ProxyURL

- Configuration

  NGORK to localhost to Heroku proxy server to OT servers

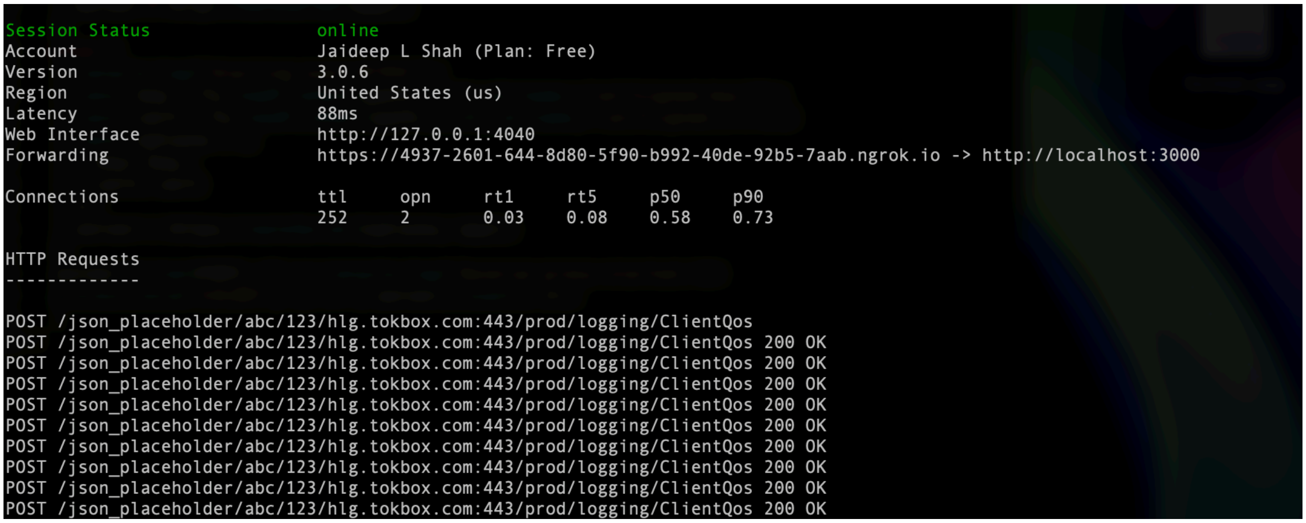
Tests

```
OTSessionSettings * set = [[OTSessionSettings alloc] init];
//set.proxyURL = @"https://4937-2601-644-8d80-5f90-b992-40de-92b5-7aab.ngrok.io";
//set.proxyURL = @"https://4937-2601-644-8d80-5f90-b992-40de-92b5-7aab.ngrok.io/";
//set.proxyURL = @"https://4937-2601-644-8d80-5f90-b992-40de-92b5-7aab.ngrok.io/json_placeholder";
//set.proxyURL = @"https://4937-2601-644-8d80-5f90-b992-40de-92b5-7aab.ngrok.io/json_placeholder/";
//set.proxyURL = @"https://4937-2601-644-8d80-5f90-b992-40de-92b5-7aab.ngrok.io/json_placeholder/abc/123";
set.proxyURL = @"https://4937-2601-644-8d80-5f90-b992-40de-92b5-7aab.ngrok.io/json_placeholder/abc/123/";

_session = [[OTSession alloc] initWithApiKey:kApiKey
                                   sessionId:kSessionId
                                    delegate:self
                                    settings:set];
```

Tools

- NGORK only NGORK to redirect to local host. Other direct approaches messes up in certs in iPhone simulator

  ngrok http http://localhost:3000



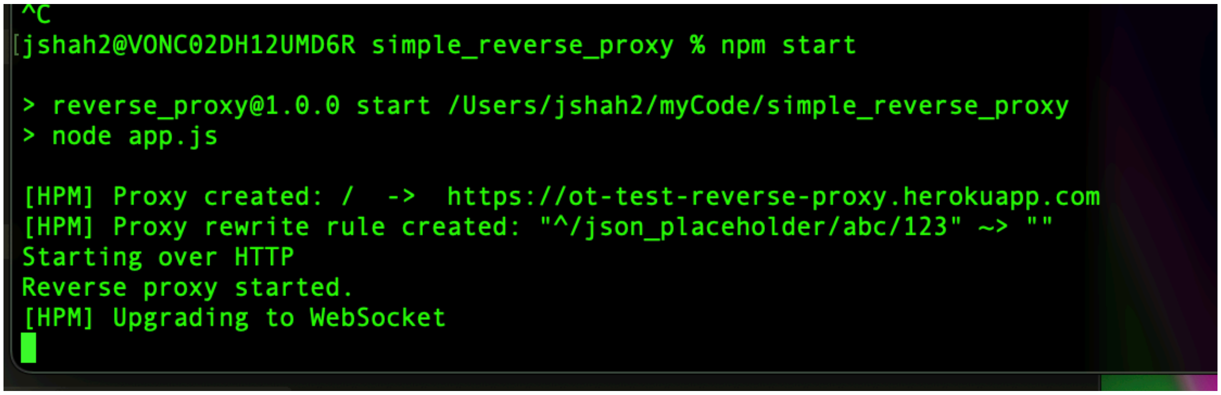
```
Session Status                online
Account                       Jaideep L Shah (Plan: Free)
Version                       3.0.6
Region                        United States (us)
Latency                       88ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://4937-2601-644-8d80-5f90-b992-40de-92b5-7aab.ngrok.io -> http://localhost:3000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              252     2       0.03    0.08    0.58    0.73

HTTP Requests
-------------

POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos
POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos 200 OK
POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos 200 OK
POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos 200 OK
POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos 200 OK
POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos 200 OK
POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos 200 OK
POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos 200 OK
POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos 200 OK
POST /json_placeholder/abc/123/hlg.tokbox.com:443/prod/logging/ClientQos 200 OK
```

Use roberto proxy sample with changes to redirect again to heroku. **CODE in NOTES**

https://github.com/robjperez/simple_reverse_proxy

```
const express = require('express');
const proxy = require('http-proxy-middleware');
const cors = require('cors');
const fs = require('fs');
const path = require('path');
const http = require('http');
const https = require('https');




const otProxy = proxy({
  target: 'https://ot-test-reverse-proxy.herokuapp.com',
  pathRewrite: {
        [`^/json_placeholder/abc/123`]: '',
    },
  changeOrigin: true,


  ws: true,
});

const app = express()
const PORT = 3000;
const USE_SSL = 0


app.use('/json_placeholder',  otProxy);

let server;

if (!USE_SSL) {
  console.log("Starting over HTTP");
  server = http.createServer(app);
}
else {
  console.log("Starting over HTTPS");
  const key = fs.readFileSync(path.join(__dirname, 'selfsigned.key'));
  const cert = fs.readFileSync(path.join(__dirname, 'selfsigned.crt'));
  const options = {
    key: key,
    cert: cert
  };
  server = https.createServer(options, app);
}
server.listen(PORT, () => console.log(`Reverse proxy started.`))
process.once('SIGINT', () => { server.close(); });
```

- ws needed



```
^C
[jshah2@VONC02DH12UMD6R simple_reverse_proxy % npm start

> reverse_proxy@1.0.0 start /Users/jshah2/myCode/simple_reverse_proxy
> node app.js

[HPM] Proxy created: /  ->  https://ot-test-reverse-proxy.herokuapp.com
[HPM] Proxy rewrite rule created: "^/json_placeholder/abc/123" ~> ""
Starting over HTTP
Reverse proxy started.
[HPM] Upgrading to WebSocket
```