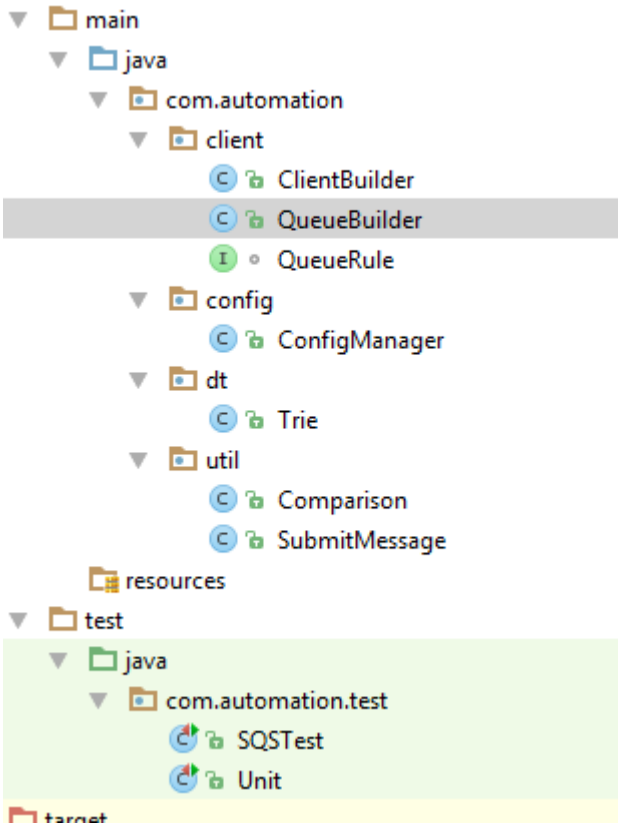
Framework Design Overview

This is a tiny automation framework which I design for amazon SQS testing.

Framework Structure



The framework is written in Java 8 together with maven and testNG ,it contains 4 core packages (client, config,dt,until) and 1 test package

- ***ClientBuilder is a thread safe class, no matter how thread access to it, it guarantees only one instance of AmazonSQS created. This design can have better performance in memory.***

- ***QueueBuilder also follows singleton pattern, which creates instances of standard queues and FIFO. I am using a Map and Trie(Prefix Tree) structure as in-memo cache to speedup look up by queue name and queue URL. Also, if a queue is created and its instance will be cached, so the framework will not create it again but just return its instance from the cache. Again, the purpose of this design is still for better memory usage.***

- ***ConfigManager is designed to read configurations from an external XML, I am using 4 parameters for the project but users can add their owns customization parameters. The purpose of this class is to let users easily switch to different test environments***

- ***The dt package's full name is data structure, which contains an optimized prefix tree. The structure has better memory usage in storing URLs and its lookup's speed is very closed to hash structure. Again, the purpose of this design is still for better memory usage.***

- ***Comparison provides utilities for test assertion and verification.***

- **SubmitMessage will emulate multiple clients accessing to one queue instance, which is for performance testing.**

Unit Test

- ***Unit contains my unit tests of the framework. It currently contains a test which verifies my prefix tree structure.***

Functional Test

I designed 6 scenarios for the project

- ***testSingleMessage_send_receive is for basic functionality test for sending and receiving message***

- ***testSingleMessage_send_receive_delay is for testing delay functionality.***

- ***testSingleMessage_send_receive_delay_without_wait is a negative test on delay functionality***

- ***TestSendBatchMessages is for batch message functionality testing***

- ***TestSendMultipleMessages is for throughput testing single client with massive messages***

- ***multi_thread_connection_sending is for throughput testing multiple clients with massive messages***

What I would do if I were given more time

- ***Add more unit tests for the framework, the current version is not fully tested, especially for the two thread safe classes. I believe a robust and efficient framework will determine the success of test automation.***

- ***Read and understand more on Amazon SQS API documents, By doing this ,I can add more tests to ensure coverage.***
- ***Add more functionalities to the framework, like data driven, result verification... and try to make it as a generic framework.***