Effective Testing with API Simulation and (Micro)Service Virtualisation

Bonus Module Six: Load Testing and Operations Testing

Purpose of this Lab

- Practice developing simulations for load testing
- Practice developing simulations for operational-style testing

Questions

Feel free to ask questions and ask for help at any point during the workshop. Also, please collaborate with others.

Exercise 1: Load testing using Hoverfly

During this exercise, we will explore:

- Simple load testing with the Apache Benchmark “ab” tool
- Using Hoverfly to simulate and control downstream dependencies within the load test

Steps

1. Examine the code for the flights service that we have been using within the previous modules, particularly the Controller classes:
https://github.com/SpectoLabs/api-simulation-training/tree/master/flights-service/src/main/java/io/specto

2. Note that there is a CommentsController that can be accessed via “api/v1/comments”. The code is here:

https://github.com/SpectoLabs/api-simulation-training/tree/master/flights-service/src/main/java/io/specto

3. Start the flights api, either by compiling (with “mvn clean install”) and running with “java -jar <flights-api-jar-location”, or by using the utility scripts and pre-compiled JAR from a previous exercise. Curl the comments API and examine what is returned.

4. Imagine we wanted to load test the comments API endpoint (which is dependent on a third-party service). Can you see any problems?

5. You can run a simple load test with the Apache Benchmark tool “ab”, which can be installed easily in Mac OS or Linux. Explore the help options “ab --help” and try and run a load test on the Comments API endpoint. Experiment with the number of requests and concurrency. Do you see any errors coming back from the third party service?

6. Use Hoverfly to capture data from the third-party service, and configure the Flights API Java application to talk to your simulation. Run the load test again, and see if you can break Hoverfly.

7. Attempt to add delays or middleware to the Hoverfly third-party comments simulation, and observe what happens with the ab tool.

Exercise 2: Testing Ops with Hoverfly

During this exercise, we will explore:

- How to simulate operations/cloud APIs

Steps

1. Explore the localstack framework, a simulation of Amazon Web Services (AWS) APIs like S3, DynamoDB, and Route53

2. Pick another cloud provider (or simply make one up), and attempt to simulate their key/value (Dynamo-like) service API or DNS service API or. You’ll most likely need to explore the use of state

3. Explore how suited Hoverfly is to this type of simulation, and the type of tests you may like to run on each service e.g. how can you capture or create traffic from these APIs, and how large/complex is the data being captured?

4. Sometimes it easier to write middleware that simulates operation API endpoints. Explore creating middleware that simulates a blog storage (S3-like) service API, and store the corresponding files within the tmp directory of your local machine.

5. Discuss what types of APIs would be beneficial to simulate using this method.

- Hint: think about the trade-off between accidentally re-writing the service within middleware.
- Is this type of simulation good for APIs with complex server-side processing but simple state management and outputs, like a cloud DNS system or financial trading API
- What about simple server-side processing but complex state management, like a key/value store query API or CRUD-like REST API?