**SUGARBOX-User API**

**Design Document &**

**API REST Specification**

**Jun 9th 2021**

**A solution (REST API) used by Admin and Users.**

**1. Technology Used:**

* NodeJS, IDE- VSCode
* MySQL

**2. Project Setup:**

GitRepo:  **[https://github.com/KishoreM12/sugarbox.git](https://github.com/kishore12rao/vendormachine) or SugarBoxTask.zip**

* Database: **vending\_machine**

1. Create database name with “vending\_machine”, Load the backup script using Dumpfile: vending\_machine14.sql (also present in project root folder)
2. Users table are preloaded with 2 users admin and service1 wrt roles

Command to load dump from file:

[mysql dir]/bin/mysql -u username –p password vending\_machine < vending\_machine.sql

* Spring-boot: **JioVendingMachine**

Import Project as Maven to IDE and run it as **Java Application** which runs on embeded runs on embeded tomcat container.

or

Run in cmd/terminal :

1. mvn clean && mvn package
2. java –jar /target/JioVendingMachine.jar

\*\*Note: Update Mysql Credentials, Change Tomcat Server port in JioVendingMachine/src/main/resources/application.properties

\*\*Please update your’s system path of application.properties in DBConnection.java

**3. Requirement :**

* **REST API Services**
* Vending machine service person goes to vending machine, collects the money and refills the items in the vending machine. This data needs to be uploaded in vending machine company's data base (items refilled, money collected).
* Admin and Service Person should have list of API to access Vending Machines & Inventory

**4. MySQL Database & Tuples:**

1. **users** (username,password,name,role)

– has two preloaded users **admin** and **service1.**

1. **vending\_machine** (name,description,itemcount,itemrefilled,moneycollected)
   1. **Admin:** Add,Update properties, Delete, Fetch VM’s, Fetch Report.
   2. **ServicePerson:** Update ItemsRefilled, MoneyCollected into VM’s.
2. **inventory** (products)
   1. **Admin:** Add new Products to Inventory
   2. **ServicePerson:** Update Inventory Products Data

**5. NonFunctional Usage:**

1. Git has been used.
2. Maven Project
3. Logging (Using Logbook) - /jiovmlogs/ap.log
4. Unit-Test Cases
5. API Call Data Reports: Not Created API but used /metrics, /health

**6. Postman API Documentation**:

**[https://documenter.getpostman.com/view/3346130/SWLiamS1?version=latest#49e94602-ec13-351a-a8c8-150bbee306b8](https://documenter.getpostman.com/view/3346130/SWLiamS1?version=latest" \l "49e94602-ec13-351a-a8c8-150bbee306b8)**

**7. API Specification:**

APIs are authenticated thorugh Basic Authorization

**Admin Credentials**

* (role-admin) -> username: **admin** password: **admin**

**Supervisor Credentials**

* (role-service)-> username: **service1**, password: **service1**

1. **VM-Admin** :
2. **POST :** ADD INVENTORY

**URL**

<http://localhost:8080/inventory?noOfProd=100>

**Headers**

Content-Type application/json

Authorization Basic YjiodWjiIADS=

**Params**

noOfProd 100

1. **POST :**ADD VENDING MACHINE

**URL**

[http://localhost:8080/vendingmachine](http://localhost:8080/inventory?noOfProd=100)

**Headers**

Content-Type application/json

Authorization Basic YjiodWjiIADS=

**Body**

{

"name": "VMware 5",

"description": "5 vending Machine",

"itemCount":5

}

1. **PUT :**UPDATE VENDING MACHINE PROPS

**URL**

[http://localhost:8080/vendingmachine/4](http://localhost:8080/vendingmachine/4e)

**Headers**

Content-Type application/json

Authorization Basic YjiodWjiIADS=

**Body**

{

"name": "VMware 5",

"description": "5 vending Machine"

}

1. **DELETE :** DELETE VENDING MACHINE

**URL**

<http://localhost:8080/vendingmachine/2>

**Headers**

Content-Type application/json

Authorization Basic YjiodWjiIADS=

1. **GET :** FETCH ALL VENDING MACHINE

**URL**

<http://localhost:8080/vendingmachine>

**Headers**

Authorization Basic YjiodWjiIADS=

1. **GET :** GET VENDING MACHINE BY ID

**URL**

<http://localhost:8080/vendingmachine/3/>

**Headers**

Authorization Basic YjiodWjiIADS=

1. **GET :** FETCH VENDIND MACHINE REPORT

**URL**

<http://localhost:8080/vendingmachine/report/>

**Headers**

Authorization Basic YjiodWjiIADS=

1. **Inventory**-**Service Person**
2. **PATCH:**REFILL VENDING MACHINE

**URL**

[http://localhost:8080/ vendingmachine/4](http://localhost:8080/%20vendingmachine/4)

**Headers**

Content-Type application/json

Authorization Basic YjiodWjiIADS=

**Body**

{

"itemRefilled": 100,

"moneyCollected": 5000

}

1. **PUT :**UPDATE INVENTORY PRODUCTS LEFT

**URL**

<http://localhost:8080/inventory?noOfProd=10>

**Headers**

Content-Type application/json

Authorization Basic YjiodWjiIADS=

**Params**

noOfProd 10

1. **API Metrics**
2. **GET :** METRICS

**URL**

<http://localhost:8080/metrics>

**Headers**

Authorization Basic YjiodWjiIADS=

1. **GET :** HEALTH

**URL**

<http://localhost:8080/health>

**Headers**

Authorization Basic YjiodWjiIADS=

1. **GET :** LOGPATH

**URL**

<http://localhost:8080/logpath>

**Headers**

Authorization Basic YjiodWjiIADS=