**ITEC 4860  
Dr. Im**

REST Project Part 1

In this project, you are to implement the following REST API and client code to call your API.

PART 1)

Implement the following REST API:

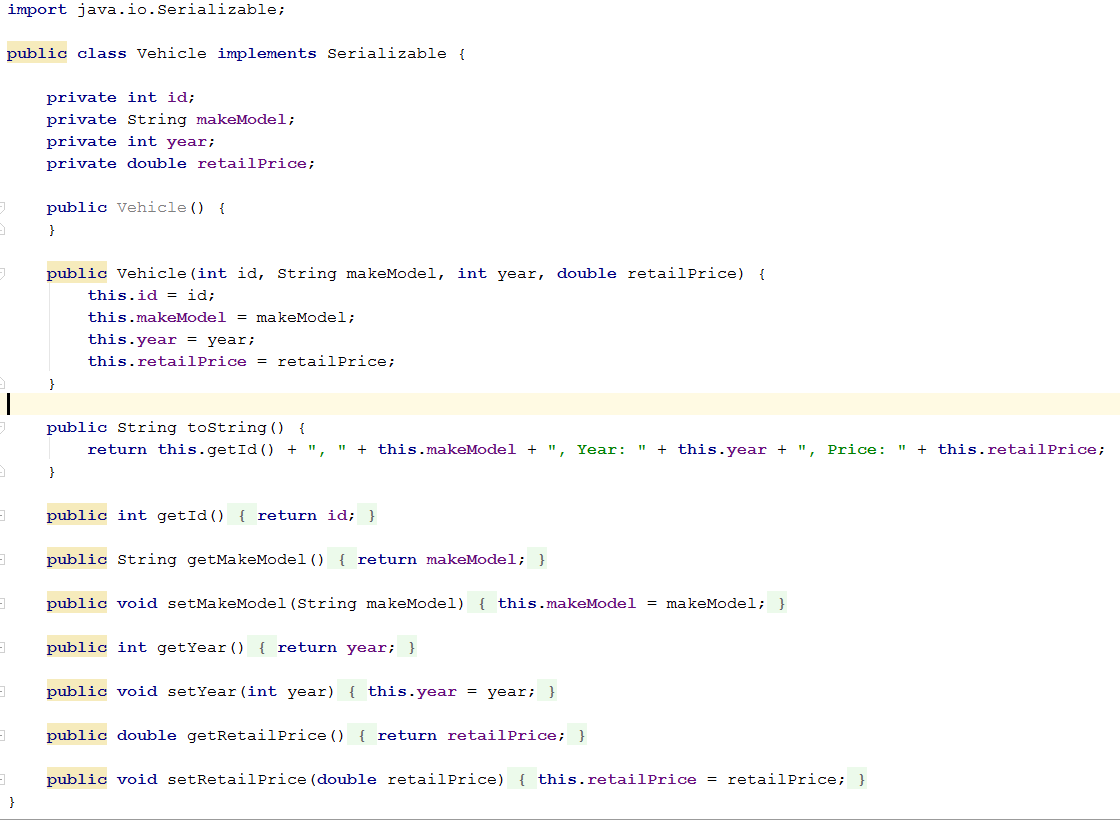
* 
  + addVehicle() will take a vehicle object and write it to a local text file.
  + It will always APPEND to the file.
  + So if I make 5 POST requests to /addVehicle, the local file will contain 5 vehicles in JSON.
* 
  + getVehicle() will take a given id, and find the vehicle that has the matching id.
  + It will iterate the local file line-by-line, check if the id matches, and if there is a match return the vehicle object.
* 
  + updateVehicle() will do the following given a vehicle object passed in:
    - Iterate the local file line-by-line
    - Check if the current line’s vehicle’s id matches the vehicle id that is passed in
    - If there is a match, update the current line with the vehicle that was passed in
* 
  + deleteVehicle() simply takes the given id and deletes from the local file.
  + It will iterate the local file line-by-line to check if the given id exists, then perform a delete.

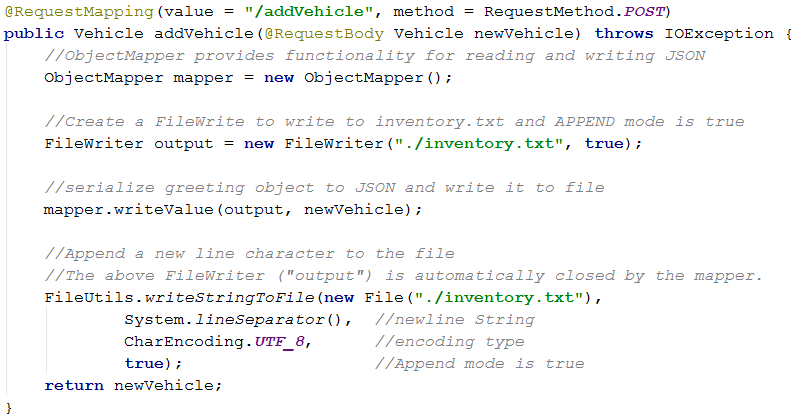
PART 2) Implement client code to call your REST API.

Create a file called MyTasks.java and implement the following @Scheduled jobs.

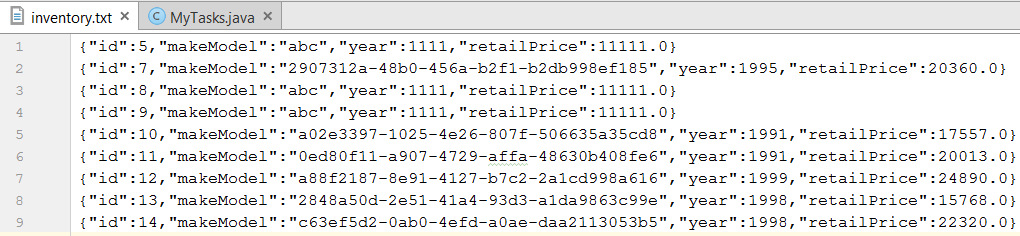
* @Scheduled(some periodic rate)  
  **public void** addVehicle() {
  + At some periodic rate, make POST request to add a vehicle.
  + Simply create a vehicle with random values and write it to file.
    - Vehicle Id should start from 1 and increment by 1 each time.
    - Vehicle makeAndModel should be a randomly generated string.
    - Vehicle year should be a random number between 1986-2016.
    - Vehicle retailPrice should be a random number between 15000-45000.
* @Scheduled(some periodic rate)  
  **public void** deleteVehicle() {
  + At some periodic rate, make DELETE request to delete a vehicle.
  + Generate a random id, with some reasonable range (0-100)
  + Then make the DELETE request.
* @Scheduled(some periodic rate)  
  **public void** updateVehicle() {
  + At some periodic rate, make PUT request to update a vehicle.
  + Create a new vehicle object with random values
    - Id should be some reasonable random number between 0 – 100
      * The id should be likely to exist in the file already
    - makeAndModel, year, and retail price can be hard-coded with some special values in this case to easily identify that you actually updated the vehicle.
  + After the update, make a GET request on the same id
    - To verify that you actually updated the vehicle

Provided Code)





Sample File Contents)



Id 5, 8, and 9 shows updated vehicles with some hard-coded values to easily identify they were updated.

The rest of the line shows randomly generated vehicles added to the file.

Id 1, 2, 3, 4, and 6 are now shown because they were deleted through the DELETE request.

Submission Guidelines:

1. Take screenshots of your output, paste it into Word, and upload to D2L.
2. Compress the entire directory of your project into a zip file and upload it to D2L
3. Relax and enjoy a cold drink ☺