Effective Testing with API Simulation and (Micro)Service Virtualisation

Module One: Setup and Installation

Setup Instructions

Currently the course requires either Linux or a Mac. **We aren't going to support Windows**, so if you have a Windows machine we would recommend running Linux in VirtualBox. You will need to install the following dependencies (We have also created a VirtualBox image with everything already installed and pre-configured here -- username and password "spectolabs"):

1. A Terminal running Bash. If you're on a Mac, iTerm is a good option which can be found here.
2. A Text Editor. We use Atom in all our examples, which can downloaded here.
3. Hoverfly, for which installation instructions can be found here.
4. We also use JQ to more easily format JSON in the terminal, which can be found here.
5. For some middleware exercises we will use Python or NodeJS. If they are not already on your machine, download them here and here.
6. Java is required to executed some jar files. It's found here.
7. Git, in order to be able to clone the repository which can be found here.

Once you are ready, you can clone the repository (It will not be made public until the morning of the workshop)

git@github.com:SpectoLabs/api-simulation-training.git

Tips for Using the VM

If you decide to use the VirtualBox VM here are some tips and potential issues to watch out for:

- You can configure the hardware available to the VM (CPU cores and memory etc) be right-clicking the api-simulation-training-2018 VM in the VirtualBox Manager, and selecting "System". We would recommend 2-4 cores (or more) -- don't forget that even if

you “only” have a four core machine, the use of HyperThreading and other hardware tricks often means that 16 logical “cores” are available to your OS
- You can install the VirtualBox “Guest Additions” utility -- which enables two-way copy and pasting -- by clicking on the “device” menu of the VirtualBox VM and selecting “Insert Guest Additions CD Image”. You will need to follow installation instructions.
- We have configured the VM to use a “English (US)” keyboard. If you are not using this type of keyboard you can change this by clicking on the “Settings” icon in the Ubuntu dock (it looks like a cog with a wrench), and selecting “Keyboard”. You can configure your keyboard from the window that is displayed
- Be aware that copy/paste from a PDF to the terminal does not always work as expected. If you see a bizarre error then you may need to type the commands directly (such, as “hoverctl” not being recognised, double-quotes not being recognised, and pipes and filters not working -- | > & etc)

Course Abstract

Testing microservices is challenging. Dividing a system into components naturally creates inter-service dependencies, and each service has its own performance and fault-tolerance characteristics that need to be validated during development and the QA process. Join this one day workshop and learn the theory, techniques and practices needed to overcome this challenge.

- Introduction to the challenges of testing distributed microservice systems
- Breaking the reliance on dependent services and APIs
- A practical guide to API simulation
- Isolating tests within a large microservice ecosystem
- Implementing fault-injection testing to validate non-functional requirements
- The use of API simulation for testing work undertaken during dev/ops, legacy system and cloud migration, and high-volume load testing

Course Outline

This course will be divided into multiple modules. Generally, each module contains:

- Presentations
- Live Demos and practical Exercises

Questions

Any questions, please email andrewmorgan36@gmail.com or daniel.bryant@tai-dev.co.uk