# SmartReads REST Service

## Requirements

* Create REST service to return gas and electricity reads for a customer
* Api endpoint : [http://localhost:8080/api/smart/reads/{ACCOUNTNUMBER}](http://localhost:8080/api/smart/reads/%7bACCOUNTNUMBER%7d)
* Response from the service has to be in JSON Format
* Use in memory database for persistence

## Assumptions

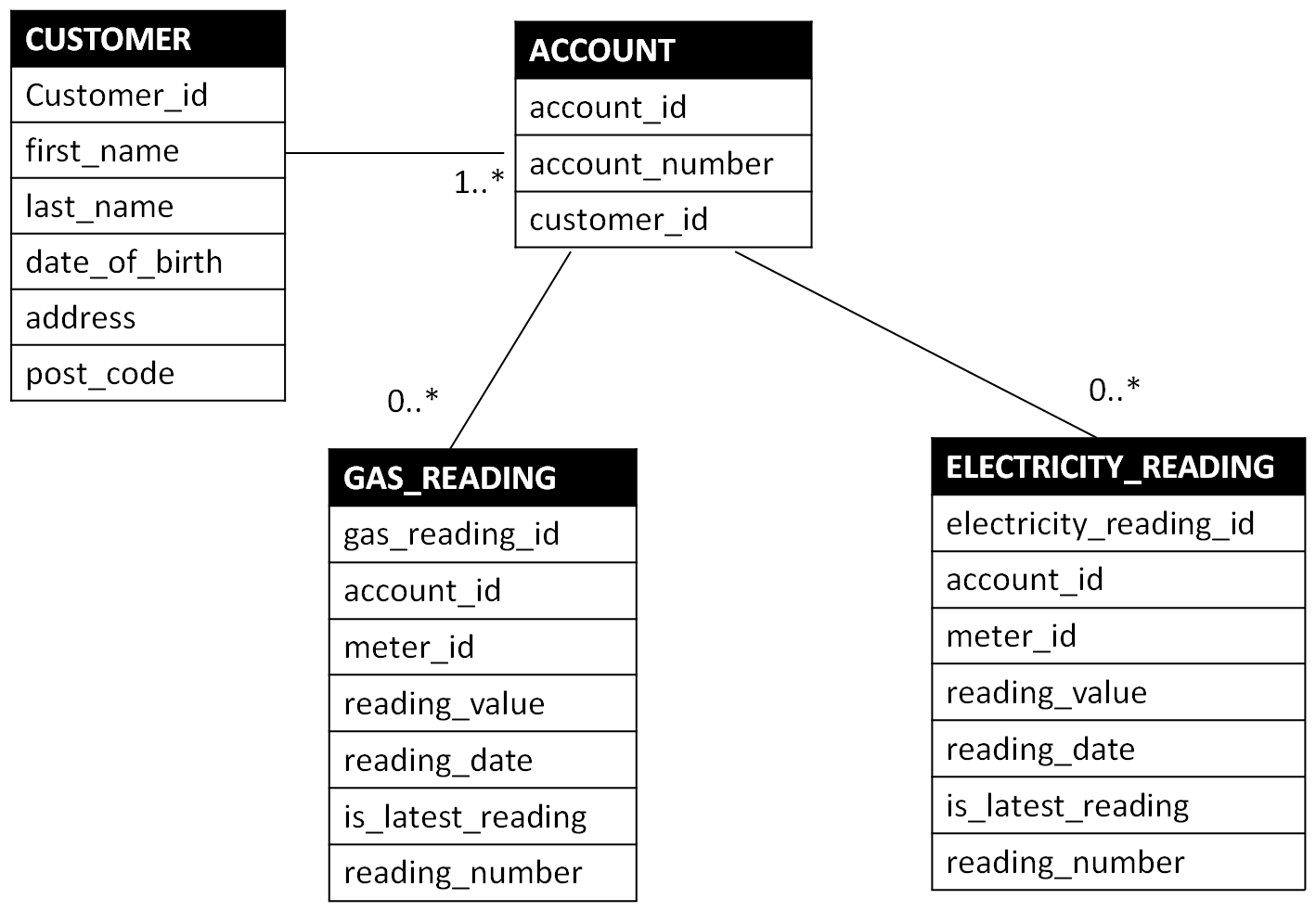
* Single customer will be associated with one or more accounts with the provider
* Electricity reads from a customer meter will be associated to a customer account
* Gas reads from a customer meter will be associated to a customer account
* Each meter will have multiple reads, this will enable service to be expanded to offer all or a set of readings for an account
* When the api endpoint is accessed, service will return the LATEST set of readings
* If api is accessed using invalid accountnumber, return NOT\_FOUND Http Status
* Error response will also be in JSON format

## Code Frameworks and tools:

* Java 1.8
* Spring boot (spring-boot-starter-data-jpa, spring-boot-starter-web, spring-boot-starter-test)
* H2 for in memory database
* Build tool: maven

## Entity Design:

* A customer can have multiple accounts
* An account will have gas and /or electricity reads
* An account can have multiple readings for single meter
* Reading will be versioned using reading\_number column
* Is\_latest\_reading column will indicate the latest reading for the meter

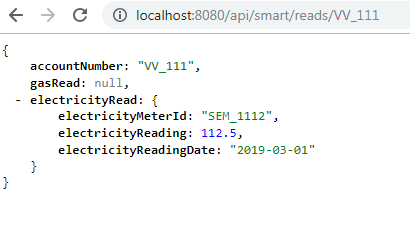


## Using the Service

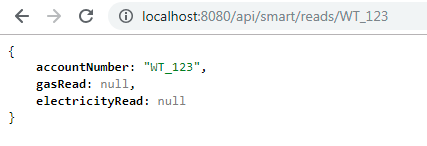
Jar file with all required dependencies has been uploaded in target folder. When application starts, insert sqls added in init\_data.sql will be executed.

**Access application with valid account number**

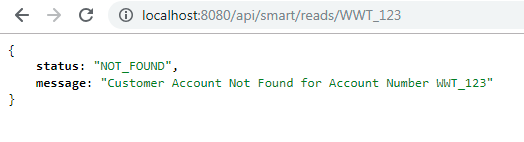
**Access application with valid account number – only electricity reading**



**Access application with valid account number – no readings**



**Access application with invalid account number**



**Access application with invalid URL**

