Google Cloud

Edge Unit Testing

API Proxy Testing



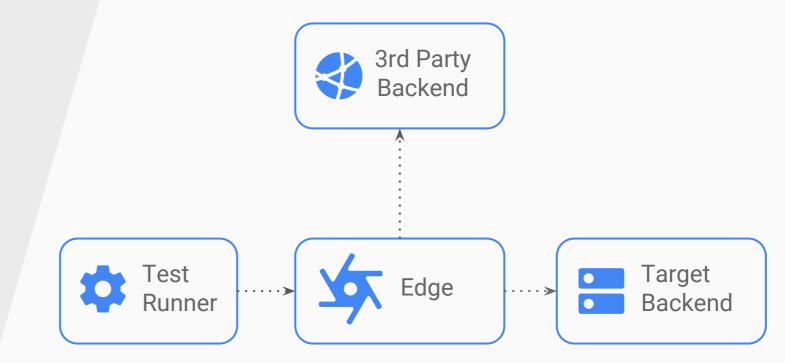
Do you think it is possible to test an API proxy just with integration testing?

Is Integration Testing Enough?

Service Callouts?

Script Callouts?

Async Operations?



Main Use Cases for Unit Testing



- Operations that our integration testing cannot intercept and therefore cannot assert.
 - Service callouts to external APIs (e.g., Loggly).
 - Testing IP blacklisting / whitelisting need to simulate different IP addresses coming in to Apigee with many proxies in the middle.
- Very important code (e.g. security code, encryption code, signature generators/validators)
- Where coverage is extremely important, e.g. security code (again)

Unit Testing Benefits

- Code can be tested locally without deployment to Edge first.
- Can create hooks to enforce testing during commit.
- Much faster than integration testing.



Unit Testing - Boundary Principle



"Test within your boundaries, don't test libraries you don't control,

The Edge Team

Unit Testing – Types of policies

Candidates for Unit Testing

Traffic Management

Quota

Reset Quota

Spike Arrest

Concurrent Rate limit

Response Cache

Lookup Cache

Populate Cache

Invalidate Cache

Security

XML Threat Protection

JSON Threat Protection

Regular Expression Protection

OAuth v2.0

Get OAuth v2.0 Info

OAuth v1.0a

Verify API Key

Access Control

Generate SAML Assertion

Validate SAML Assertion

Mediation

JSON to XML

XML to JSON

Raise Fault

XSL Transform

SOAP Message Validation

Assign Message

Extract Variables

Access Entity

Key Value Map Operations

Extension

Java Callout

Python

JavaScript

Service Callout

Statistics Collector

Message logging

Google Cloud

Tools









PyUnit