**FINANCE TRUST BANK INTERVIEW**

**RECIPE APP DEPLOYMENT GUIDE**

**BY**

**HEDWIG ORIEBA**

*(*[*oriehedwig@gmail.com*](mailto:oriehedwig@gmail.com)*,* [*oriebahedwig@hotmail.com*](mailto:oriebahedwig@hotmail.com)*)*

*mob: 0778439060/0783900914*

**JUNE 2024**

Contents

[**Overview** 3](#_Toc170481698)

[**High Level Process Flow** 3](#_Toc170481699)

[**Process Flow Description** 3](#_Toc170481700)

[**Setup Components** 4](#_Toc170481701)

[**Setup Component Specifications** 4](#_Toc170481702)

[Windows Client Machine 4](#_Toc170481703)

[Hardware: 4](#_Toc170481704)

[Software: 4](#_Toc170481705)

[**DEPLOYMENT WORK BREAK DOWN STRUCTURE (WBS)** 5](#_Toc170481706)

[Nodejs & Docker 5](#_Toc170481707)

[SERVICE CONSUMPTION 6](#_Toc170481708)

[Invoke the unique recipe names endpoint in your browser as shown below: 6](#_Toc170481709)

[END 7](#_Toc170481710)

# **Overview**

This document provides the technical synopsis of the deployment steps and activities needed to successfully deploy the Recipe Statistics tracking API service. Once successfully deployed, the service shall expose the following set of functionalities:

1. Ability to know the number of unique recipes in each Json dataset.
2. Ability to know how many times each recipe occurs.
3. Ability to know the highest most ordered recipe(s).
4. Ability to know recipes that contain a certain string of characters.

# **High Level Process Flow**

The API service is implemented to support a typical client server architecture. The service runs as a backend Nodejs process that receives request through a logical connection channel (port) called 3044 which grands client’s access to the service resources or endpoints. A typical client can be a web browser or mobile application.

The invocation of the service is by using port forwarding on the docker app container. Specifically, the below command starts the application in a docker environment.

***docker run -p 4098:3044 orieba\_hedwig\_ftbsln***

# **Process Flow Description**

A client agent/application (google chrome / an ASP. Net app.) makes an HTTP GET Request to the service base controller interface called index.js.

1. The application controller then routes the request to a respective handler logic in alignment to the business delegate design pattern.
2. The route handler actions on the request immediately makes a call to the data persistence layer and returns a transfer object known as a response which is returned to the client.
3. The client then either consumes the result set or logs it as per the process front end logic.

# **Setup Components**

1. Windows Client machine with the following:
2. Docker Desktop installed.
3. Nodejs v20.9.0 installed.
4. Virtualization Enabled.
5. Windows Subsystem for Linux enabled.

# **Setup Component Specifications**

## Windows Client Machine

### Hardware:

Windows 11 Pro PC, 8GB RAM, 500 GB Hard Disk space with working network adapter for Internet Access.

### Software:

1. Nodejs.
2. Docker Desktop

# **DEPLOYMENT WORK BREAK DOWN STRUCTURE (WBS)**

## Nodejs & Docker

1. Download and install Nodejs for windows from <https://nodejs.org/en/download>.
2. Verify successful node installation executing the following command at the CLI.



**Input:** node -v

**Output:** v20.9.0

1. Download Docker Desktop for windows from

https://docs.docker.com/desktop/install/windows-install/

1. Perform a pull using the below public URL to gain access to the app docker image.

***docker pull orieba/ftb-app:latest***

1. Run the image in your local docker container environment created by the Docker desktop software: This will start the service. Use the command.

***docker run -p 5555:3044 orieba\_hedwig\_ftbsln***

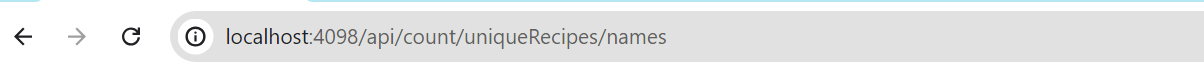
*In this scenario, I am port forwarding from the local port 5555 to the app port 3044. Any port other than 5555 can do.*

*Output:*

**

## SERVICE CONSUMPTION

### Invoke the unique recipe names endpoint in your browser as shown below:



**Response:**

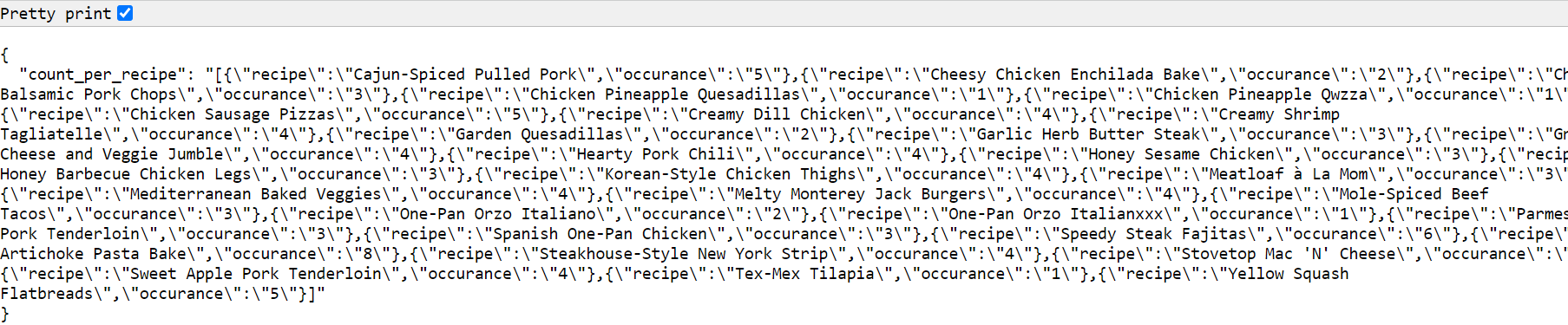
A white background with a black border

Description automatically generated with medium confidence

**Invoke the occurrences endpoint in your browser as shown below.**



**Response:**

****

**Invoke the highest ordered recipe endpoint in your browser as shown below.**



Response:

A white rectangular object with a black border

Description automatically generated

**Invoke the recipe name matching endpoint in your browser as shown below.**



Response:

A screenshot of a computer

Description automatically generated

### END