Clone, build and run instructions:

1. Create project directory and clone the repository from https://github.com/rdelacruz1/coupon-service-api.git.
2. In order to build the project, run **./gradlew build** command from the project directory on the mac terminal. This will create a .jar file at /build/libs.
3. In order to run the API, change directory to /build/libs and run the .jar file using **java –jar jarfilename.jar** command**.**
4. The API will automatically execute a data.sql file at startup that will initialize the database.

Using the API:

There are 6 main endpoints available for use. They are:

1. addBusiness()
2. issueCoupon()
3. findCouponsByCity()
4. findCouponsByRegion()
5. findCouponsByState()
6. findCouponsByCountry()

Using addBusiness endpoint:

Endpoint path: demo/addBusiness.

Parameter: String businessName.

Request example: http://localhost:8080/demo/addBusiness?businessName=example

Using issueCoupon endpoint:

Endpoint path: demo/issueCoupon.

Parameter: Long businessId, String description, Long cityId

Request example from browser:
http://localhost:8080/demo/issueCoupon?businessId=3&description=uber%20coupon&cityId=3

When issuing coupons by city, refer to the following table for cityId's:

cityId	cityName
1	San Jose
2	Oakland
3	Cupertino
4	San Francisco
5	Milipitas

Using findCouponsByCity endpoint:

Endpoint path: demo/_findCouponsByCity

Parameter: String cityName, String page, String limit

Request example from browser:
http://localhost:8080/demo/findCouponsByCity?cityName=San%20Francisco&page=1&limit=100

Using findCouponsByRegion endpoint:

Endpoint path: demo/findCouponsByRegion

Parameter: String regionName, String page, String limit

Request example:
http://localhost:8080/demo/findCouponsByRegion?regionName=San%20Francisco&page=1&limit=100

Using findCouponsByState endpoint:

Endpoint path: demo/findCouponsByState

Parameter: String stateName, String page, String limit

Request example:
http://localhost:8080/demo/findCouponsByState?stateName=San%20Francisco&page=1&limit=100

Using findCouponsByCountry endpoint:

Endpoint path: demo/findCouponsByCountry

Parameter: String countryName, String page, String limit

Request example:
http://localhost:8080/demo/findCouponsByCountry?countryName=San%20Francisco&page=1&limit=100

Database Structure Design:

When designing the database structure, optimizing for the **find** operation had the most priority. One of the possible designs consisted of having a separate identity for each location information (city, region, state and country) as well as for the coupon and business. This would have caused the **find** query to have 4 **join** SQL operations at the worst-case scenario. Finding by country would require to join the coupon table with the city table, then **join** the result with the region table and so on. The join operation is a costly one, so we should strive to keep its usage at the minimum. The design chosen has only one **join** and is also the simplest design. It consists of having one table with all location entities as columns and two other tables for business and coupon entities. A many to one relationship exist between the coupon entity and the location entity. Every coupon is mapped to a city and also the rest of the location information. In this manner, we can find coupons by only joining the coupon table with the location table and filtering by city, region, state or country.

Hibernate Query Caching and Pagination:

Pagination:

For the endpoints in charge of finding coupons, page and limit variable are used with SQL to achieve pagination. The limit specifies how many coupons to show. The page specifies index (1 based) from which to start displaying the coupons.

Caching:

The hibernate query caching was enabled per query.