Project Support Documentation

Technology stack:

1. **Java 8**
2. **Micro-service architecture:** A method of developing the application as a suite of independently deployable, small, modular services in which each service runs a unique process and communicates through a well-defined, lightweight mechanism to serve a business goal.
3. **Spring boot 2:** An opinionated view of building production-ready Spring application.

Spring boot provides these starters:

- **Spring-boot-starter-web:** Starter for building the web application using Spring MVC. Uses Tomcat as the default embedded container
- **Spring-boot-starter-actuator:** Starter for using Spring Boot's Actuator which provides features to help monitor and manage the application.
- **Spring-boot-starter-security:** Starter for using Spring Security.
- **Spring-boot-starter-test:** Starter for testing the application with libraries including JUnit and Mockito.

4. **Apache HttpCore**: Set of low level HTTP transport components.
5. **Junit**: Testing framework for Java.
6. **Mockito**: Mocking framework for unit tests in Java.
7. **Maven**: Project management tool.
8. **Docker**: Micro-service container platform.
9. **Sonar**: Static code analysis tool.
10. **Swagger**: Automated JSON API documentation for API's.
11. **Angular**

Tools:

1. **Apache Maven version 3.5.4**
2. **Docker version 18.06.1-ce**
3. **SonarQube version 7.3**
4. **SonarQube Scanner version 3.2.3**
5. **NodeJs version 10.11.0**
6. **NPM version 6.4.1**
7. **Angular CLI**

Back-end “fizz-buzz”:-

Build and Run:

- **Clean project**

mvn clean

- **Run project with dev profile**

mvn install spring-boot:run

Note:

- Run using Tomcat server on port: 8585
- Dev is the default profile

- **Check actuator services**

http://localhost:8585/actuator

Notes:

- User name = admin & password = admin, located in application.yml file.
- All actuator endpoints are exposed by overriding the default config in application.properties file.

- **Run project with docker profile**

mvn install spring-boot:run -Dspring.profiles.active=docker

Note:

- Run using Tomcat server on port: 8585

- **Create docker image using docker profile**

mvn package -Dspring.profiles.active=docker docker:build

- **Run docker image container**

docker run -d -e "spring_profiles_active=docker" -p 8585:8585 --name fizz-buzz --hostname fizz-buzz fizz-buzz

Note:

- Run using Tomcat server on port: 8585

- **Check unit test report**

{project.directory}/target/surefire-reports/TEST-com.hanfy.fizzbuzz.FizzbuzzServiceImplTest. xml

Note:

- Can be checked from Sonar.

- **Build Sonar reports**

mvn sonar:sonar

- **Check Sonar reports**

http://localhost:9000

Note:

- http://localhost:9000 is the default value for sonar server.

Swagger Rest API documentation:

- **Response returns in JSON format.**
- **Check Swagger API documentation**

http://localhost:8585/swagger-ui.html

Front-end “fizz-buzz-ui”:-

Run:

- **Run project**

Ng serve