Simplecab

This module intentions is to:
* Cab Data Researcher is a company that provides insights on the open data about NY cab trips * Service to process query cab trips from csv.

How to run the program

Run MYSQL Database

1. Run mysql database
   - mysqld.exe

Run Simple Cab Server and load csv data

1. Run Simple Cab Server
   - java -jar simplecabserver.jar
   - curl -X GET "http://localhost:8080/cab/loadCSV?filepath=E%3A%2FTEMP2%2Fcabdata2.csv" -H "accept: */*"

Run Simple Cab Client

1. Run Simple Cab Client java -jar simplecabclient.jar -h <== display help

- usage: Main-d,--deleteCache Delete cache.-f,--inputFileName inputFileName with ids.-h,--help show help.-i,--ignoreCache Ignore cache.-m,--medallionIds medallion ids.-p,--pickupDate Pickup Date yyyy-MM-dd.
- java -jar simplecabclient.jar -h
- java -jar simplecabclient.jar -d
- java -jar simplecabclient.jar -i -p 2013-12-01 -m D7D598CD99978BD012A87A76A7C891B7 5455D5FF2BD94D10B304A15D4B7F2735
- java -jar simplecabclient.jar -p 2013-12-01 -f input.txt

Continous Integration and Code Coverage

- Continous Integration
- Code Coverage
- Codacy

Relevant Articles:

- How to Quickly Load 380K Items Into MySQL
- Importing CSV using LOAD DATA INFILE quote problem
- Simple apache commons CLI

What is this repository for?

- 1.0.0

How do I get set up?

- Summary of set up
- Configuration
- Dependencies
- Database configuration
- How to run tests
- Deployment instructions

Contribution guidelines

- Writing tests
- Code review
- Other guidelines

Who do I talk to?

- Repo owner or admin
- Other community or team contact

Building The Project

Eclipse:

1. Before importing into eclipse execute the `mvn eclipse:eclipse` command to prepare the project to be able to be simply imported into the IDE.
2. Within the IDE, select `import existing eclipse project` option and as the eclipse:eclipse command will have prepared **.project** and **.classpath** files, this will mean minimal setup required.
3. Add the integration-test src and resources to your classpath to ensure your can easily execute integration tests from the IDE.

Compiling

The project has been designed to be built via `maven` the pom has been configured to be able to execute unit and integration tests, generate code coverage reports, and code checkstyle testing.

Four Test Profiles have been configured to aid in building the project depending on what you wish to achieve:

1. `unit-tests` will only execute unit tests.
2. `integration-tests` will only execute integration tests.
3. `all-tests` both unit and integration tests are run.
4. `no-tests` no tests run.

The default profile is **all-tests**.

Some example executions are as followed. * mvn *clean test* => Creates code coverage report for unit tests. * mvn *clean verify -P unit-tests* => Creates code coverage report for unit tests and fails if does not meet coverage ratios. * mvn *clean verify -P integration-tests* => Creates code coverage report for unit tests and fails if does not meet coverage ratios. * mvn *clean verify -P all-tests* => Creates code coverage reports for unit and integration tests and fails if does not meet coverage ratios). * mvn *clean package -P no-tests* => Does not execute tests and creates the distributable. * mvn *clean verify -P all-tests checkstyle:check* => runs all tests, checkstyle and packages.

Before submitting your code to GIT run **clean verify -P all-tests checkstyle:check** and build successfully to ensure that all tests and code style checks are completed to ensure Continuous Integration build errors do not arise.

Test Profiles

no-tests - all tests are ignored all-tests - all tests are executed unit-tests - only unit tests are executed integration-tests - only integration tests are executed