**Client Design**

My client package is an extension of the swagger auto-generated package **io.swagger.client**. In this package there are auto-generated classes for making api calls such as ApiClient, JSON and There are 6 packages.

**api** (auto-generated): api clients for calling the apis.

**auth** (auto-generated): authentication for calling apis.

**model** (auto-generated): data models defining the format of request and response data

**setup:** contains one class **ParamParser** for getting the number of threads, number of skiers, number of lifts, number of runs and server address from command line arguments.

**runner:** this package contains three classes for executing a multi-threaded application. There are 2 classes MultiThreadApp and Runner. **Runner** implements runnable and define how each thread makes requests and store the running metrics for each request. **MultiThreadApp** contains the main method and is responsible for the general workflow. It contains methods to start the service for 3 phases, print out the metrics and write records to csv file after all threads complete.

**measurement:** contains one class **Metrics** which is used to store the running result including number of successful requests, number of failed requests, a list of latency and a list of records for each request including the start time, response code, request type and latency.

**How to run client**

Run the main method in class **MultiThreadApp** in package **runner** with 5 arguments: number of threads, number of skiers, number of lifts, number of runs and server address.

Arguments requirement:

number of threads: integer between 1-256

number of skiers: integer between 1-50000

number of lifts: integer between 5-60

number of runs: integer between 1-20

server address: string

example:

32 20000 40 20 http://ec2-54-218-250-206.us-west-2.compute.amazonaws.com:8080/upicServer\_war