GiG Code Challenge

For Big Data Engineer in Test

Overview

This is a code challenge for prospective Big Data Engineers in Test to work for Gaming Innovation Group.

It is expected that all points are addressed.

This challenge covers:

- Development of a small test automation framework in C#
- Behaviour Driven Development (BDD) using SpecFlow
- RESTful API Testing
- Virtualization using Docker
- Stream processing using Apache Kafka

Deliverable

You are expected to develop a layered test automation framework aimed at testing RESTful APIs and stream processing technologies. The developed framework should include feature files and step definition classes. The framework could also include any other layer you deem appropriate (object models, configuration files, wrapper classes, helper classes, etc.)

Expected Goals

- Design your test automation framework in a way generic enough to be able to plug any other kind of test into it.
- You should strive to make the framework as flexible as possible and aim to have tests as easily maintainable as possible.
- Code compiles and all automated tests to execute and pass.
- Push (regularly) all the code to a GitHub repo.

Task 1 – RESTful API tests

Use the provided API at https://reqres.in/ for all tests in this section. All request payloads and responses are in JSON format.

Registration

1. Successful registration
    - POST on https://reqres.in/api/resgister
    - Payload: email and password
    - Response: 200 along with a token
2. Unsuccessful registration
    - POST on https://reqres.in/api/register
    - Payload: email
    - Response: 400 along with an error

Get User List

3. List users
    - GET on https://reqres.in/api/users
    - No payload
    - Response: 200 with list of users

Task 2 – Stream processing tests

Setup

- Install Docker virtualization tool on your machine
- Pull an Apache Kafka Image
- Have a Docker Container running the downloaded Apache Kafka Image

Produce Messages

- Produce a number of messages on Kafka Topic with name ‘cars’. Message should contain details on a Car as follows:
    - Brand name
    - Model
    - Number of Doors
    - Indicating whether it is a Sports car or not

Consume messages

- Consume previously produced messages on Kafka Topic with name ‘cars’.