Integration Testing At Uber

Luke Geiger, Software Engineer, Uber

MAY 17, 2016

UBER

Unit Testing At Uber

3 Types

UBER

Standard Unit Tests

- XCUnit Tests

View Based Testing

- FBSnapshotTests

Integration Tests

- OCTOPUS 🐙

Why do Integration Tests Matter?

The Build Cut

- Mobile Build Cut (iOS & Android) every Wednesday at
   12:00 pst
- Program teams do sanity tests once a week
  - ~2000 manual tests and counting!
  - Easily misses basic flows such as sign ups and
     payments
- Easily neglected.

Integration Testing

A Uber Challenge

- Uber has a interesting testing challenge when it comes to automated testing. Can anyone guess what it is?
- Not only Uber, but any game, or messaging app have this same problem.

Our Uber Useful Solution

Octopus

- A tool created internally that allows us to test both the rider and driver app in a true end to end test flow.
- In other words, we can automate an entire trip flow in a real life situation using both the rider app and the driver app at the same time.

Demo 1 UberRUSH

Demo 2 UberPOOL

Demo 3 ALL THE DEVICES

Deep Dive

How we make the tests

- On iOS we are writing tests using Apples UIAutomation
  framework (The tests are written in Javascript)
- Inside of this test is where we can use our Octopus framework.
- If we are testing a end to end test using both apps, a test
  directory will contain two files, a testDriver.js and a testRider.js.
- They communicate using what we call signals.
- Test directories are configurable!
  - Specific feature flags enabled/disabled
  - Custom Locations
  - Custom Accounts, etc.

Deep Dive

Terminology

- Test Host: Where octopus runs, Mac Mini, or a laptop
  you work on.
- Test Target: Where the actual test code runs,
  simulators, emulators or real devices.
- Channel: Between test targets.
- Signal: In communication channels we pass around
  signals.

Deep Dive

Signaling Api

- readSignal(channel, timeout)
  - blocking until a signal or timeout

- writeSignal(channel, content)
  - nonblocking

Deep Dive

Running tests

- An Octopus test runs on CI
  - Record/Replay Mode
  - Record Mode: Records network responses
  - Replay Mode: Plays back the network response from the cache
- To run these tests locally, we can use a command line script, and we also have a internal Mac App.

Questions

geiger@uber.com

Links

Blog

https://eng.uber.com/rescued-by-octopus/

Thank you

Luke Geiger
geiger@uber.com

UBER

Signaling - How it works?