Trading Technology SDET Assessment

Version 2022-03-18 13:48:44.

In order to demonstrate your comfort and proficiency completing the types of tasks you will need to complete on a daily basis in this role, please create, and **return within 3 days of receipt of this document**, a test project that runs some tests against the public Kraken WebSockets API. No Kraken account is needed as public-data feeds are open to the public and do not require any authentication.

The code can be written in a programming language of your choice.

You should aim to include a total of at least 10 test cases over at least 3 channels (public channels include book, OHLC, spread, ticker, and trade). Examples of test cases include the order book not being crossed for some time, timestamps strictly increasing over time on feeds, schema validation, input validation/errors, etc. but bonus points will be given for test case ideas of your own devising. One way to come up with test cases is to imagine that this is a legacy API service that has no existing tests and that you have been tasked to quickly write a regression test suite that can be used to capture some existing behavior and help ensure against breaking functionality as the code is radically refactored. The subscription options also might provide a rich source of ideas.

Do not attempt to test for denial-of-service vulnerabilities by sending very large requests, large numbers of connection requests, etc., or otherwise attack the API endpoints.

Requirements

1. This project must be completed by you alone and must be submitted within 3 days of receiving this document.
2. Do not use Kraken API clients written by others from GitHub or anywhere else. The project should be written by you from scratch. Use of a basic WebSockets protocol library (such as IXWebSocket, gorilla/websocket, mongoose, websocket-client, etc.) and a basic test framework (such as pytest) are permissible, but aside from this anything outside the standard library generally should be avoided.
3. You must include a Dockerfile, including an entry-point command that allows us to execute the tests by building and running the Docker image. Test execution should print results of individual tests, including basic details of assertion failures and overall pass/fail result, as would be done in a CI pipeline.

Deliverables

Please submit a single compressed archive file (zip, gzipped tarball, or something similar) and be sure to include the following contents:

1. A top-level directory containing your source code
2. A Dockerfile capturing any dependencies and allowing for trivial creation of a ready-to-run test image.
3. A README file, in a format of your choice, that includes usage instructions.
4. A text file, NOTES.txt, that contains any written remarks as desired. This file should include a brief description of the tests you wrote, and optionally might include other items like a summary or high-level overview of the project, any notes about difficulties encountered, bad Kraken API documentation the hindered your work, and so forth.
5. If you worked on this task within a version control repository like Git, bonus points will be awarded for including repository metadata (.git directory, etc.) so that we can see your development timeline and commit messages and practices).

Good luck!