**TEST PLAN**

* Functional testing / main requirements:
  + Test that the API works as intended.
  + Post vehicle, ensure correct status code and response is received
  + Delete vehicle, ensure correct status code and response is received
  + Get odometer alerts to view status on scheduled maintenance
* What to check:
  + HTTP status code
  + Response payload is JSON object
  + Data accuracy
  + HTTP headers
  + API Response time
* Inputs
  + “Positive” valid requests (Should pass w/ valid input)
    - Expected status codes: 2xx
    - On DELETE, vehicleID should no longer be found by a following GET request of the same ID
  + “Negative” requests (Should fail w/ valid input & invalid input)
    - Expected status code: NOT 2xx
    - Verify error response is received
    - Error message should be clear, descriptive
* Testing methods:
  + POST vehicles with ID numbers > 0
    - Expected response code: 201
    - Check inputted vehicle ID against response body vehicleID return value
    - Check that the vehicle enrolled value is true
    - Perform GET request on vehicleID to ensure it was properly posted
  + POST vehicles with ID numbers <= 0 OR non-valid ID
    - Expected response code: 400 for bad request
    - Check that response body is in valid JSON format or string
    - Error message should explain the error
  + DELETE vehicles with ID numbers > 0
    - Expected response code: 204
    - Check that response body is empty or confirms deletion status in JSON format
  + DELETE vehicles with ID numbers <= 0
    - Expected response code: 404 for resource not found
    - Check that response body is in valid JSON format or string
* Manual testing:
  + POST an invalid vehicle to monitor returned error code and assess clarity of error
  + DELETE an invalid vehicle to monitor returned error code and assess clarity of error
  + Use a test device to mock service alerts at different odometer thresholds.
  + Verify that alerts are only sent at the threshold when the user is enrolled in service alerts
  + Verify that front-end UI matches up with POST/DELETE requests