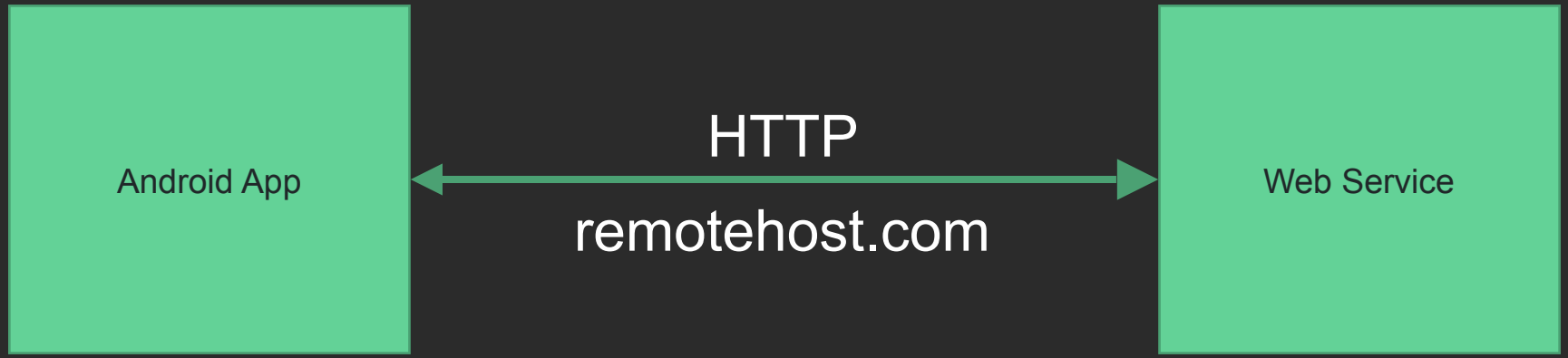
ADVANCED HTTP MOCKING WITH WIREMOCK

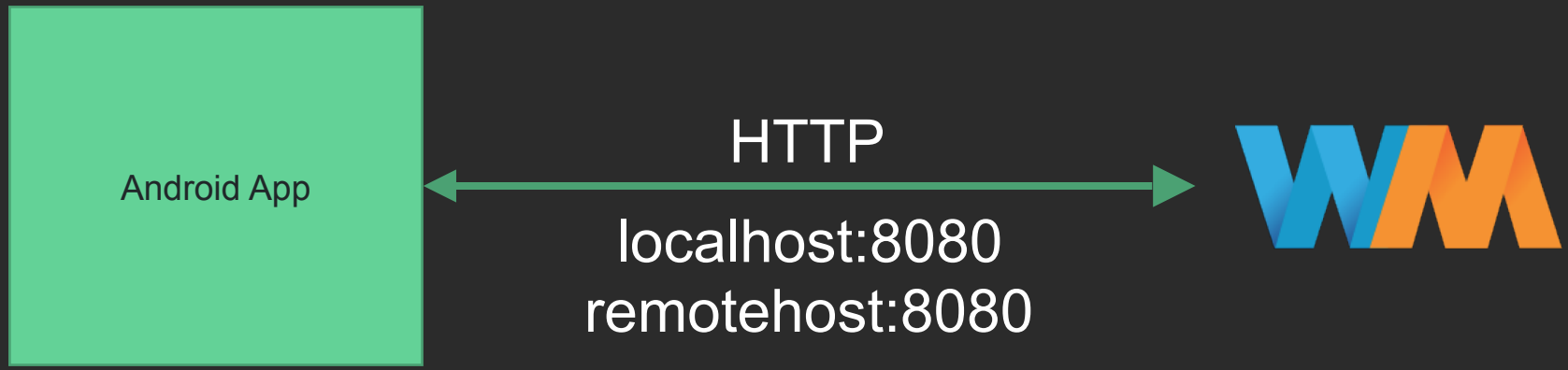
Sam Edwards - @HandstandSam

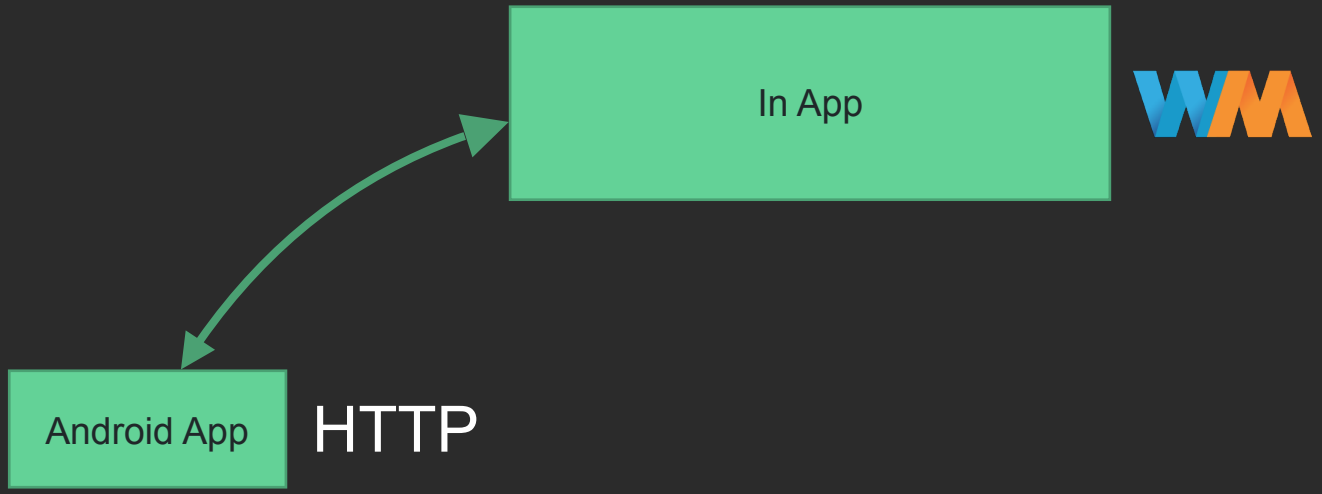
WireMock runs as
an HTTP Server.

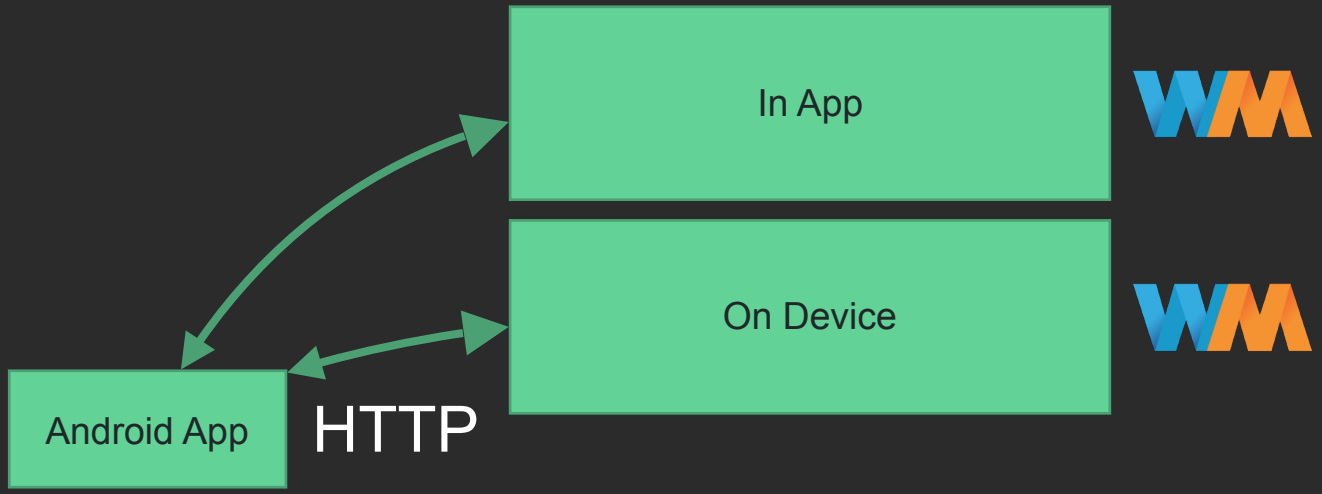
Normally

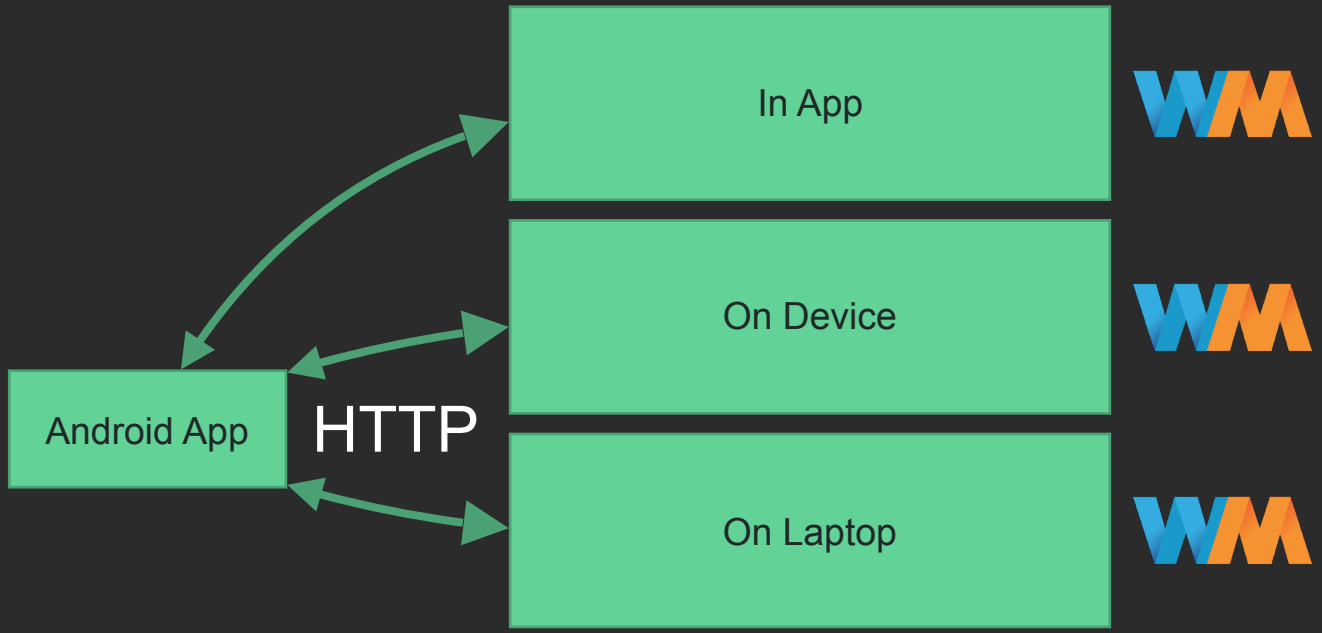


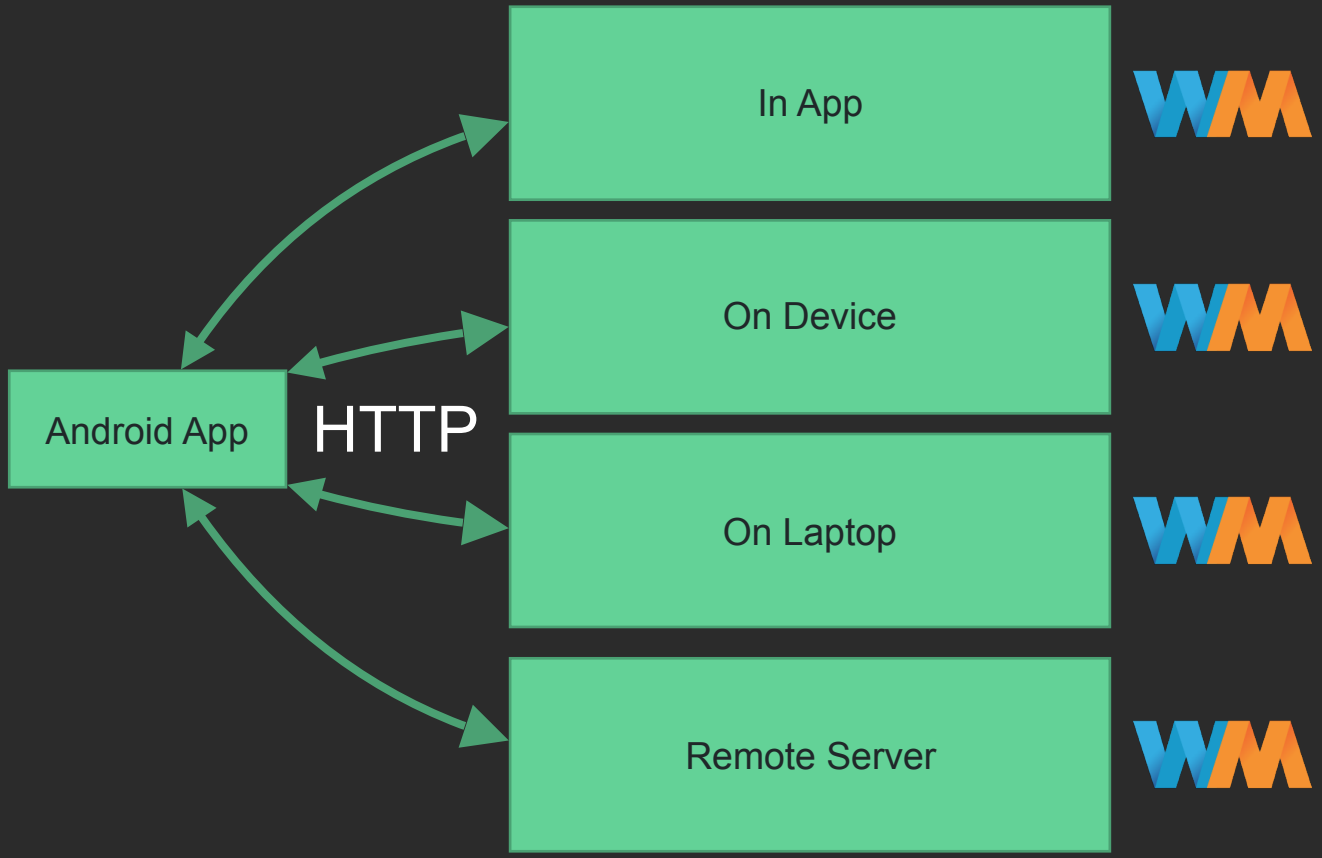
With WireMock

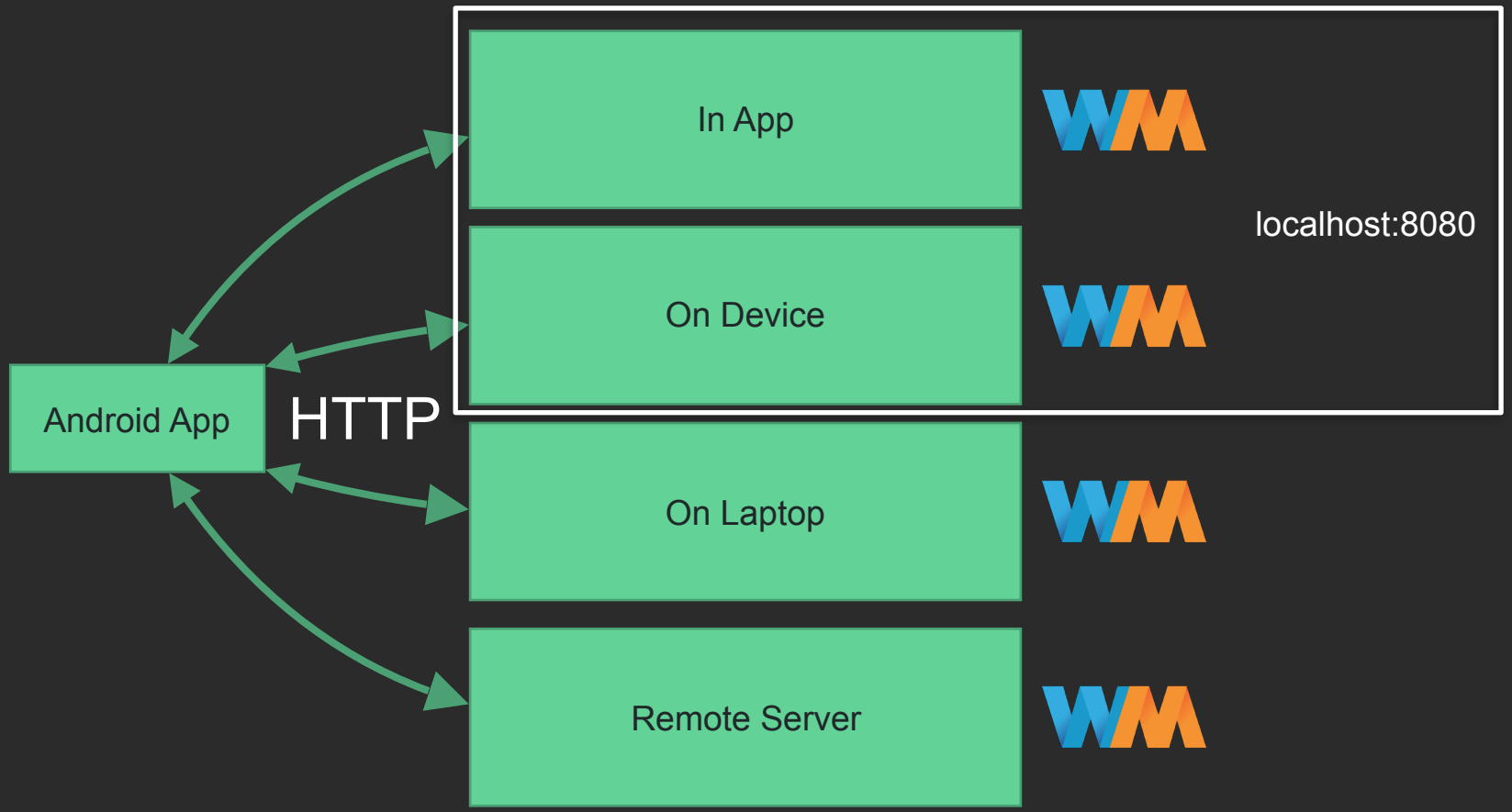


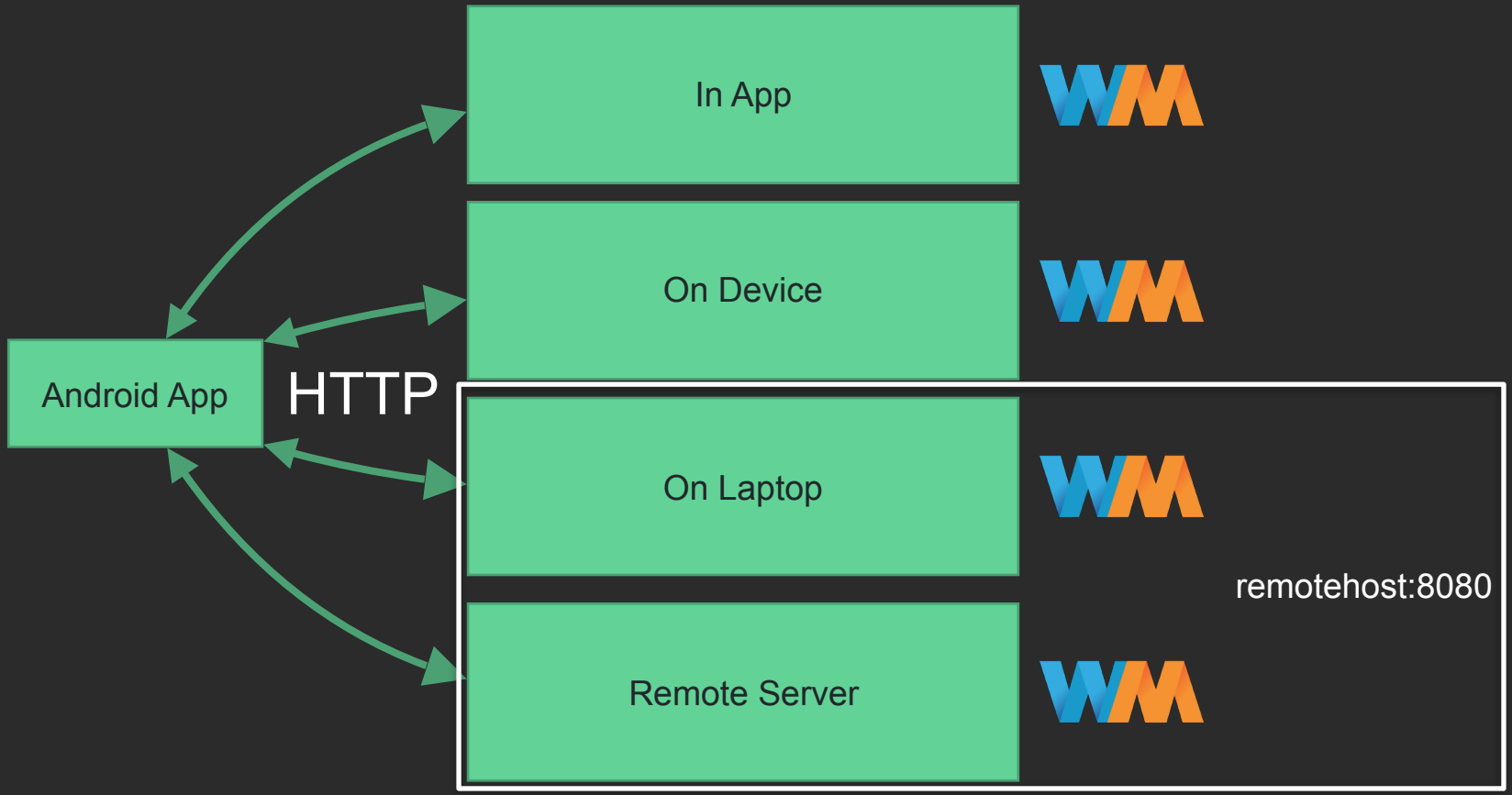

In App
Android App
HTTP


In App
On Device
Android App
HTTP

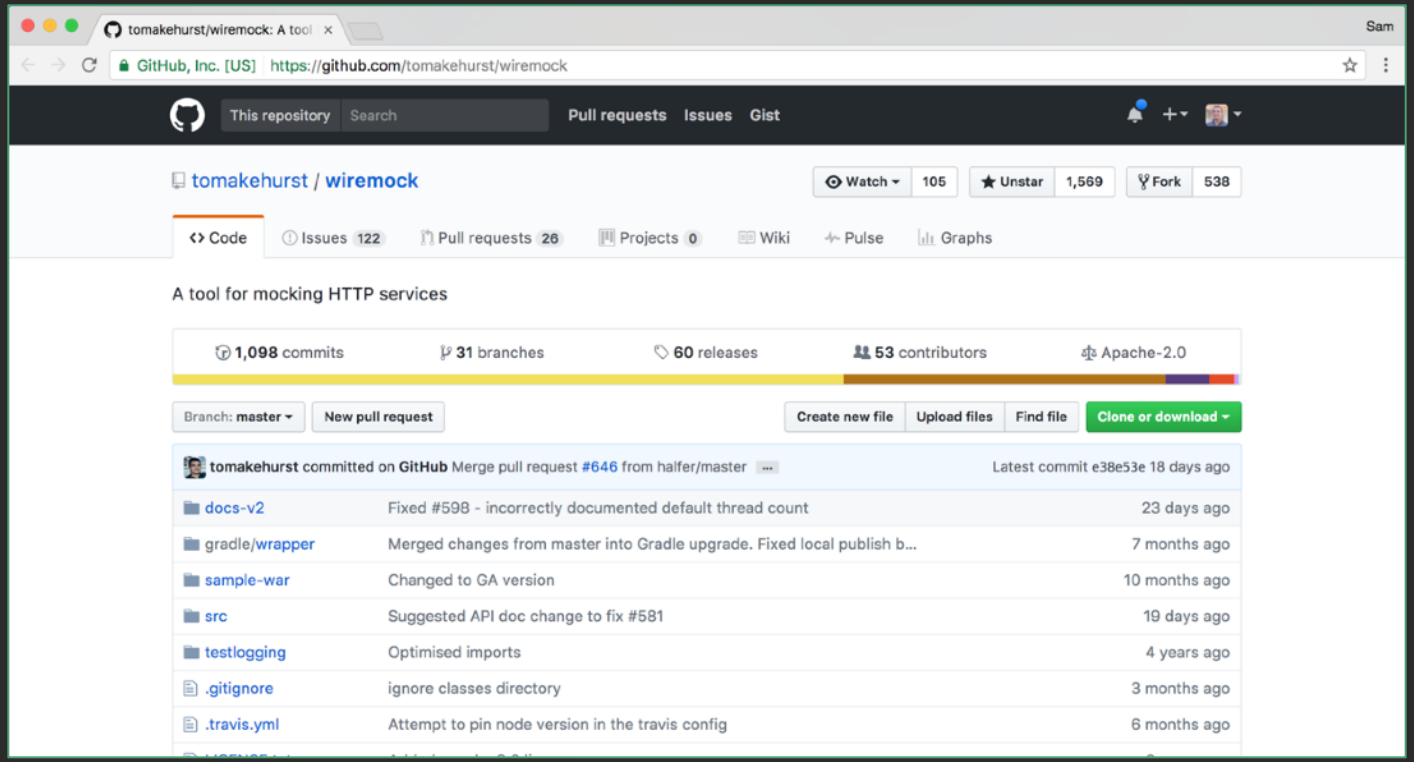

In App
On Device
On Laptop
Android App
HTTP


In App
On Device
On Laptop
Remote Server
Android App
HTTP


In App
On Device
localhost:8080
Android App
HTTP
On Laptop
Remote Server

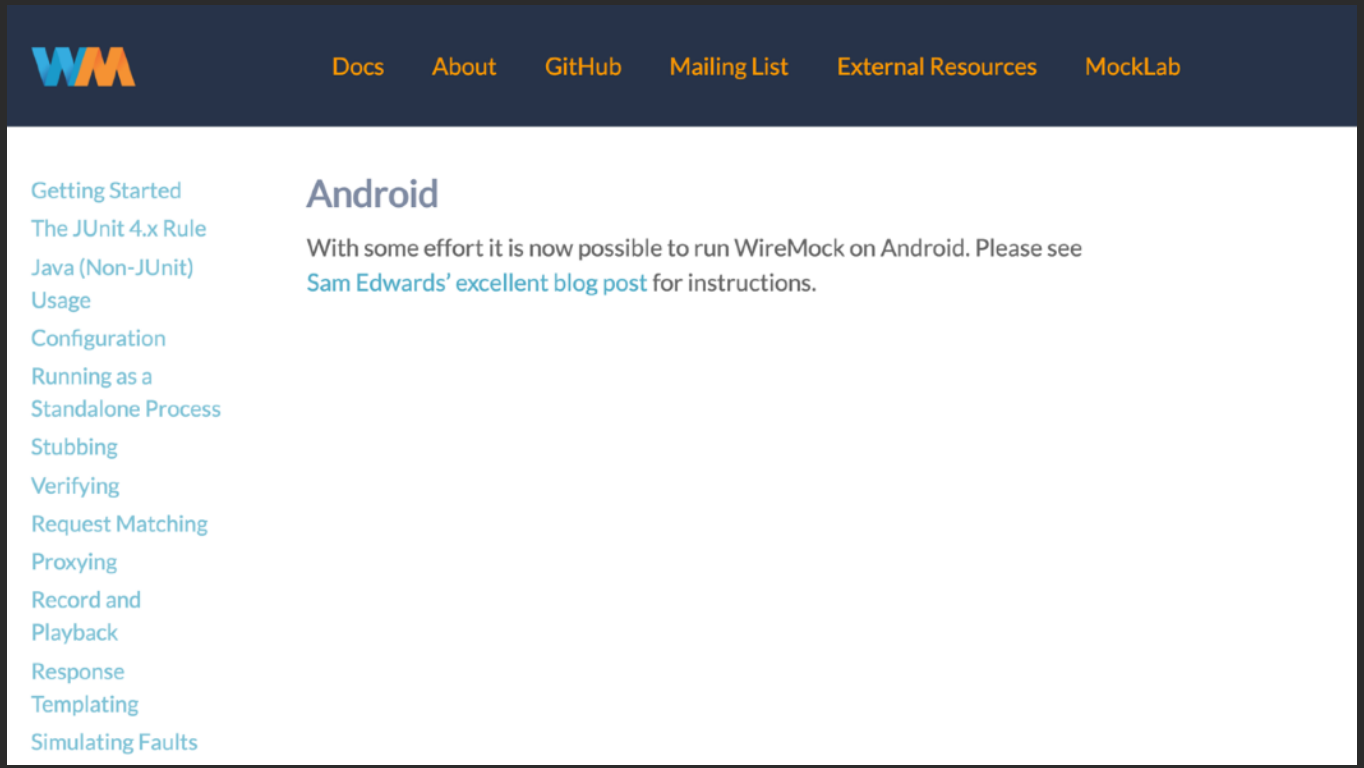

In App
On Device
Android App
HTTP
On Laptop
remotehost:8080
Remote Server

Started in 2011 by Tom Akehurst, Open Source, Apache 2.0

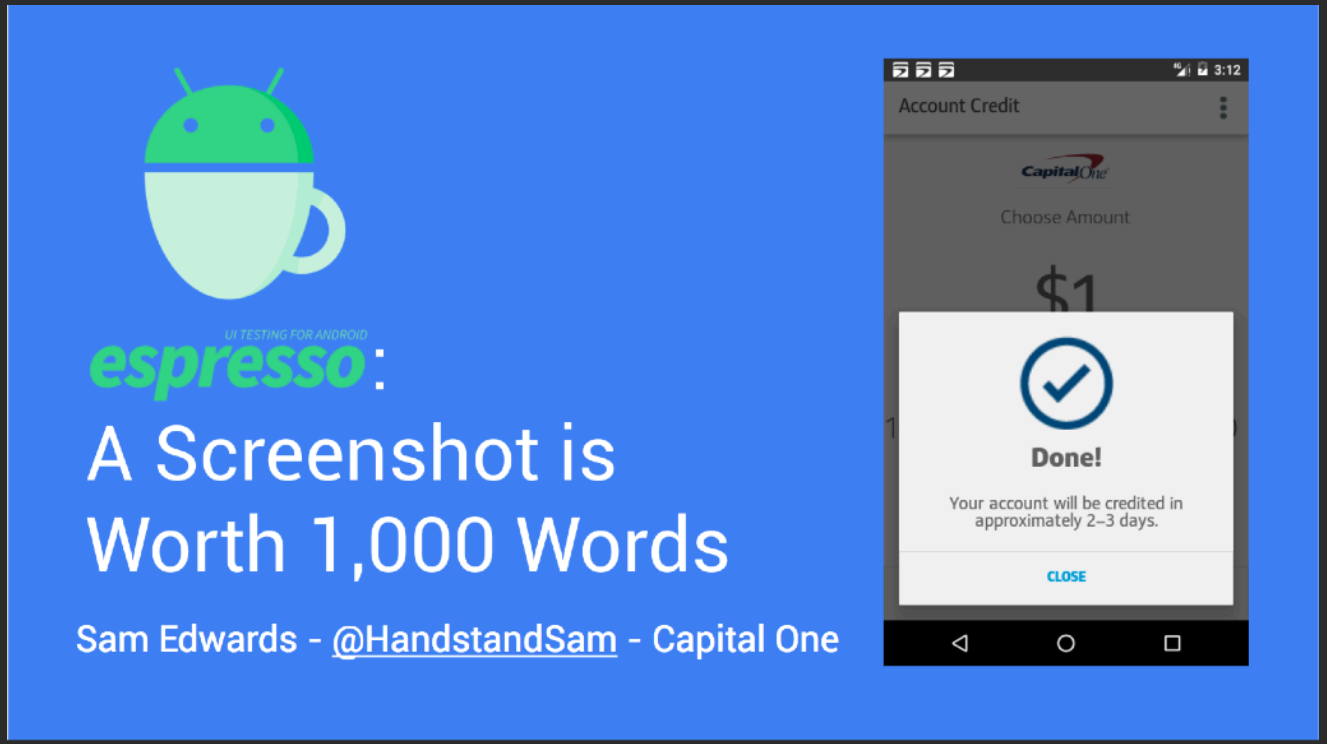


Presented at GTAC in November 2014 by Michael Bailey

Official Android Compatibility in January 2016



Droidcon NYC 2016

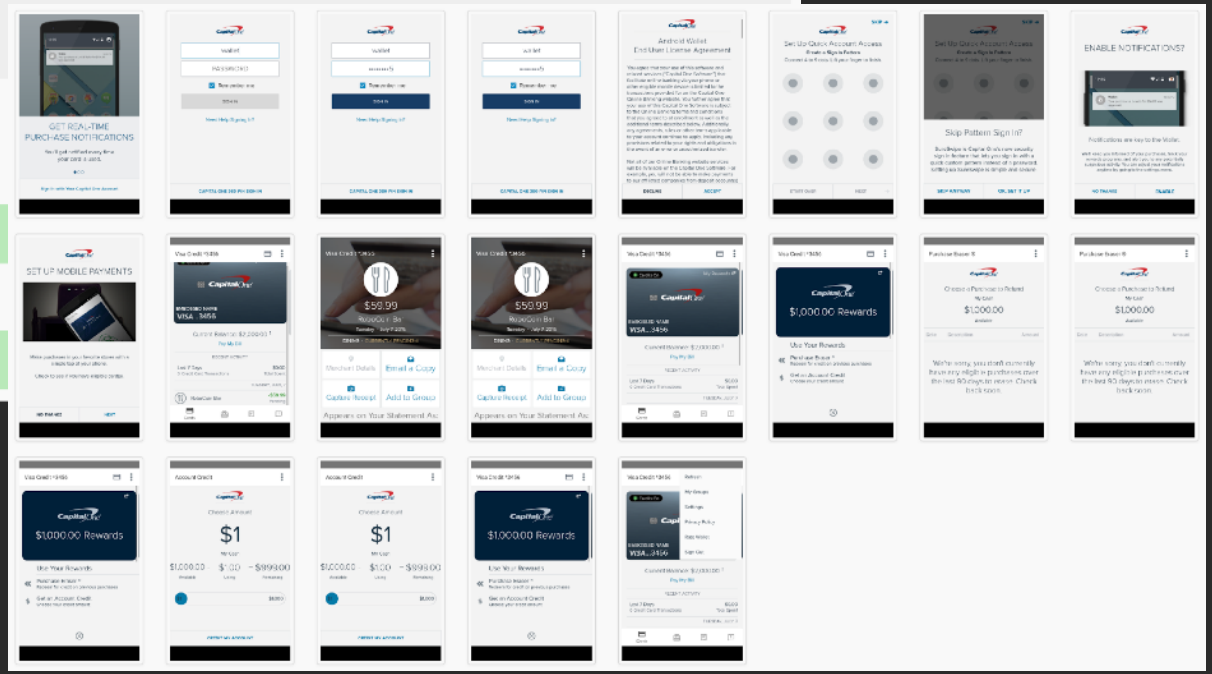


Spoon Execution

140 tests run across 2 devices with 140 passing and 0 failing in 20 minutes, 44 seconds at 2016-09-20 08:01 PM

384x640_mdpi_api_23

Nexus_6P_API_23



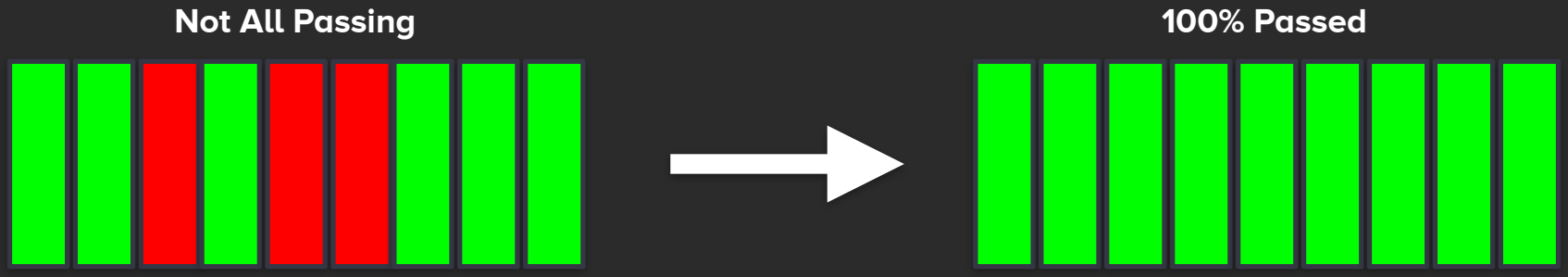
Continuous Integration & Cloud-Based Testing

buddybuild

circleci

etc...

Deterministic HTTP == More Reliable Tests



Why Mock HTTP Calls?

- Determinism & Reliability
- Avoid Rate Limits and Server Costs
- Avoid Modifying State
- Edge Cases & Error Scenarios
- Speed

Why Mock the HTTP Server?

- Exercises your full networking stack.
- Knowledge of the app's inner-workings are not needed.
- Swap your HTTP implementation without issues.
- Use recorded HTTP responses to create representative test cases.
- Test that your code survives in situations like 500 errors or slow-loading responses.

When should I use HTTP Mocking?

- Android Tests!
- Mock Flavor of App
- Not in your normal Android builds

```
public class ErrorScenariosTest {

    @Rule
    public ActivityTestRule<HomeActivity> mActivityTestRule
            = new ActivityTestRule<>(HomeActivity.class);

    @Test
    public void http500ErrorTest() {
        stubFor(get(urlEqualTo("/category/Nintendo/items"))
                .willReturn(aResponse().withStatus(500)));

        onView(allOf(withId(R.id.categories), isDisplayed()))
                .perform(actionOnItemAtPosition(0, click()));

        onView(withText("Networking Error")).check(matches(isDisplayed()));
    }
}
```

```
public class ErrorScenariosTest {

    @Rule
    public ActivityTestRule<HomeActivity> mActivityTestRule
            = new ActivityTestRule<>(HomeActivity.class);

    @Test
    public void http500ErrorTest() {
        stubFor(get(urlEqualTo("/category/Nintendo/items"))
                .willReturn(aResponse().withStatus(500)));

        onView(allOf(withId(R.id.categories), isDisplayed()))
                .perform(actionOnItemAtPosition(0, click()));

        onView(withText("Networking Error")).check(matches(isDisplayed()));
    }
}
```

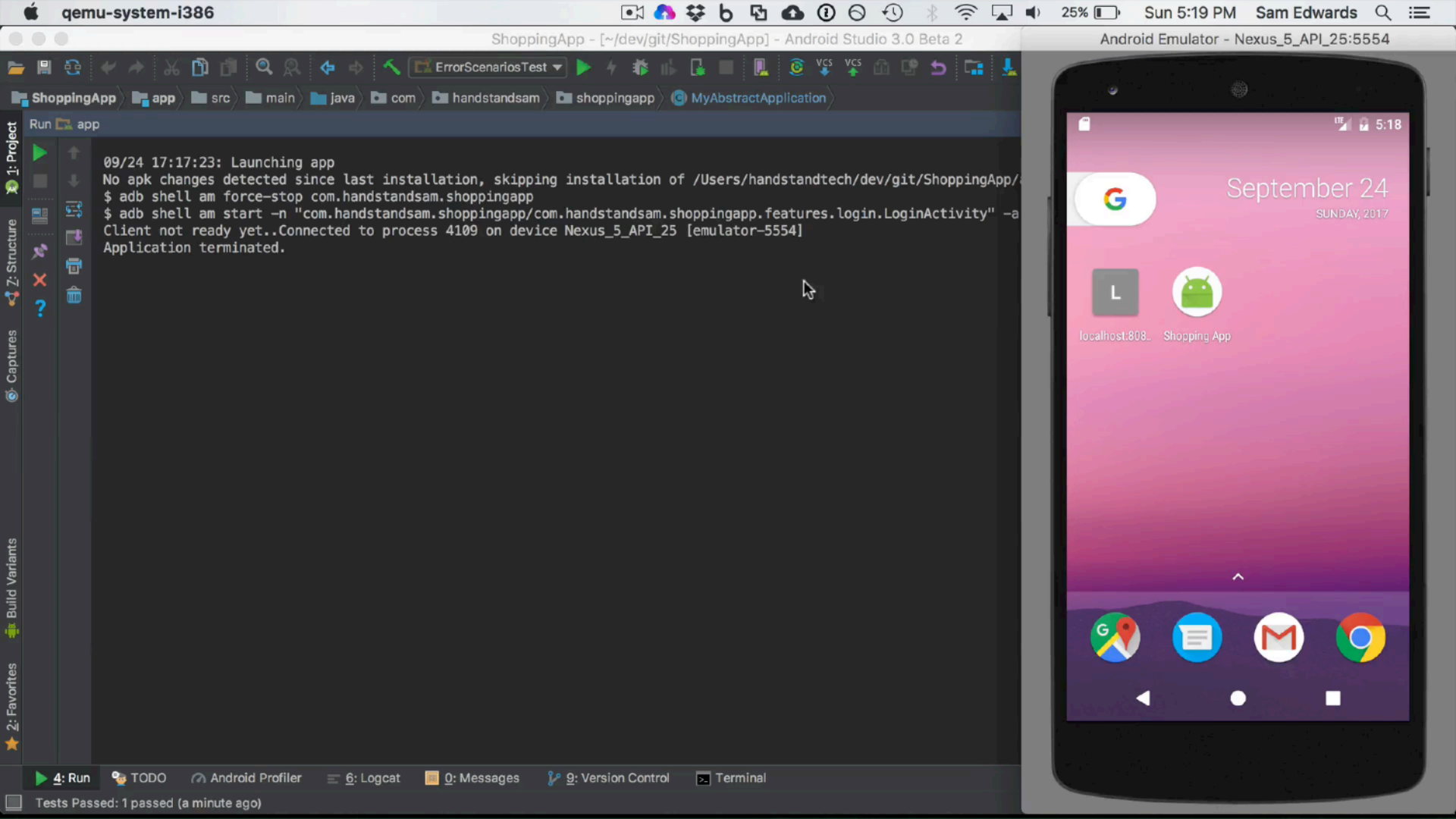
```
public class ErrorScenariosTest {

    @Rule
    public ActivityTestRule<HomeActivity> mActivityTestRule
            = new ActivityTestRule<>(HomeActivity.class);

    @Test
    public void http500ErrorTest() {
        stubFor(get(urlEqualTo("/category/Nintendo/items"))
                .willReturn(aResponse().withStatus(500)));

        onView(allOf(withId(R.id.categories), isDisplayed()))
                .perform(actionOnItemAtPosition(0, click()));

        onView(withText("Networking Error")).check(matches(isDisplayed()));
    }
}
```

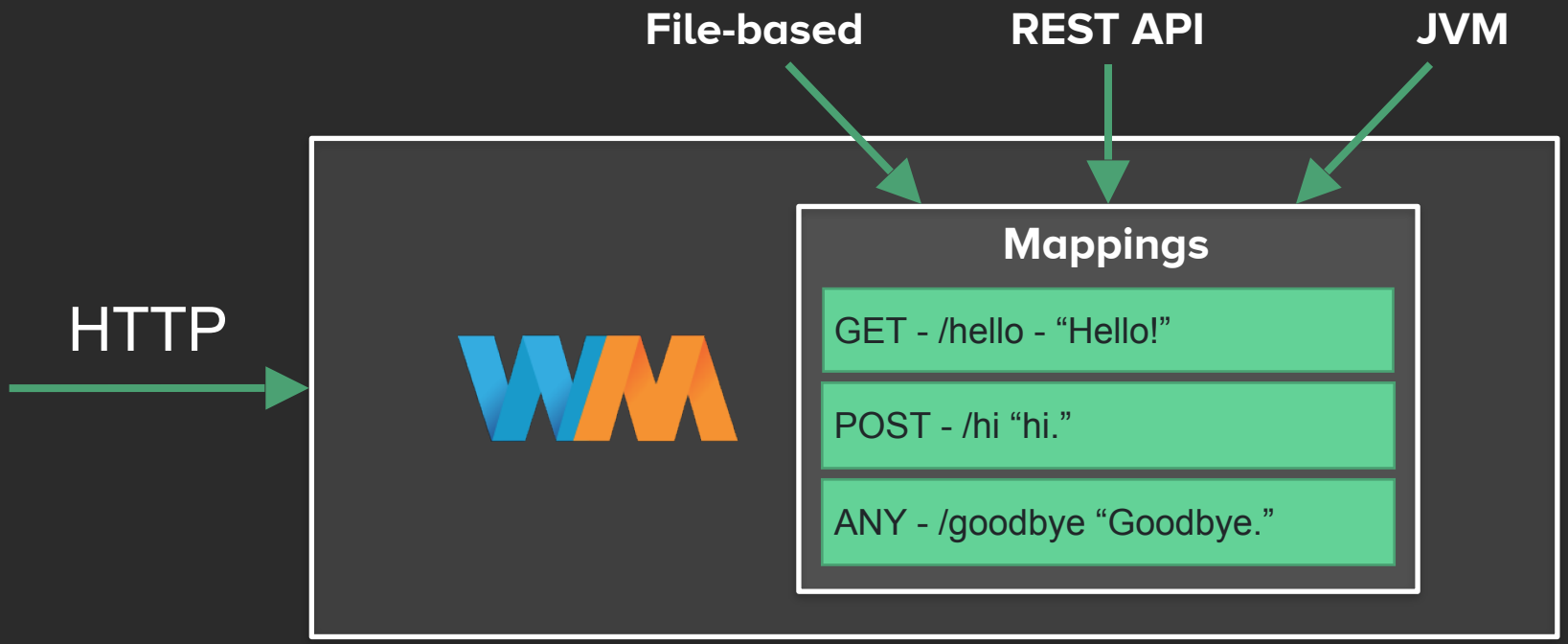

qemu-system-i386
25% Sun 5:19 PM Sam Edwards
ShoppingApp - [~/dev/git/ShoppingApp] - Android Studio 3.0 Beta 2
Android Emulator - Nexus_5_API_25:5554
ErrorScenariosTest
ShoppingApp app src main java com handstandsam shoppingapp MyAbstractApplication
Run app
09/24 17:17:23: Launching app
No apk changes detected since last installation, skipping installation of /Users/handstandtech/dev/git/ShoppingApp/
$ adb shell am force-stop com.handstandsam.shoppingapp
$ adb shell am start -n "com.handstandsam.shoppingapp/com.handstandsam.shoppingapp.features.login.LoginActivity" -a
Client not ready yet..Connected to process 4109 on device Nexus_5_API_25 [emulator-5554]
Application terminated.
1: Project
7: Structure
Captures
Build Variants
2: Favorites
4: Run
TODO
Android Profiler
6: Logcat
0: Messages
9: Version Control
Terminal
Tests Passed: 1 passed (a minute ago)
5:18
September 24
SUNDAY, 2017
localhost:808..
Shopping App

WHAT IS WIREMOCK?

Mapping

GET - **/hello** - should return “**Hello!**”

WireMock Mappings



WireMock Mapping

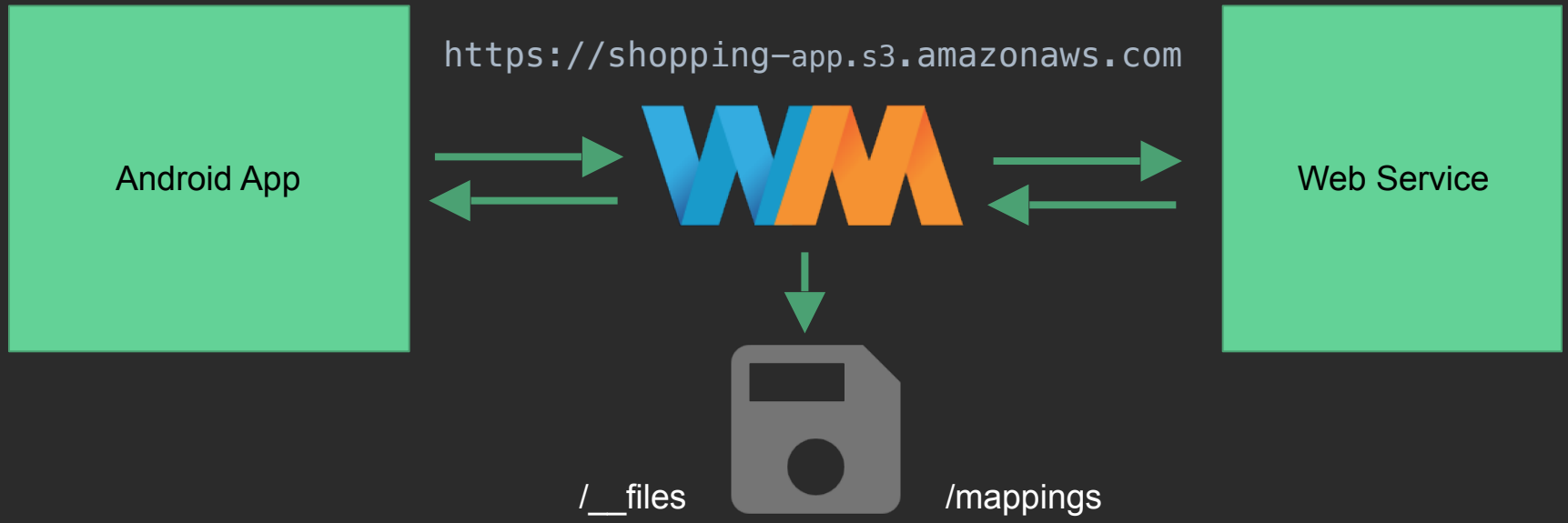
```
{
    "request": {
        "method": "GET",
        "urlPath": "/hello"
    },
    "response": {
        "status": 200,
        “body": “Hello!”,
        "headers": {
            "Content-Type": "text/plain"
        }
    }
}
```

Recording

Proxying


Android App
https://shopping-app.s3.amazonaws.com
Web Service

Record File-based Mappings to Disk while Proxying



Record File-based Mappings to Disk

Programmatic Java

```
WireMockServer wireMockServer = new WireMockServer(wireMockConfig().withRootDirectory(getRootDirectory()));
wireMockServer.enableRecordMappings(new SingleRootFileSource(getMappingDirectory()),
   new SingleRootFileSource(getFileDirectory()));
wireMockServer.stubFor(any(urlMatching(".*")).willReturn(aResponse()
  .proxiedFrom("https://shopping-app.s3.amazonaws.com")));
wireMockServer.start();
```

Command Line

```
java -jar wiremock-standalone-2.8.0.jar \
  --proxy-all https://shopping-app.s3.amazonaws.com \
  --record-mappings
```

Demo:
Android Recording

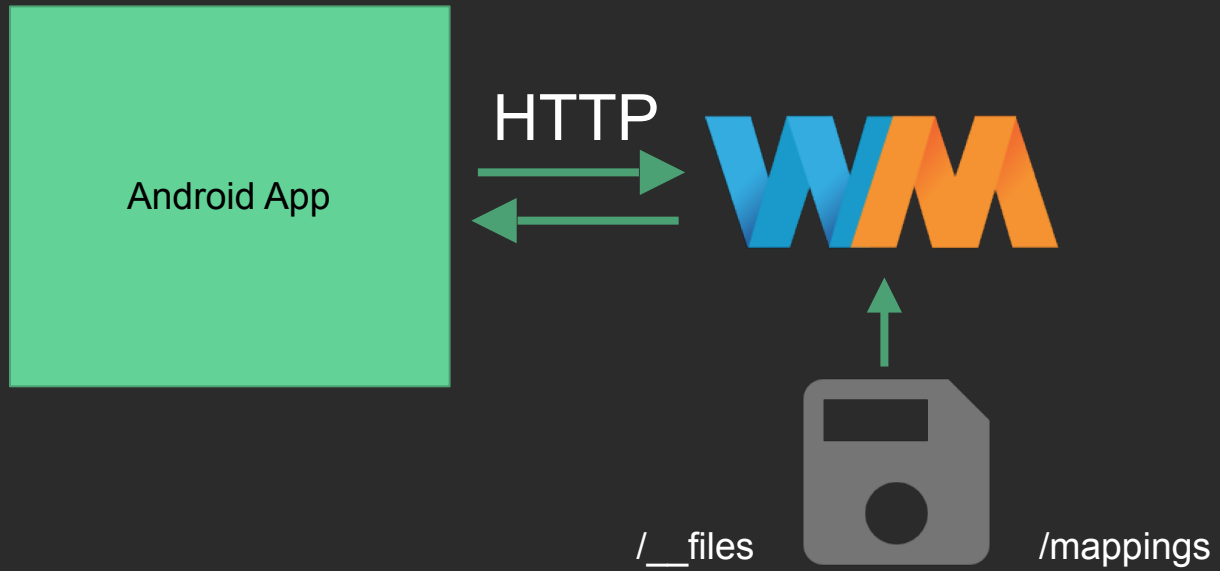
File-Based Recording & Playback on Android

- Caveats:
  - Must configure to use External Storage/SD Card for __files and mappings
- Recommendation (getting started):
  - Use Emulator & Run WireMock on Laptop
  - Connect Emulator to http://10.0.2.2:8080

Playback of File-Based Mappings on Android

@HandstandSam

Playback File-based Mappings



Playback File-based Mappings from Disk

Programmatic Java

```
WireMockServer wireMockServer = new WireMockServer(wireMockConfig().withRootDirectory(getRootDirectory()));
wireMockServer.start();
```

Command Line

```
java -jar wiremock-standalone-2.8.0.jar
```

Viewing Mappings

- http://localhost:8080/__admin/

Demo:
Android Playback

Stubbing

Terminology

“Stubbing” is the act of creating a “Mapping” programmatically

Stubbing Options

- File-based
- JVM
- REST API

Programmatic Stubbing via Java API

```
stubFor(get(urlEqualTo(“/hello"))
        .willReturn(aResponse()
                .withHeader("Content-Type", "text/plain")
                .withBody("Hello world!")));
```

Programmatic Stubbing via Java API

```
stubFor(get(urlEqualTo(“/hello"))
        .willReturn(aResponse()
                .withHeader("Content-Type", "text/plain")
                .withBody("Hello world!")));
```

Programmatic Stubbing via Java API

```
stubFor(get(urlEqualTo(“/hello"))
        .willReturn(aResponse()
                .withHeader("Content-Type", "text/plain")
                .withBody("Hello world!")));
```

Programmatic Stubbing via Java API

```
stubFor(get(urlEqualTo(“/hello"))
        .willReturn(aResponse()
                .withHeader("Content-Type", "text/plain")
                .withBody("Hello world!")));
```

Programmatic Stubbing via Java API

```
stubFor(get(urlEqualTo(“/hello"))
        .willReturn(aResponse()
                .withHeader("Content-Type", "text/plain")
                .withBody("Hello world!")));
```

WireMock Mapping

```
{
    "request": {
        "method": "GET",
        "urlPath": "/hello"
    },
    "response": {
        "status": 200,
        “body": “Hello!”,
        "headers": {
            "Content-Type": "text/plain"
        }
    }
}
```

Stubbing Options

- File-based
- JVM
- REST API

Running WireMock Standalone JAR

- Download Standalone JAR from Maven Central
- java -jar wiremock-standalone-2.8.0.jar
- --port
- --https-port
- http://localhost:8080/__admin/

Demo:
Language Agnostic Stubbing via REST API

Running WireMock as a Proxy

```
java -jar wiremock-standalone-2.8.0.jar \
  --proxy-all https://shopping-app.s3.amazonaws.com
```

Programmatic Stubbing via REST API

```
curl --verbose \
-X POST \
--data '{
    "request": {
        "method": "GET",
        "urlPath": "/hello"
    },
    "response": {
        "status": 200,
        "body": "Hello world!",
        "headers": {
            "Content-Type": "text/plain"
        }
    }
}' \
'http://localhost:8080/__admin/mappings'
```

Stubbing via REST API - String Body

```
curl --verbose \
-X POST \
--data '{
    "request": {
        "method": "GET",
        "url": "/category/Nintendo/items"
    },
    "response": {
        "status": 200,
        "body": "[{\"label\": \"?\"}]"
    }
}' \
'http://localhost:8080/__admin/mappings'
```

Stubbing via REST API - JSON Response

```
curl --verbose \
-X POST \
--data '{
  "request": {
      "method": "GET",
      "url": "/category/Nintendo/items"
  },
  "response": {
      "jsonBody": [{
        "image": "https://shopping-app.s3.amazonaws.com/droidcon-nyc/droidcon-nyc.png",
        "label": "Droidcon NYC"
      }]
  }
}' \
'http://localhost:8080/__admin/mappings'
```

Stubbing via REST API - Binary Response

```
curl --verbose \
-X POST \
--data '{
    "request": {
        "method": "GET",
        "url": "/category/Nintendo/items"
    },
    "response": {
        "base64Body": "W3sibGFiZWwiOiAiQmFzZTY0IEVuY29kZWQifV0="
    }
}' \
'http://localhost:8080/__admin/mappings'
```

Stubbing via REST API - 500 Error Response

```
curl --verbose \
-X POST \
--data '{
    "request": {
        "method": "GET",
        "url": "/category/Nintendo/items"
    },
    "response": {
        "status": 500
    }
}' \
'http://localhost:8080/__admin/mappings'
```

chrome web store

ssaammee@gmail.com

wiremock

« Home

Extensions
Themes
Apps

FEATURES

Runs Offline
By Google
Free
Available for
Works with

RATINGS

Wiremock extension

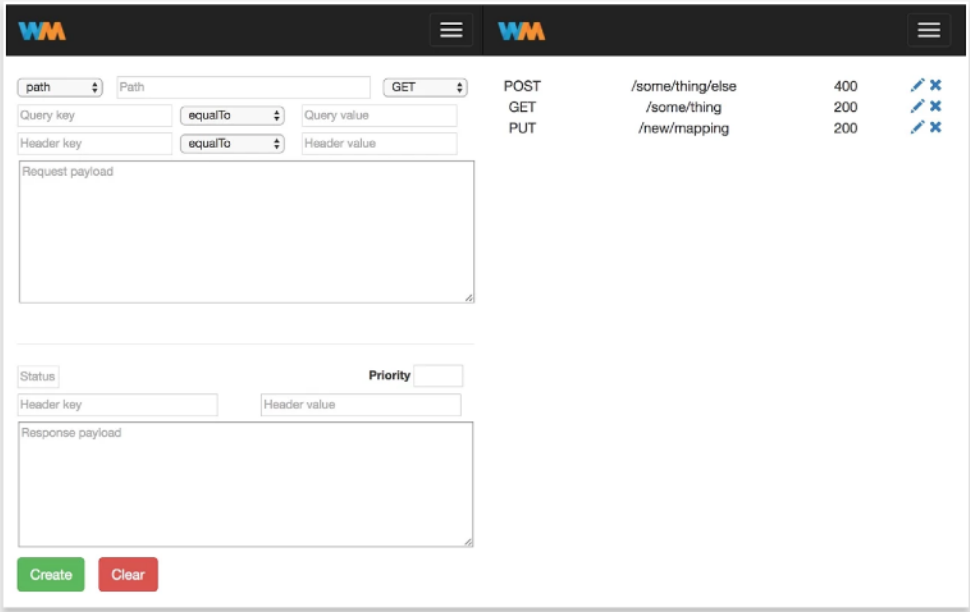
offered by www.mwtestconsultancy.co.uk

(0) | Developer Tools | 68 users

ADDED TO CHROME

OVERVIEW | REVIEWS | SUPPORT | RELATED

G+1 2



Compatible with your device

Chrome extension for managing stubs within your Wiremock instance.

Create, edit and delete stubs with ease with this extension for Wiremock. Features include:

* Configurable extension to work with any instances of Wiremock you might have running
* Full editor to create and edit stubs within Wiremock
* Mappings view to view all our your Wiremock stubs
* Ability to easily delete stubs

Report Abuse

Additional Information

Version: 1.0.0
Updated: 1 February 2017
Size: 418KiB
Language: English (UK)

https://chrome.google.com/webstore/detail/wiremock-extension/ikiapfdpbmofgmlhpjfnhdjelkleljbl

Complex Stubbing

URL Pattern Stub

```
curl --verbose \
-X POST \
--data '{
    "request": {
        "method": "GET",
        "urlPathPattern": "/category/.*/items$"
    },
    "response": {
        "jsonBody": [{
          "image": "https://shopping-app.s3.amazonaws.com/droidcon-nyc/kittens.jpg",
          "label": "Kittens"
        }]
    }
}' \
'http://localhost:8080/__admin/mappings'
```

Query Parameter Stub

```
curl --data '{
    "request": {
        "method": "GET",
        "urlPath": "/protected",
        "queryParameters":{
          "password": {
            "equalTo": "password"
          }
        }
    },
    "response": {
        "body": "<h1>How did you know that my password was 'password'?</h1>"
    }
}' \
'http://localhost:8080/__admin/mappings'
```

Other Complex Stubbing

- Header Matching (Serve Multiple API Versions)
- Request Body Matching
    - Equal To
    - Contains
    - JSON Body
        - Doesn’t care about order of JSON elements

Simulating Errors and Latency

Pre-Defined Fault Types

- EMPTY_RESPONSE
    - Return a completely empty response.
- MALFORMED_RESPONSE_CHUNK
    - Send an OK status header, then garbage, then close the connection.
- RANDOM_DATA_THEN_CLOSE
    - Send garbage then close the connection.

Simulating Faults

```
curl --verbose \
-X POST \
--data '{
    "request": {
        "method": "GET",
        "url": "/category/Nintendo/items"
    },
    "response": {
        "fault": "MALFORMED_RESPONSE_CHUNK"
    }
}' \
'http://localhost:8080/__admin/mappings'
```

Stub Specific Fixed Delay

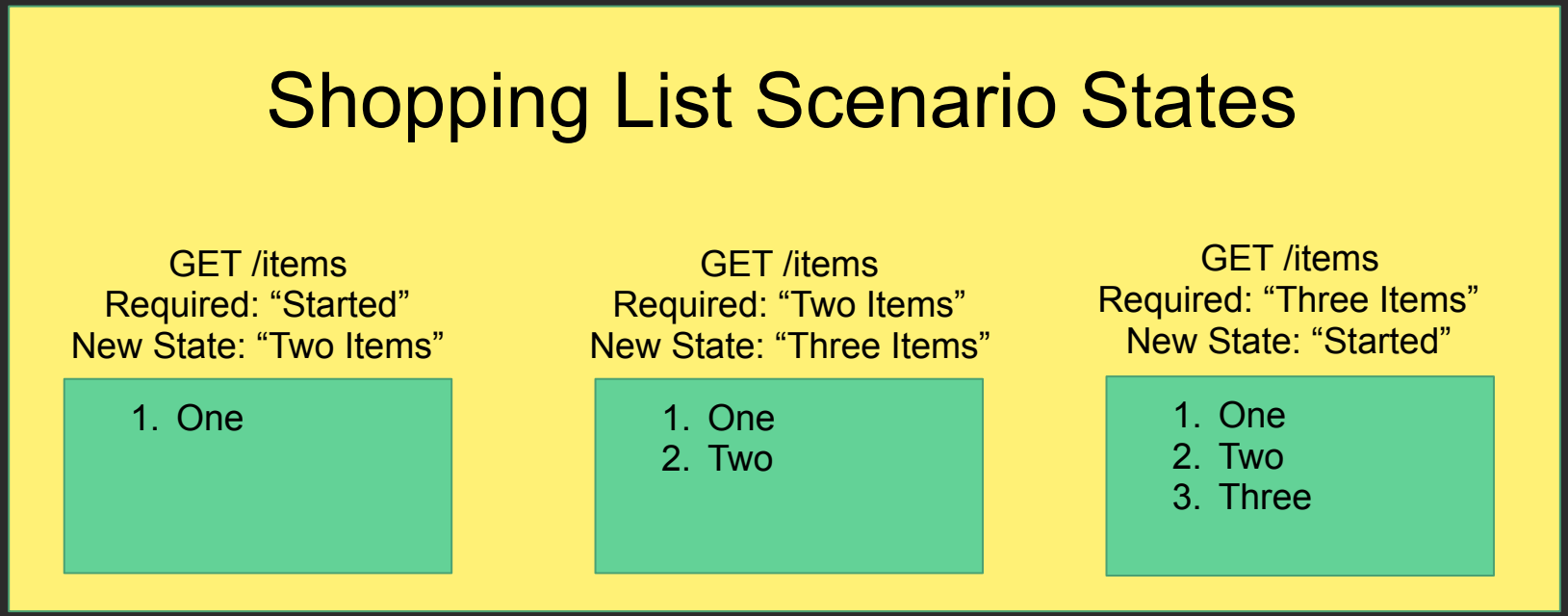
```
curl --verbose \
-X POST \
--data '{
    "request": {
        "method": "GET",
        "url": "/category/Nintendo/items"
    },
    "response": {
        "body": "[{\"label\": \"Better Late Than Never\"}]",
        "fixedDelayMilliseconds": "2000"
    }
}' \
'http://localhost:8080/__admin/mappings'
```

Delay Types

- Stub-specific Delays
- Global Delays
- Static Delays
- Random Delays

Stateful Behavior with Scenarios

Stateful Behavior with Scenarios



Verifying Networking Calls

Verifying Network Calls (Mockito-like Syntax)

- Verify Called Once

```
verify(getRequestedFor(urlEqualTo("/category/Nintendo/items")));
```

- Verify Called 3 Times

```
verify(3, getRequestedFor(urlEqualTo("/category/Nintendo/items")));
```

- Full Request Log

```
List<ServeEvent> allServeEvents =  wireMockServer.getAllServeEvents();
```

Programmatic Resets

- wireMockServer.resetMappings();
- wireMockServer.resetScenarios();

What Kind of Stubbing Should I Use?

File Based vs. Programmatic

Programmatic Mocks > File-based Mocks

- Save time by avoiding time consuming record/playback cycle.
- Generate time-sensitive data (yesterday, etc).
- Iterate through models to generate multiple ID specific mappings.
- Hold a programmatic model of expected results for assertions.
- Again, save a LOT of time after a small-ish upfront investment.

File-based Mocks > Programmatic Mocks

- Recording
- Getting Started
- Simple Cases
- Working with Non-Developers

Android Caveats

WireMock on Android - Dependencies

```
dependencies {
    androidTestCompile(“com.github.tomakehurst:wiremock:2.8.0") {
        exclude group: 'org.apache.httpcomponents', module: 'httpclient'
        exclude group: 'org.json', module: 'json'
        exclude group: 'org.ow2.asm', module: 'asm'
    }
    androidTestCompile 'org.apache.httpcomponents:httpclient-android:4.3.5.1'
}
```

METHODS COUNT

Your solution for a perfectly fit APK

MAIN PLUGINS ABOUT

com.github.tomakehurst:wiremock:2.8.0

BADGES SHARE

Library name	Total count
com.github.tomakehurst:wiremock:2.8.0	73660

Methods count	Dependencies	JAR Size (KB)	DEX size (KB)
3224	34	1493	366

Dependencies methods count	Dependencies JAR size (KB)	Dependencies DEX size (KB)
70436	10152	8879

DEPENDENCY LIST COMPUTED VERSIONS CHARTS

Sebastiano Gottardo | Nicola Miotto | Dario Marcato

DONATE

Compile Dependencies (18)

Category/License	Group / Artifact	Version	Updates
JSON Lib Apache 2.0	com.fasterxml.jackson.core » jackson-core	2.8.9	2.9.1
JSON Lib Apache 2.0	com.fasterxml.jackson.core » jackson-annotations	2.8.9	2.9.1
JSON Lib Apache 2.0	com.fasterxml.jackson.core » jackson-databind	2.8.9	2.9.1
Apache 2.0	com.flipkart.zjsonpatch » zjsonpatch	0.3.0	0.3.6
Template Engine Apache 2.0	com.github.jknack » handlebars	4.0.6	✔
Core Utils Apache 2.0	com.google.guava » guava	20.0	23.0
JSON Lib Apache 2.0	com.jayway.jsonpath » json-path	2.4.0	✔
Testing EPL 1.0	junit » junit	4.12	✔
CLI Parser MIT	net.sf.jopt-simple » jopt-simple	5.0.3	5.0.4
Core Utils Apache 2.0	org.apache.commons » commons-lang3	3.6	✔
HTTP Clients Apache 2.0	org.apache.httpcomponents » httpclient	4.5.3	✔
Web Server Apache 2.0 EPL 1.0	org.eclipse.jetty » jetty-server	9.2.22.v20170606	9.4.6.v20170531
Apache 2.0 EPL 1.0	org.eclipse.jetty » jetty-servlet	9.2.22.v20170606	9.4.6.v20170531
Apache 2.0 EPL 1.0	org.eclipse.jetty » jetty-servlets	9.2.22.v20170606	9.4.6.v20170531
Apache 2.0 EPL 1.0	org.eclipse.jetty » jetty-webapp	9.2.22.v20170606	9.4.6.v20170531
Logging MIT	org.slf4j » slf4j-api	1.7.12	1.7.25
Apache 2.0	org.xmlunit » xmlunit-core	2.3.0	2.5.0
Apache 2.0 BSD 3-clause	org.xmlunit » xmlunit-legacy	2.3.0	2.5.0

WireMock on Android - Project Config

Recommended, but not required:

```
android {
    defaultConfig {
        minSdkVersion 21
        multiDexEnabled true
    }
}
```

Final Thoughts

Recommendations

- Play with WireMock as a desktop/server tool immediately. It's awesome.
- Evaluate other libraries. Don't use WireMock in your app until you've looked at other options.
  - Mock Web Server - HTTP Server
  - Mock Web Server + RESTMock - More powerful
  - OkReplay - OkHttp Interceptor + File Based Recording
  - Retrofit - Behavior Delegate
  - Mockito
- 70,000 methods??? - Multidex and get over it

Don’t carry a Swiss-Army Knife if you only need a knife.

THANKS!

https://github.com/handstandsam/ShoppingApp