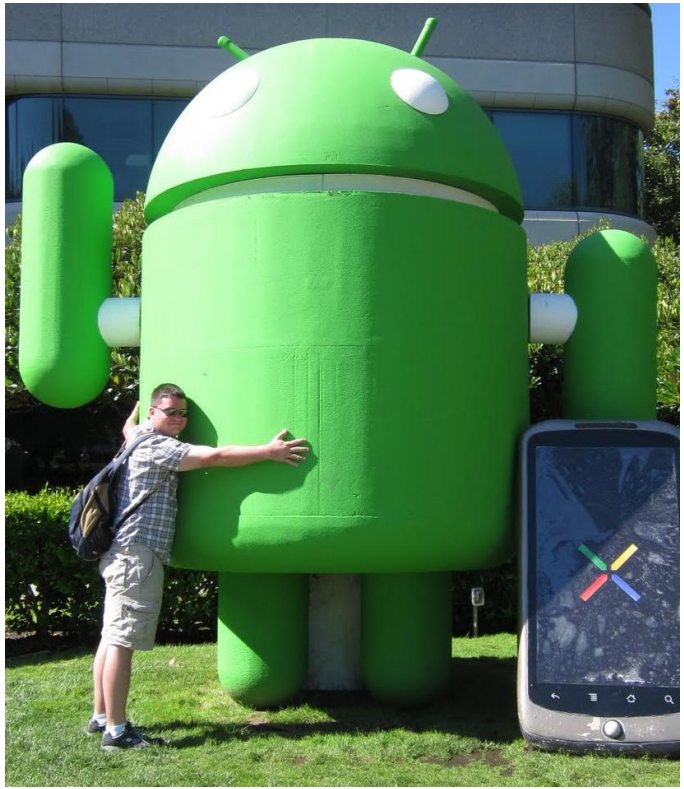
Android Developer Power Tools:
Charles Proxy

Patrick Hammond

ATOMIC ROBOT

About Me



- Over four awesome years developing Android apps!
- Atomic Robot cofounder
- Husband, Dad, Geek, Insomniac, and Foodie (yum!)

Outline

- What is Charles Proxy
- Android Integration
- Basic Usage
- Advanced Usage
- Discussion

Disclaimer

- **Warning:** prepare for jumping between the presentation, code, demos, etc. Take your Dramamine now.
- Sample code is aimed to drive discussion and show concepts only. It is far from polished so use with caution.

What’s the problem?

Have you ever found yourself saying these...

- “I wish I knew what my app and the server were saying to each other...”
- “I wish I could easily return different server responses to my app…”
- “I wish it was easier to make bad network stuff happen in my app…”

What is Charles Proxy?

Tell me more, please...

From the website

“Charles is an HTTP proxy / HTTP monitor / Reverse Proxy that enables a developer to view all of the HTTP and SSL / HTTPS traffic between their machine and the Internet. This includes requests, responses and the HTTP headers (which contain the cookies and caching information).” - charlesproxy.com

Product Highlights

- Windows, Linux, and Mac support
- Free 30 day trial
- $50 individual license
- Group licensing available

Will become essential to your development workflow. You will not regret this purchase!

Bonus!

Charles works great for ALL of these uses too:

- Android development
- iOS development
- Web development
- Lots of other development and testing where something talks to a server!

Android Setup

Let’s get started!

Emulator Startup

- Starting an emulator via the command line **with proxying turned on** is one extra flag

```
emulator -avd my_avd_name \
-http-proxy localhost:8888
```

Enabling SSL proxying

- “Man in the middle” proxying setup:
  1. Download and unzip: http://www.charlesproxy.com/ssl.zip
  2. `adb push charles*.crt /sdcard/charles.crt`
  3. Settings -> Security -> Install from SD card
- By design this enables a man in the middle attack. **You should still be careful!**

Charles Proxy Basics

Charles Proxy 101

Basics (demo)

- View logged requests/responses
  - Tree view: requests sorted by host and path
  - Sequence view: requests sorted by execution order
- Overview info, headers, raw output, and many more views!

Advanced Usage

Making the most of your $50

Advanced Usage

Demos:

- Reverse proxies
- Breakpoints
- Request/response editing
- Replaying/editing requests
- Introducing latency

Many more useful features available!

Reverse Proxy

Problem: You need to quickly and/or frequently repoint the app at a different location but are unable/unwilling to update the app configuration.

Example: During development and/or testing you are frequently switching between different environments.

Reverse Proxy

Reverse proxy setup:

- Charles -> Proxy -> Reverse Proxies…
- Check “Enable Reverse Proxies”
- Set the local port that will accept incoming requests
- Set the remote host/port to forward those requests to

Reverse Proxy

App reverse proxy setup:

- If using an emulator, point the app at 10.0.2.2:PORT where PORT is the local port you just setup.
- If using a physical device, use the IP of that machine running Charles in combination with the local port you just setup.

  This assumes that the physical device can reach the machine running Charles...which means ensuring firewalls, routers, and other network devices are all playing nice and you're able to setup this configuration.

Reverse Proxy

Note about SSL with a reverse proxy...

- As best I can tell, you will need to setup your app to trust the Charles SSL certificate or to trust all certificates.
- DO NOT SHIP YOUR APP WITH THIS TURNED ON!

Breakpoints

Problem: Between mobile networks and slow servers, apps don’t always get the info they need in a reasonable amount of time and this can cause you issues.

Example: The WiFi network your app is using freezes and your app makes requests but keeps getting timeouts. How do you test this?

Breakpoints

Breakpoint setup:

- Right click on a previously logged request and select “Breakpoints”.
- Can configure further by going to Charles -> Proxy -> Reverse Proxies
- Works best with a reverse proxy setup if the final destination routes to multiple IPs.

Breakpoints

Triggering breakpoints:

- Rerun something in your app that will call a path setup with a breakpoint.
- Notice that Charles will take you to a breakpoint screen.
- You can now cancel, abort, or execute the request.

Request/response editing

Problem: It isn’t easy to modify requests coming out of the app or responses coming back into the app.

Example: You don’t control the server your app talks to but you need to easily see how it behaves when it receives different responses.

Request/response editing

Request/response editing setup:

- Setup and trigger a breakpoint
- Open the “Edit Request” or “Edit Response” tab.
- Edit any part of the request or response (ex: change the status code from HTTP 200 to HTTP 500).

Introducing latency

Problem: You are testing on a set of devices and need to introduce consistent latencies for testing.

Example: You are trying to simulate a slow connection to make it easier to expose a lifecycle bug that only shows up on a physical device.

Introducing latency

Introducing latency setup:

- Setup a reverse proxy and point your app at the reverse proxy.
- Charles -> Proxy -> Throttling Settings…
- Turn on “Enable Throttling” and set the latency to a large value (ex: 30000 to setup a 30 second delay).

Introducing latency

Note: Similar network control features are available when using the Android emulator but not when using physical devices.

Resources

- www.charlesproxy.com

For your convenience...

Get the presentation and any code here:

https://github.com/patrickhammond/Presentation-AndroidPowerToolsCharlesProxy

madebyatomicrobot.com

Questions?

Email: **patrick@madebyatomicrobot.com**

Twitter: **@patrickhammond**

Google+: **+PatrickHammond**

GitHub: **patrickhammond**