Logo

Description automatically generated with low confidence

Sprint #0

Kieran Massey A00144505 | MSc in Cloud Native Computing | 22/02/2022

Contents

[**Introduction** 3](#_Toc96381775)

[**User Stories** 5](#_Toc96381776)

[**Demo of one Microservice (CRUD)** 6](#_Toc96381789)

[**Code Annex** 12](#_Toc96381790)

# 

# **Introduction**

My application will be developed based on video games. Service A will be known as **Games A** which will include details such as:

* Game ID
* Game Name
* Console
* Age Rating
* Price
* Genre
* Child ID (Game\_Detail\_ID) relating to another table

While Service B will be known as **Application B** which will mainly have entertainment devices related information such as:

* Device it can be played on
* Any external devices that can be added to your gaming experience
* Can it be played on an AR or VI system, i.e. Oculus or Playstation VR headset
* Screen resolution – 4K, 8K, 1080P, 1080I, 720P, etc
* Relating ID to Games A.

Both service files will have CRUD implemented functionality. I will use the H2DB for the underlying database for this sprint and have 2 tables for both game files for storage, then in Sprint 1 I will use MySQL to implement my DB for storage.

For Sprint 0, Postman will also be used as the client application to mainly expose endpoints using REST API. A GitHub repository will be created to backup any files and have a repo home for all files regardless.

Diagram

Description automatically generated

*Figure 1 – Application Diagram for Sprint 0 & 1.*

For Sprint 1, I will incorporate the MySQL database instead of the use of the H2DB in the browser. I will also add in multiple new applications to aid my Microservices pipeline project. Of course, also further developing my **Applications B** Microservice and **Games A** Microservice in multiple ways.

Applications I intend on adding for extra functionality are:

* Jenkins
* Swagger
* SonarQube
* Spring Cloud
* API gateways
* Logins & Validation functionality
* Spring Eureka discovery service
* Sleuth, Zipkin.

# **User Stories**

## As a user I want to Start the Application ✔️

## As a user I want to Create a Github repository ✔️

## Code Games A service files to completion ✔️

## Setup 2 Database tables for Game A & Application B ✔️

## Run Games A within Postman Application for REST API acknowledgement ✔️

## Code Application B service files to completion Sprint 1

## Run Application B within Postman Application for REST API acknowledgement Sprint 1

## Ensure each microservice instance registers with a Spring discovery service. Sprint 1

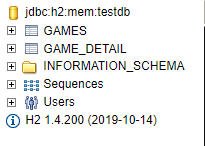
## Ensure the entire Application uses Authentication of some sort. Sprint 1

## REST calls to be traced with Sleuth & Zipkin. Sprint 1

## REST Endpoints to have data validation API. Sprint 1

## Ensure Cloud Native behaviour is included on some level. Sprint 1

# **Demo of one Microservice (CRUD)**



Table

Description automatically generated

Table

Description automatically generated

*Figure 2,3,4 – H2 Database information, tables for Sprint 0.*

Text

Description automatically generated

*Figure 5 – Postman GET call showcasing all stored DB content.*

Graphical user interface, text, application, email

Description automatically generated

*Figure 6 – Postman POST API Call into table.*

Table

Description automatically generated

*Figure 7 – Database after POST was complete as proof.*

Graphical user interface, text, application, email

Description automatically generated

*Figure 8 – Postman GET API Call after POST was complete.*

Graphical user interface, text, application, email

Description automatically generated

*Figure 9 – Attempting a Postman GET API Call into table when ID does not exist.*

Table

Description automatically generated

*Figure 10 – Showing DB before a DELETE action is complete.*

Graphical user interface, text, application, email

Description automatically generated

*Figure 11 – Postman DELETE API Call.*

A screenshot of a computer

Description automatically generated with medium confidence

*Figure 12 – Table after DELETE was complete as proof.*

Graphical user interface, text, application, email

Description automatically generated

*Figure 13 – Postman PUT API Call Complete on ID 8.*

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

*Figure 14,15 – GitHub repository of code backed up.*

# **Code Annex**

My Maven project & code looks like the following:

Graphical user interface

Description automatically generated with low confidence

**GamesSpringApp.java**

Graphical user interface, text, application

Description automatically generated

**JpaGameController.java**

Text

Description automatically generated with low confidence

**GamesRepository.java**

Graphical user interface, text, application

Description automatically generated

**GameDetail.java**

Graphical user interface, text, application

Description automatically generated

**Games.java**

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

**Pom.xml**

Graphical user interface, text, application

Description automatically generated

**Data.sql**

A screenshot of a computer

Description automatically generated with medium confidence

**Application.properties**

Graphical user interface, text, application

Description automatically generated