**Pre-requisites:**

1. JDK 1.8 or older
2. Maven 3.x

**How to Run:**

* The backend apis are developed using Spring boot.
* Build the package using **mvn clean install** command. This will create a JAR file with embedded tomcat server in it.
* Run using mvn spring-boot:run
* Application will be accessible via port 8080 (localhost:8080). Port can be changed if required in application.properties.

## **About the Service**

## The application has 4 entities (Accounts, bookings, guestlogs and records) and each entity has controller,service,service Implementation and Repository classes.

* API mapping is done in the controller and its implementation is done in the service layer.
* For configuring couchbase db,used couchbase config class and mapped entity classes to the buckets in the database.
* Change the connection string(url),username,password and bucket name in application properties to connect to your database.

### **To view Swagger 2 API docs**

* Run the server and browse to <http://localhost:8080/swagger-ui/index.html> or at <http://localhost:8080/v2/api-docs> in Postman

**Queries API**

* Account
  + GET - example.com/accounts/{accountUID}

Get by body uid = phone or email

* + GET - example.com/accounts/{accountUID}/getAccountName

Get account name alone by using accountUId as pathvariable

* + POST - example.com/accounts

Post account by RequestBody

* + GET - example.com/accounts

To find/get all accounts

* + PUT -example.com/accounts/{AccountUId}

Find and update account by passing accountUId as RequestParam save Updated account

* + DELETE - example.com/{AccountUId}

Delete account by passing accountUId as RequestParam

* Record 
  + GET - example.com/record/{record\_Id}
    - Get by query; only used if UJET is configured to re-use existing records.

* + POST - example.com/record
  + GET -example.com/record

find/get all records

* + DELETE -example.com/records/{record\_Id}

Delete record by passing record\_Id as RequestParam

* + PUT or PATCH - example.com/record/{record\_Id}
    - Find and update record by passing record\_Id as RequestParam save updated record

* Guest logs -Done
  + GET - example.com/logs/{logId}
    - Get by query; only used if UJET is configured to re-use existing records.
  + POST - example.com/logs
  + GET -example.com/logs

find/get all guestlogs

* + DELETE -example.com/logs/{logId}

Delete log by passing logId as RequestParam

* + PUT or PATCH - example.com/logs/{logId}
    - Find and update Guestlog by passing logId as RequestParam save updated log
* Bookings
  + GET - example.com/bookings/{bookingId}
    - Get by booking by bookingId
  + GET- example.com/bookings

Get all bookings

* + DELETE -example.com/bookings/{bookingId}

Delete booking by passing bookingId as RequestParam

* + POST -example.com/bookings
  + PUT or PATCH - example.com/bookings/{bookingId}
    - Find and update Guestlog by passing bookingId as RequestParam save updated booking