

java CLOUD ASSESSMENT TOOL

[Document subtitle]



March 4, 2019

capgemini

Capgemini Pvt. Limited Mumbai

Scope: This document contains details about development of cloud assessment tool using JAVA in back end, ANGULAR in front end and MYSQL for database.

Cloud Assessment Tool is developed to assist an application for cloudability. It also provides information for Migration Patterns (REHOST, PUBLIC-PASS and RE-PLATEFORM) and cloud providers (Public, Private).It is a Configurable Application.

It will decide application is cloudable or not on the basis of some questions which will be client specific. It provides some features like:

* Client can add, update and delete questions as well as applications.
* It can have many users where admin can add, update and delete user.
* This tool have a great feature where client can change language as per use means provide “MULTI LENGUAL feature”.
* It provides a feature to generate report of the application if assessment is done.
* It provides “Audit trail”.
* Import export feature makes client feasible to add application.
* Client can add rules for the questions they have created.
* Providers, Platform and cloudability will decide, based on the rule which has been set.

# Requirements:

Java details:

1. Eclipse
2. JRE, JDK should be installed
3. Spring Boot
4. Maven
5. ORM TOOL(Hibernate framework with JPA)

Angular details:

1. Visual Studio Code
2. Angular CLI should be installed in Node.js cmd
3. Node.js

Database details:

1. My-SQL Workbench

# Procedure:

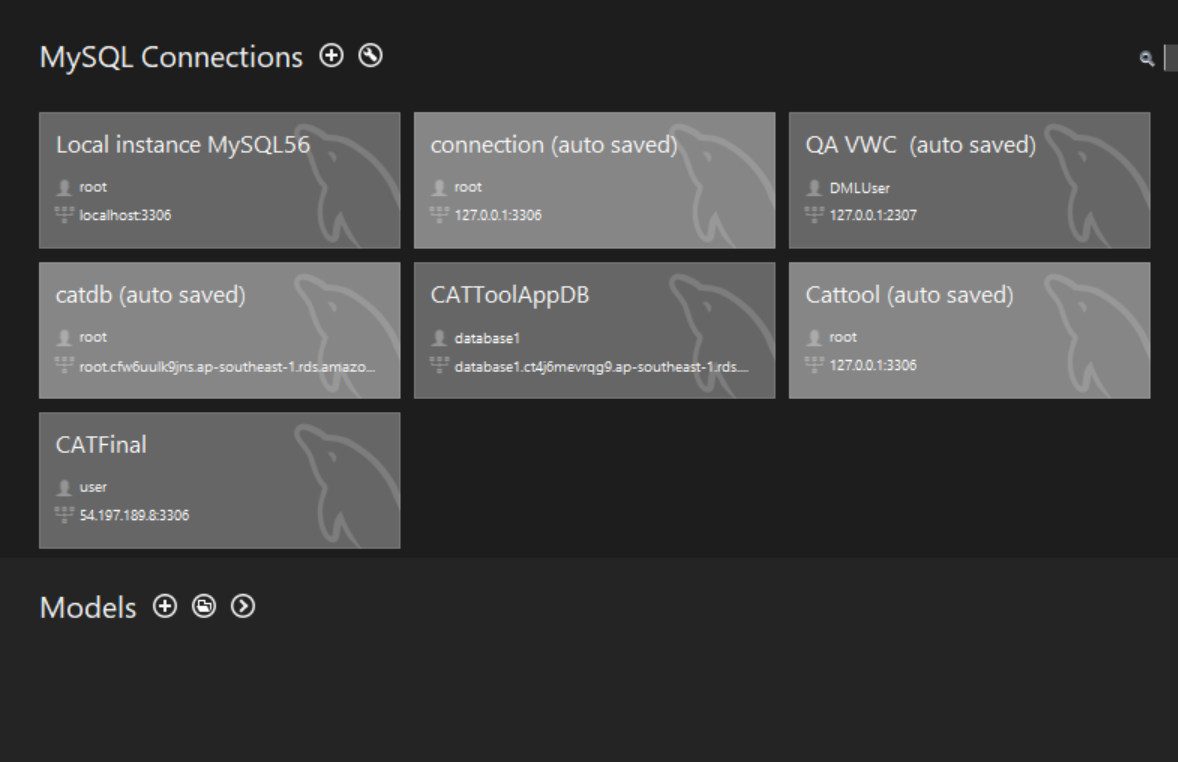
## Database Creation

## Spring boot application creation and implementation

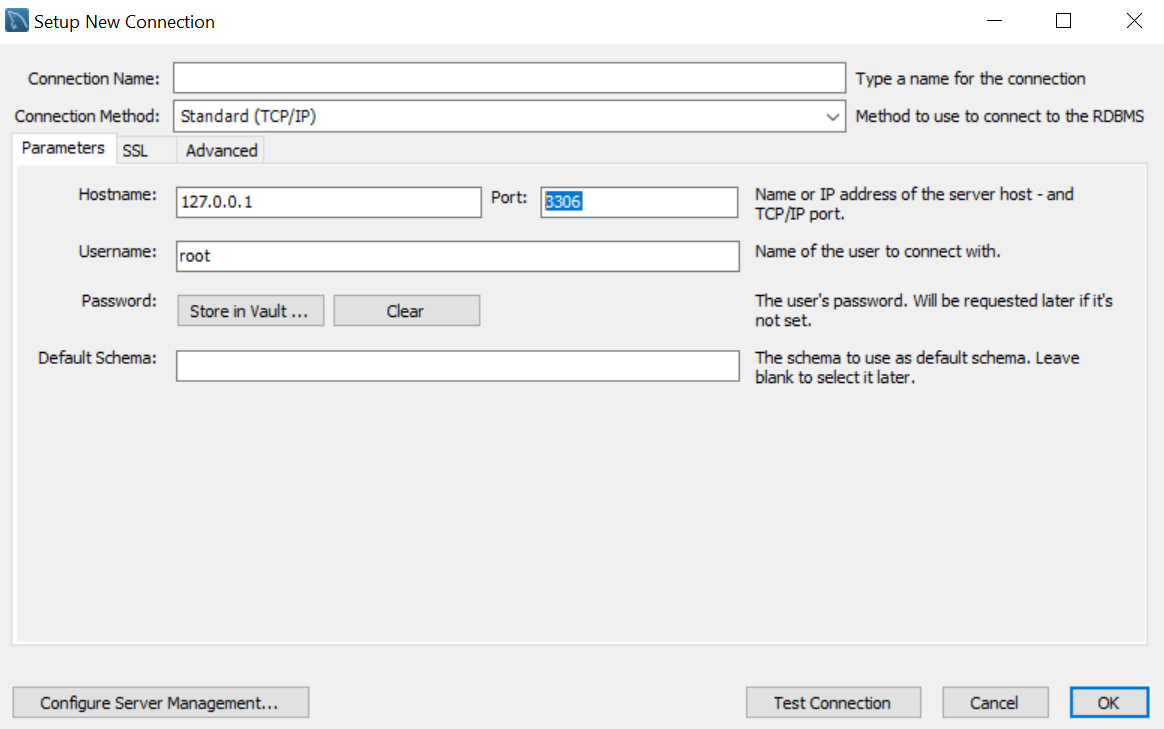
## Angular Project Creation and implementation

# STEP-1: Database Creation

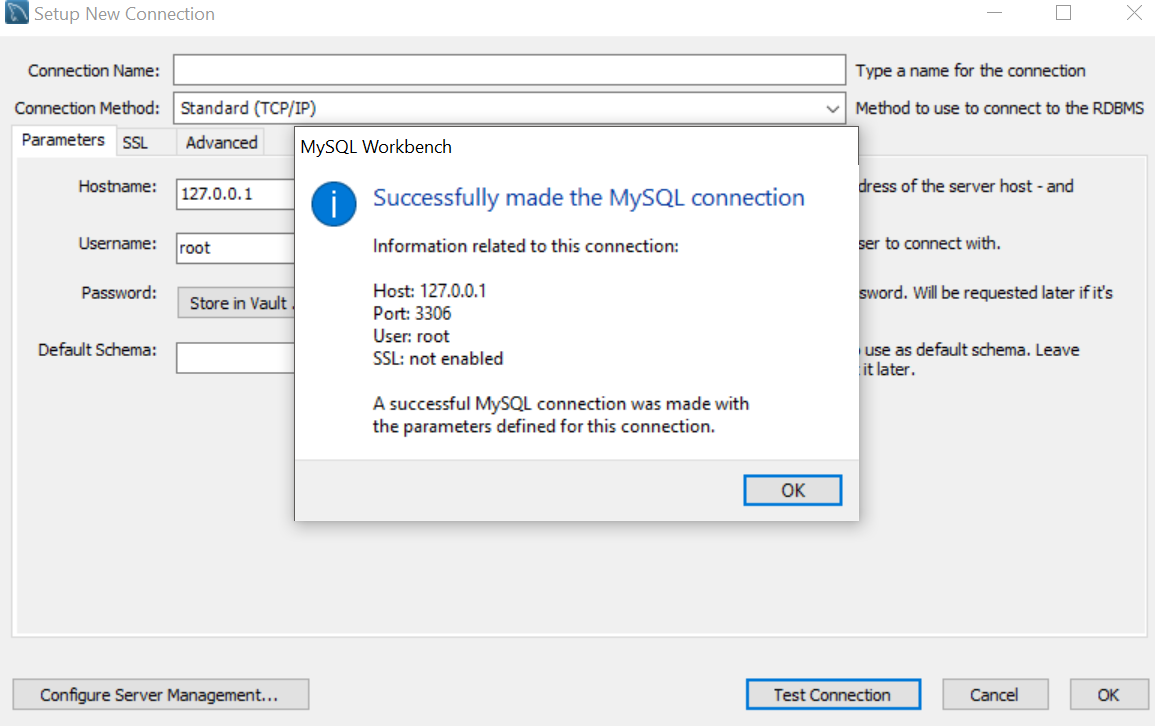
* Open MySQL workbench.
* Click on MySQL connection, click “+”.



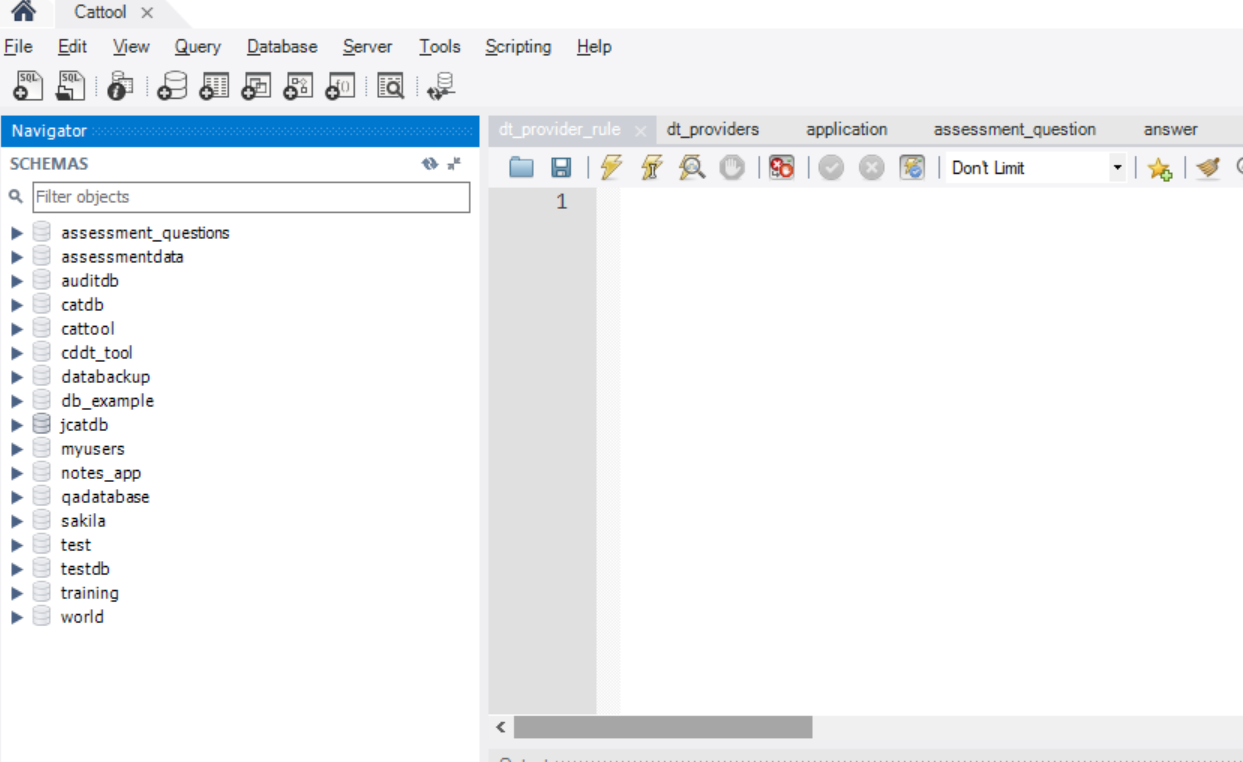
* Then enter hostname and port and the test connection.



* After clicking on “Test Connection” following window prompt. Then click on “ok ”.



* Now you can see you connection has been established and then double click on your connection and it will open your connection. Now create database using “create database <database name>”. And then you will be able to see database name in Navigator.



## STEP-2 Spring boot application creation and implementation

* Go to the site

<https://start.spring.io/>

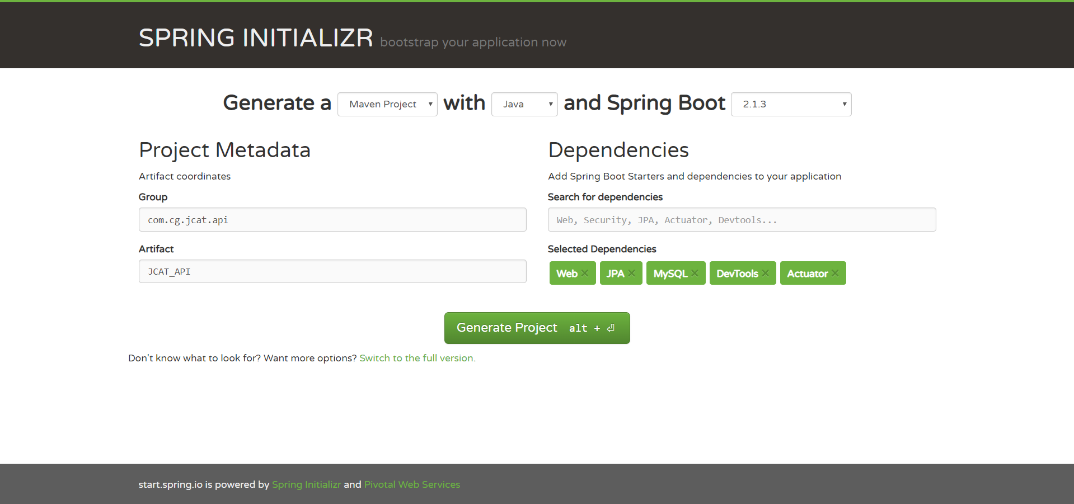
* Fill the following details for project creation.

Group-name: Package name

Artifact: Project name

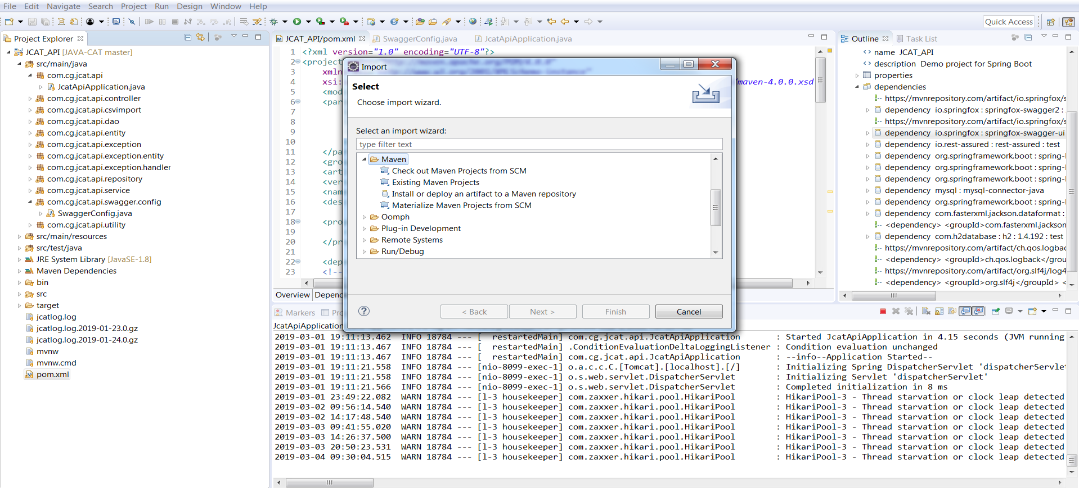
Search for dependency: Web, JPA, MySQL, DevTools, Actuator

Then click on generate project it will download a zip file.

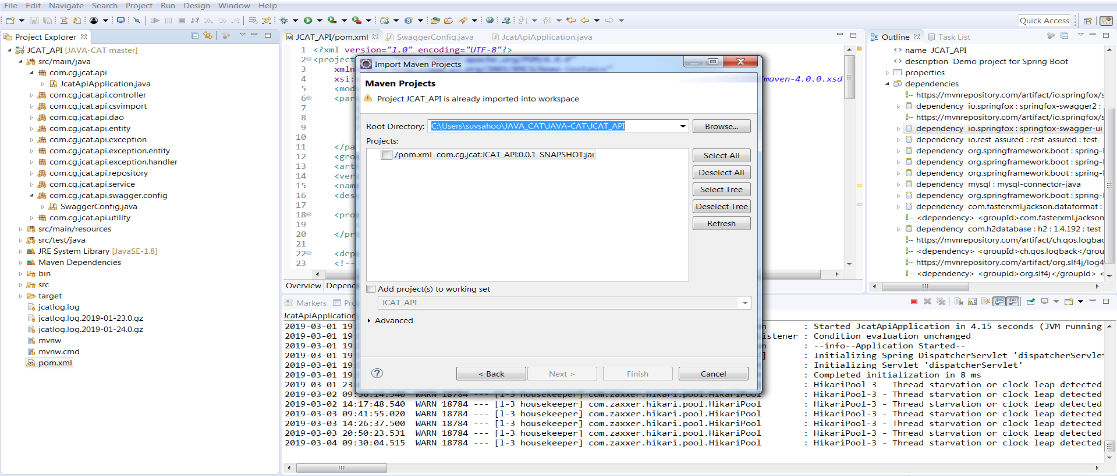


* Then open eclipse then click on “File” and then choose “Import project ”

And where you have to select “Existing Maven Projects” as shown below



* Click on “Next” and specify path from where you want to import the project in the root directory section and click on finish. Then you will able to see your project in “Project Explorer”



### Configuration of Project

server.port: Assign port where you want to run projects

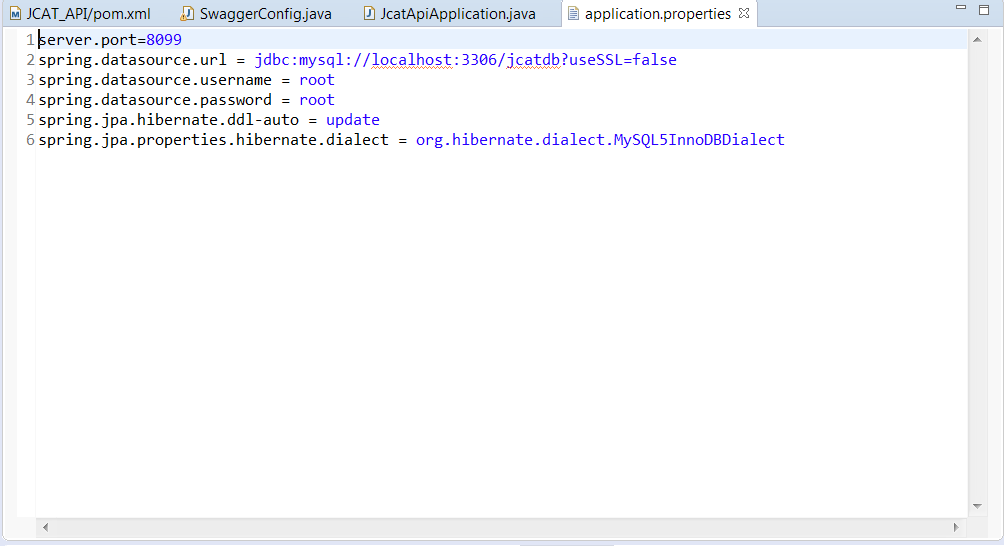
datasource url: Database server url

datasource username and password: mysql username and password

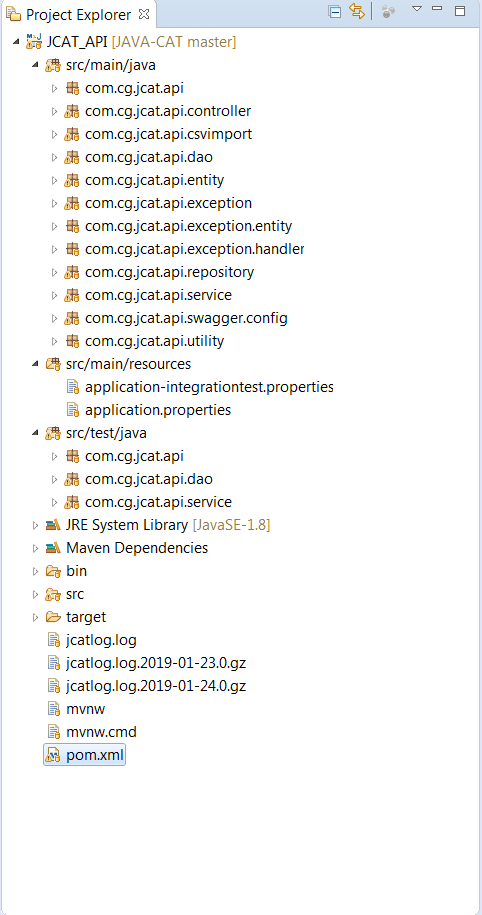
datasource ddl-auto: update query first time will create table once and then update table.

Hibernate.dialect- use mysql dialect.

Other than it logger and other details can specify here.



## Folder Structure:

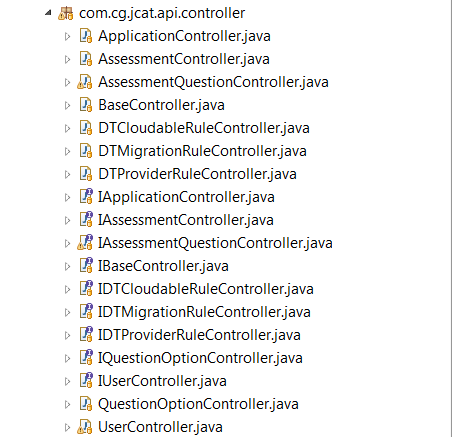


### Package 1: com.cg.jcat.api

Under this package spring boot creates a main class which will responsible to run project.

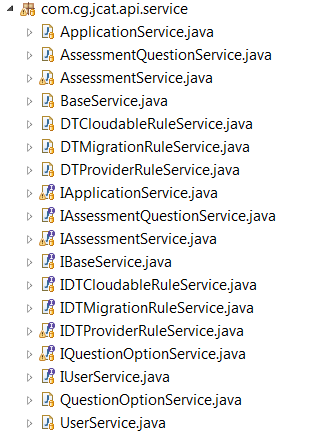
### Package 2: com.cg.jcat.api.controller

This package contains all the controller classes and interfaces. Controller handle http requests. When API will call by UI, directly request come in controller. Controller interface talks with http request which internally calls controller classes. Then controller class call service interface.



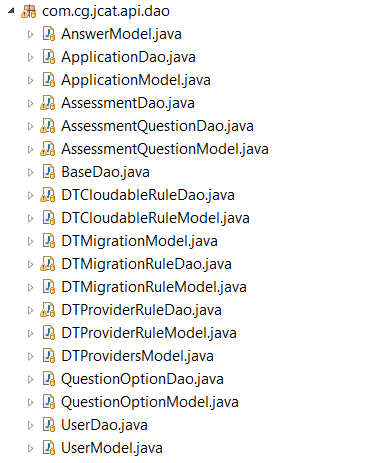
### Package 3: com.cg.jcat.api.service

The package contains validation and business logic and service interface is called from controller classes and its method implements by service class. Further service class calls Data access object layer.



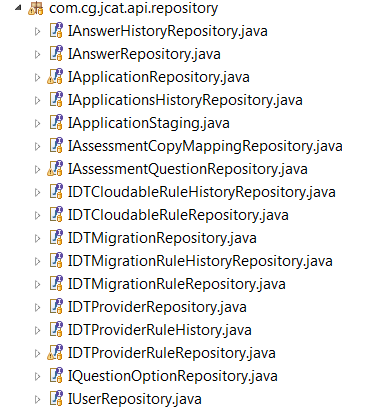
### Package 4: com.cg.jcat.api.dao

The package contains dao class which communicate with repository and it also contains model class which is a partial implementation of entity class for sending data to the user. Dao class converts entity object to model and vice versa. Means this layer will retrieve data from database. DAO class called from service class and dao class will call repository interface to retrieve data from database.



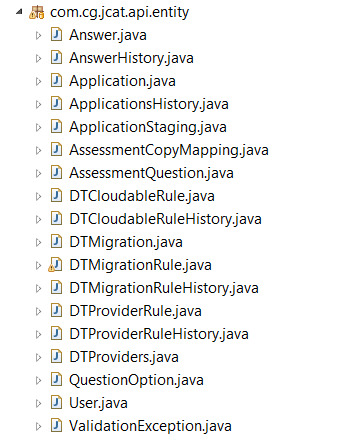
### Package 5: com.cg.jcat.api.repository

This package contains all repository interface which map java object with database object. Where jpa repository uses hibernate framework for mapping the objects. This repository interface called from dao(data access object) class and retrieve data from database.



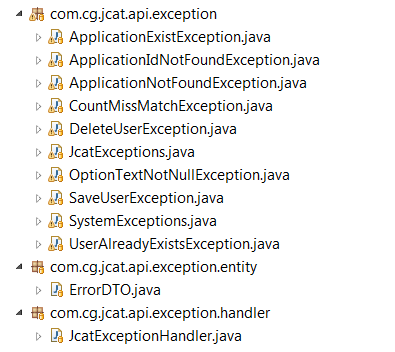
### Package 6: com.cg.jcat.api.entity

This package contains all entity which is POJO (Plain Old Java Object) class which has some attributes which describe behavior.



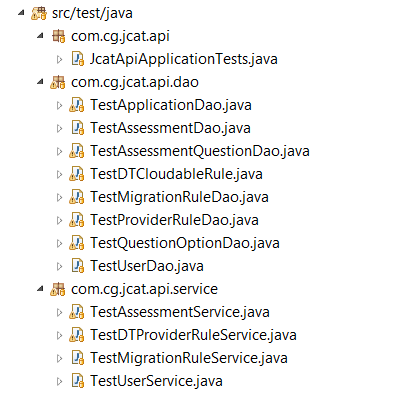
### Package 7: com.cg.jcat.api.exception

This package contains all the exceptional classes which will be thrown from the different classes. Exception is the base class of JcatException. ErrorDTO class contains all the fields which will be shown to the user when exception occurs. Exception handler class is the global class for handling all the exceptions thrown by the different controller.



## JUNIT test cases:

JUNIT test cases used to test method which implemented in class. For services mockito object is used means provide a dummy data for testing and for dao it uses h2 database for testing where h2 database’s configuration is given in application-integrationtest.properties.



### Package 1: com.cg.jcat.api.service

It contains test cases for all service classes.

### Package 1: com.cg.jcat.api.dao

It contains test cases for all dao classes.