaskiotapp.eu-gb.mybluemix.net/equipmentService/equipment>

RequestBody:  
{

"equipmentNumber":"1301",

"address":"Willishire",

"contractStartDate":"189",

"contractEndDate":"107",

"status":"running"

}

**API Details to get information of a particular equipment:**

**Get Request:**

[https://equipmenttaskiotapp.eu-gb.mybluemix.net/equipmentService/equipment/{EquipmentNumber}](https://equipmenttaskiotapp.eu-gb.mybluemix.net/equipmentService/equipment/%7bEquipmentNumber%7d)

**API Details to get information about ‘n’ number of equipments:**

**Get Request:**

[https://equipmenttaskiotapp.eu-gb.mybluemix.net/equipmentService/equipment/search/{limit}](https://equipmenttaskiotapp.eu-gb.mybluemix.net/equipmentService/equipment/search/%7blimit%7d)

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF THE DOCUMENT\*\*\*\*\*\*\*\*\*\*\*\*\***