



Consuming HTTP at Scale

Node Summit, Jan 24, 2012

Hello, I'm

Subbu Allamaraju

@sallamar

<http://www.subbu.org>

ql.io Brought to you by

ebay eBay's platform engineering

github
SOCIAL CODING <https://github.com/ql-io/ql.io>
Open source (Apache 2)

API Billionaires Club, 2011 edition



13 billion API calls / day *(May 2011)*



10 billion API calls / month *(January 2011)*



Over 260 billion objects stored in S3 *(January 2011)*



1.6 billion API-delivered stories / month *(October 2010)*



5 billion API calls / day *(April 2010)*



5 billion API calls / day *(October 2009)*



8 billion API calls / month *(Q3 2009)*



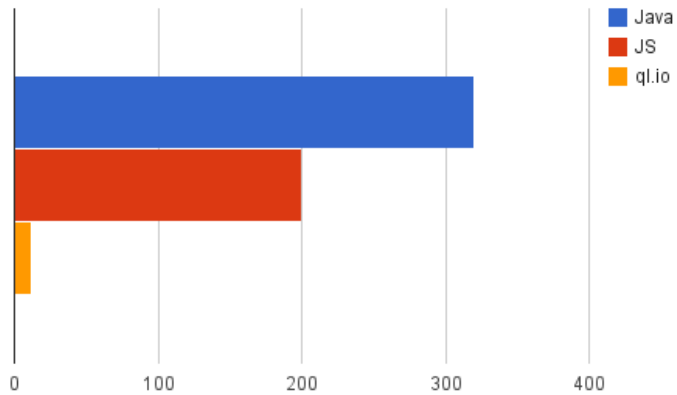
3 billion API calls / month *(March 2009)*

[<http://blog.programmableweb.com/2011/05/25/api-business-models-then-and-now/>]

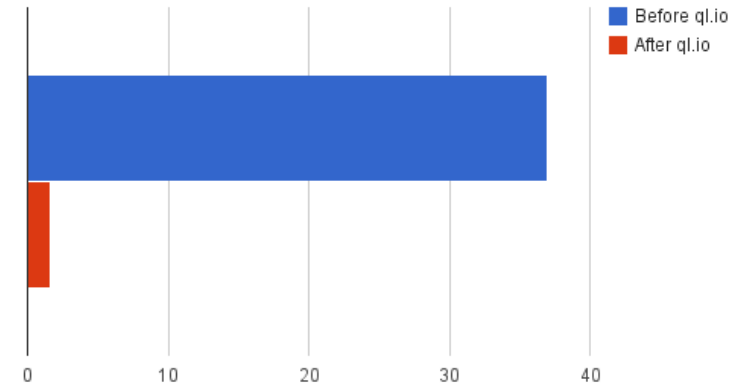
A **DSL** for HTTP
An HTTP **gateway**
Built on **node.js**

Make HTTP APIs **easy**
and **fast** to consume

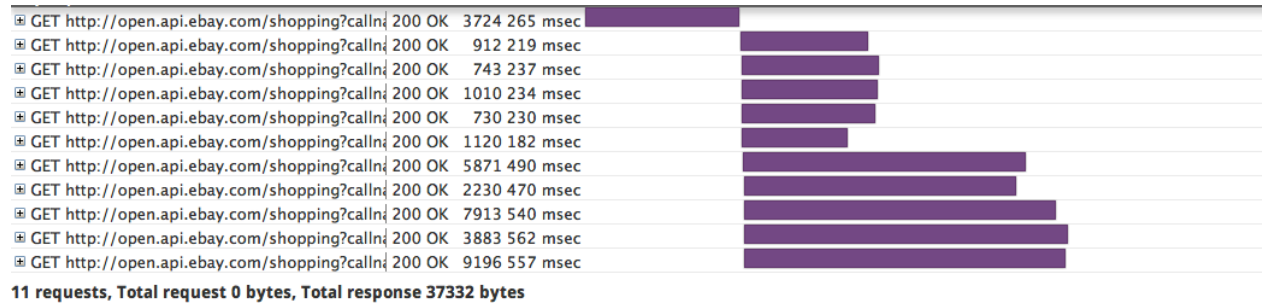
Lines of code for API calls



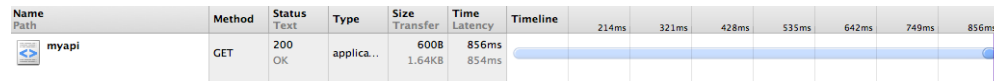
Data size (k)



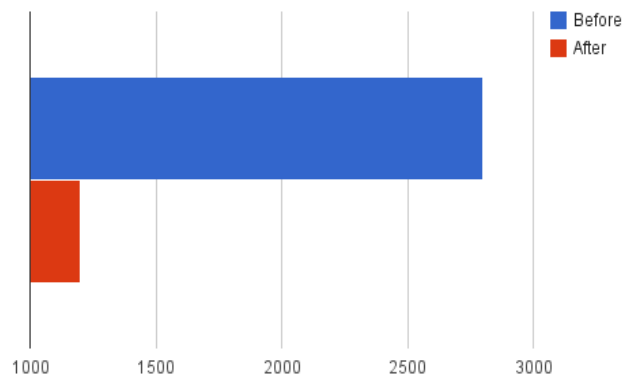
before



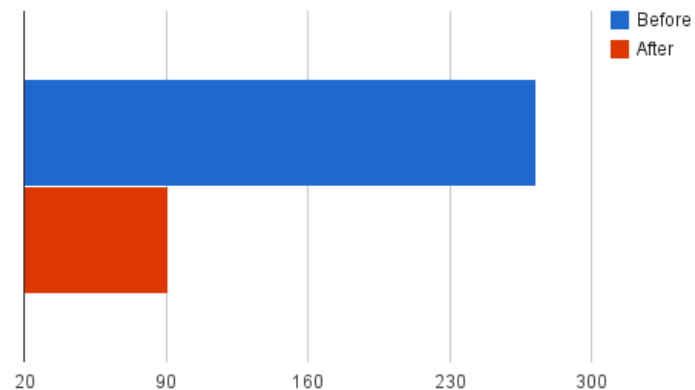
after



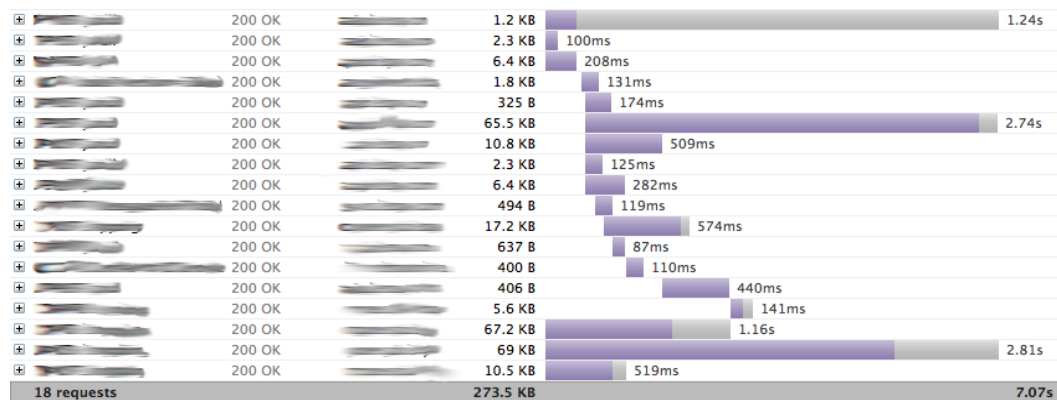
Lines of code for API calls



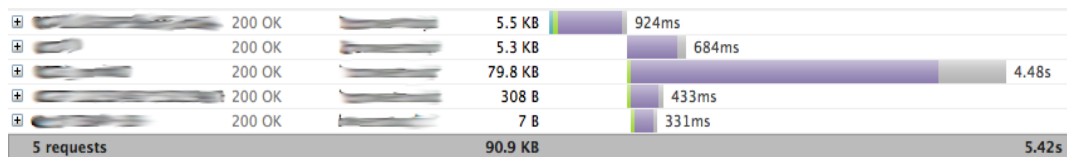
Data size (k)



before



after



How Come?

```
// HTTP and network coding is already easy
//
var http = require('http');
var client = http.request({
  method : 'GET',
  host : 'my host',
  path : 'my path'}, function(res) {
  res.on('data', function(chunk) { ... });
  res.on('end', function() { ... });
});
client.end();
```

Real code (randomized)

```
1 RbozAS1 XCJY1 6sh0eatZ0bMNgXQdIK3
2 1
3 SRbs52PA2G0JtwkHKKj no0SHN5AMVA6
4 VlwsliZz BR7Ai0mag
5 |
6 lPMsyj q58wYIhJrVQATA7
7 eZ0w66J QWdqDBkkS
8 m416Mcu RiJAKVF
9
10 qaSdpH zZ8Ka74telRxCiIyUtUeBKMtW9l0ErKgihnprzP WAL
11 7
12 071n1cDPyQH Z crP
13 IW2xiJEd 0 uHuee9ZEwmQw
14 C
15
16 u67M nM7Zv
17 Y
18 xNoCaN BgFqLJAyOxhsfMV 7 scmgK31eaLIAnwqSkUNdFSxFM0ZGKAxPkA0l6F
19
20 sp bExI0mtpNnLRPB0UerdnZoy9DSHL
21 71V7eYA6clHROIm Q 7RbmIel7VPgzCrJ 8 vb9tmHLrILN9A4
22
23 TR R70W6tS7LiKQxqlxKAGB5ybAp2iev
24 oe1SvY0KEgbCwn7 f bUxl8dD0aCHge7L M OEIfNzuYITexK7t
25
26
27
28 I9HS4JrrBnNDI 5tmkYHffV0Qaat 6 0Zr L5x0Z5bnZPG62YAZN ZW1dLvmeXXszh1yrQ T
29 OubeETVx2 uKNZ7f AiQ T JR4SSgBbjN0iozYyhCJmxr611nrVQeUm WF7goUa7z x
30 IZiWBcliwbc wf gBfds
31 A5Xpb2Z8QBd8mRc2rccGoKGu9yQ82xZbBlxdT5JSbrhdyvrCIpMtDnQUvRjehBlX4
32 4
33 ZZN
34 w446Fy28diUB8oINKky 2 t7IzQPQmi
35 h2veXhG0T2MdXdhbdGXoAy8gCbEE0S i 7DI0cgEc50F1dgXT
36 Yr9Riu50CBesdFZAR9YSCiNd
37 b
38
39
40 e0bHpS9nvssQ109 hr1a0ti1UPD6 9HqTd1Gc t PNrWU
```

```

n7cejlhPnA8twkd GcGCytRu6GvLAUdK RUAQroSY1V5E z AMJXQ
CXhU XYwLpgg5L5jouqasGptY8pv08 ZAGqsZMh
9
1y aBe957rY eT 3Kwv06UGNf24JFiurIBp3
o
di FS6xxEkMC3m nngXoCiiztuBzt00AFqwFpnjSpQfmJ7dX7aj 6SSJT4X7eF1eor06pEtXd0gFBm3ZclEnQJvFRsyKQUB s7woVXNXskPorVKsv0Hp73B
B6s7yzYHByj 4 Aft4x
rA6yIonU08bnl0Yj12hSuVmC6D0lsgLs8lMX
Aqgy5QJ
C
IS
5C VJEVgZrMTqmv1Av7MUv9W0jU32E3ENSb oT zcXbH7q K7gtGy SV 9tT yic9
yha3K3wC V etd UAnoHrpM9QpGtep Kr8ukNQKGb5Ekmz
fc834upT04D1 w UoO zf9adPqN1UKYs6w oWd3LUgdTmA1FG0nkwb
jLVvy9 ADq4cV 2 FTc03bT05Gc7eYpF
ze1u44xGIfo GPIQjsUM m BOPuwE1PFLH0AJaFg2QQRJjwgr05
xs VyTkuaERPQMtNgmyBAaae5d DH Gzwt8cD55mTx fzXqnsxlhsxBTCijBC0
L
lHaZzN 8HRET1rtKZ w ul7iVILL4dvHyhoBRz5Kg2YXLMile2AioD7JbXFivgc9bPFV
Rjw5wG WEpidyH x aEhzB3fIVbJiCqeEQlKv8u5L6BR
oa 7i 0dBdY sE B jrn6mtkfJCM Da wI7 UP8FP gKdf 3bP tMBGUZCr kGFZ x1Vk3R 80Xkr6H35 idDxiqI 0s 3
jHE akYrHr1l D F1
T4 h dD12Qi42sw9m8h2A0iCW 6c hTK9S
W
1fqN5qeg 4 l8LplornkV7cGoJaqE2WBZrCZCEh6GwNoye4Fs0
u
Dp gMuKyn J3t aU1Pp BnUIammuGinDfHv7 FY1 5hDmqnMg wauM5 pk su1 Oi PLO kmh8pD u4arJ0Vakwh5xz 7C4T7728 b6KGK
rz 1dU RKTC 8fLbJP Rm F9 SBD PCCMipl HN Ds qu91S 1Jy ikQm0Gp C7 iq8N 3l uT WyPr yawEhaSPNRma G1GpjhRd
be k0Uzf 4lX KSDCvOof
F9 N28PXGLppqfDDCnUswNdKczJ5JFg0Ex m WNr 0
m
QCLNcpcU V tZ
P

```

UUuDj5HjwOXTRV7g 8tu2lTSEx1nwQZCuH 4 k9V 490b0AQl2Xzw5nV6ixsg QDsQxSk8HwF77MCU4 4
15WdUg5fT 0n5n0e c0oq 1lWFdKC1LexVYcmwbPh0W3OK7nSFUd1T LbmCmFWGZ E
SrIDjMCau4iL6drly3Rz5HzzqZbw7qqj1HTl UkqULEl6Qr
36JNDos8OCFjxi7uot0Q38 Yf Sq1FG4cWKM771U1g
NFrwaNcVr0h6zJjJnl4GhrhAZUVdg7

n
CZc
wrhdhBea5LD0w0tgOX1CA10H v F046nzt
ynefL53qZVS83ApPBcNtcVg4jkr082cficy0EYTF v ERuWQt0h
ciBAh15NqTaEEkXxZRs9aKOiyaieI32ix922oVh q SWIk4JRfuXD
sSz1P4RsSQ6r3vpqFgqEeWR9zFpoDp N Llpw37zgH
KaTVqtTqBAg7BoXqbezP6qbFPDvyBc07 GHCv8eMf8D6S1JfFyz6

l

FZ asPyaqhKFo

B

jp MxId svQLbVVB 58r mX jV1Q8p LF AM XeKpQ 9N37 qnT0Ei cp4r4 rE6zG rYl lwFl eo bT0326JdjwuerMnt
Ey flnq2X53nPrGyZ5hsAuilFkj5
wmN2sb3wb8AEynYuwcF3QYaSUGS D I0ejj

JE s1Q cLVICRu yFzJ e0BrvzeXA TK7cW KY Ns7 fv 1mAki QR I iH17Ayp7 pTf2c D240 ZEIdj ELJ gpzqUD BohVnI ALacz7QShW
YQBRLbw0dp9H3kur18L1NcwbRbLjTtkC0
0VRK8Z8bh3RB28TlJinhRMFgkfc F rCI3J

7X s6b3xmkgJjnBYQrhk8EaV0ZNBQQjDR

6

lMj0Hzi sR15WgZMdHh c YFy j3SgZYUQGgZ tbL7JHpAgRQ1XIyL6 I
CBmSJqDhp Igng7h AJ19 13akmgXS0pLDtkmDt5deGE4sjqdsjhAd ShfqG8pjb k
ZUY1PKW1ubm8PzwLM1UnhNqrlD 2kjUqvYn8w
H9fjfcqYcw409BeWnsVSMN PT Pn7WEGraQzZVWjxH
Vie7cinURqQzHIjrNqPpqJd5m4JPhD

Y

4J2

1GF4wu2lzXYnXAKDgLkWSStCV2 m H6muvcp2C19F83S4Rpfk8ZzCV3eG78RV06nrT6TKi5niH7seNDp0gNPYjyMg4WAPkz0puj2cYvcj0a4XmIJoZAbOYl
QFs5fG6fARFJX0qdgmpd9WLgveN5H N oRcCl
Si3VeYWtVJTzB2Pso98G9hs GULSEDZlbu2m0

Mc 6wliJ dEmLXh g7wZXP0 ISTM2GbJySDm0kcassJo9m808IHseMF 3J n0 sNBTL YUF IE f6qAuH4P

d

W4

NpA0 x
Ve98oVCXvQ03WNjLqPYTG0N3Q9NZa9SvaQuhPcUR QC IEKtNw 271 NxrY6iw eST3RZS8Qx
c
0b ctn1kR1H5k 0tBq8PlwQw cALr dWakSe 6RJW SfJArrI 5fA4s1 HCfqf
4X pK N0jBU k8AM o2dJF 50 CuRv Li OG9 wVyH ilP X0fa VevTN GH9agKJC1C
5F MY Tns 0S XgLza NHKCB Wpcf al BRiV SA nm6 z1ray9 MvIA2xxS F0 FU3JBpXAWudF7
Ma CoFl lpQe7l xZR 9u7q MyoBQ pd fyd XrMky4 US23 n9W UisyE TNb15Zy RI6UIA
k0kLcJE93oNEd8D3RTUfbnhEqkkUQuzx D0 IdHg6p1IRTa4ZQUulFRtVsk0kx
q9 w39Lj0LcGB ZpNhUc3q0IR6prrmX 4G TgPGLv7vM m8219dwEyiQzZG PN
Vs uPaGCySsEi B2pZ7dzD7osEo2k1Y sX FUsXkmZh0lh0D0tbnpm 0l VIWItoR3K8fsfN0NJK
N
B0dzIvDyl8hRx 8Ew7AuK4j1BfEO Z Yrw 2L037rRKQvgVNHJcU 8KhjBazWUKHSmbRbo 0
GIKxD9nx0 rc35Lj YM7P AAARfMrmpyKx4d0o1SVwB992osusHZng LiCan1MdA 1
FWXgVTGmigEnjmHa5WK8qYbLAAKHECauD WKdywin7cL
tEali4NApIZNyZCfkTk2Hn Y6 XIPdnpSwr5o8MIkD
sa9H4z3VDkmIJwfA1i3syWoXUUBm3s
3
sLA
HiamMhLsbqqp9xCGqiTyv P BX4vGnA
7ipwbJkfkDxTcHmNAKjYE20vUUFpG Sing45pJe9JtTCc
v
Vo dT fYe0 jx uy yGk6 uITW ROY dfDN SwSG I7 Dkb MIRuE udDc ji922NWUfdjjzqPvmXq rW j6l aHj 6Jz3AMBBzHq NSySlcu
LV qyG7WysiEoI2HxqYzP1kJ6LZI8JKRD0WM
n
o2TqI0R6uIIPhxd2NLL nhXqMOMVt0GwFCD E biA 24vFsVSMXlq1JWZ6fYaIXCG dA3UuqvBE7SUKWoHJ z
FDM2yG0ZZ zu001w 00km 6IBcGmLGpXPKCJ0KuY0vCZq7n0WxWmeP lsBFsVrx1 C
Ap0dJCPJ7EdkqJ4Axx0X0sRqtAHehITPMNGLUZ lDdFcksgQv
KLUD66PDP8f0Fa90ajjWd zS mM0RVyiQ152UMaM8
gkfKru241uynKLqALCFGCr8t8ygvrmS
r
VPp
pYdRjah6l6evUHN4QrEAJk 9 jKyI8rt
S4FxbPOKmQiZpCNHDhNbjVe um OPKNF
AWjnbKDjeTDnoSo0sDH5EVy b fk16cA8e8P17SsH8cuYEQD1Gc4IG40o1ppn
cr5FKnE3n1YbnwXrKe384XwbHlyVdFECPSA AdGKw1lMFHrvfOnz6
A
El I3L ExXct10v 4zJZNtaHygFIz Yz UGuuF 9LxBsS HMEF

I0 Q8M 1z6W2m 82gWpF pbfak 7c xu0q FzNx bvOE5q ID SkSvg pCEW Dpd4cM 6w IKG L1
 dZ FFBB i4 tajE Qo tQDKQSW 1yBSfp4QPh qZ PVJkY n5evwqq 3H2Y xPz m0VqrRR
 7i jv Bp0 H4 SgY2nc GW6C WVXfXGXjvgoliUg0F796CG9j0YquHxu540mX5q33s0pgGZ kK zs5V ic K06g 0KQ 48
 1S UT KPtx u7sq GKT zse 3Ii25 c0 lQg8 kNtue QLsnNYtGYSQldT
 tW sSJyEN92sUHF5px83PrL1Xpdc2A2cLAMRpF56fZ 67 680glGrXJtAgzWihK z WG
 E
 0c2A13knUJVyd xPiQj6fNPH2IFk e fuQ bcuaSTq0kvmochcnj wP7hjn82Pbnp6q7R4 z
 DkaYBMtQE OW4hqq Sjzs 9L6J9bILzP0LZmctFa5sJ5WijtKwRZKL 6RTr8wWhk v
 jW3brkaaBkGF3JCynX8spD89sWLvqIKeY jaKBH1CrLJ
 Fu2WDnRcT3K73kXZGuLRYP JT DKz66jL62w80cCsF
 H5faStrMF0QrEWHMsk6rPkcmYngkj
 P
 JaS
 8JlWlj3fM3BF08Gn2CfVB W IZYQTF5
 C9J3Udmj4F1T7ccJhHk7lY5ofS8ZL EMSAeAmCDrueuMnK
 p
 E
 ZB 9SKi txj WYF xx xjo X9eYz
 KiWeUs39RIXeUHbcz 0 bb
 U
 na ueiHNdyMfYc8ByFus6XW9Jag308zV
 8i KhGdQVH H zsPs7
 9g Kj20GXiuVzFF
 ydd4TypBc7S1Gdq Z9w9 X s3wG0sEvb6eg0mzkfV
 9Am4zaRZYzwMUUH
 C
 Qnjx
 S
 9YQconheW17mb7SH8ytR75
 0
 C
 j0gVNUu jihh GJ9m1K8EawT9H9z4yLaLMbkxvby JEKjgxdnx5KxzWLL JBu7in0JiM
 M
 jPJ9Ro50Nji2 oj kAYpc
 b
 hLu7JZugpdwaeHWAJWLtG1huY2h9eZFWYFLm
 bEeVYgJ
 Z

bm1oJf2rJ0bJkrMtda4s7N2z S9lEacKMtthNrbVNUfhX2jkZ j OqASKiG3ml1zQ1SldjjPYA2DVMglDMLsiuVdaacLJxSFHE4ES6hwbfxGn0y
rD BSuBIYDmzYb13iZe33oI3wDpE Bo LbwU FF DSiyGbdxdJbc7Hb3Xkg4Zsj7YEps6TLFq

p

MDBsk2ezKrz5Tr7IK2r78P6fzVnYzArTctZ2EvWwMMSxDV3lTfT

Er ab6VNB s1T LQmDl Jh9eZktpmGJctDhS bvK M0Q8Y1lJ PCAeD Ro Q4K ue CAO Zsq1QB AbUDp9d8Gnmfs8 5vHg3rjS xaCGy
SN g9i AgV7 DIUCQZ H3 HT msz vMAWTI4 lr aM eOPa4 36R UnwBp7R Zs hcGU nG kg r2Ik b00NGqdB1yJN K21bxBcl
er qVzI8HUWJ0CtaFPwb8zHWB63SVTARUSQVexymSqTjzUG5ssrQfx p AJH r

X

GktlSsm0lUD7sSTauJTOXixVGNlxA6fsh4vrCken80gQNAqbvS

k

K

MgKJ 1InXjoqf52lwqVQNORPFNd7izD WK dL76n

M

oP Z9Jazf MyzoC CidG kv6t63RpYtr8rd1 Bg7p aE Rxx2 wraH paw5 PD7Y ZKu XMeu kK4Wjkc TXafLxAG sB EMRe B843W3g

5BdEs fWilIceZKr5eG E0G9uwYvqhFEnj07vH

ySv1dUSQoylgpjnk4ehsSBTEH BK1Mr7INIWRiCdYnGqdBYPZ86xtzpi1BWABR0Ebt

qhH5gEJxxqsK7aobn4x03KWHJ5kLSxjYN5GeWPJnfoBC07x

3jszaMkaWP2t6TVaHs

mOUvVgqi

ddcgCw0E4uFSLzoze zbW 1K khCCUsvB CyuLtot1rjp0mz0Kgxz6qT

7RnASDWriIvin6ckz rOE Yw FovrdVWu LlnBNnu0g59xnW5puyX3h

CsU1E0bYr1VeXMHsfUKYF4npH7H

uJpa0B3iRUJYlFvEUMfYQhvwe5mujdXa01

V6iFXzrL1rNNR8SAK46NJzizjUwulZaP1JmD2q5QKDdGTSXugGhbFdV0

Yma97j5m1

2dFiE08dKRExAuClp22P7oDeFvZ0GsTfINflwrM4du63

1Yp7LinuWBR9vHtW7x4fBL

bNcBHs4tCMdgU5aCjp9gTqxFBA

py 0azJHtFG8wBhFCUezYbMZ3AAZQqpPBz6sq3YaRDBV9Y0X zW 1EoEugarhQ2nmxbv6Jmq6E8dXT3pQnZTYrD kp

OMVNLBCWoEkoebS9ccAPqTqsMzVsZRUgb6Et2kTIRdGE 8Z yeqGtTD3TjRPCEzczlPpAWK3GeexDAUHNm14

f

LA0Beu8SPkQBnjI2Sh2HFWU9r5mCR7nGK3feihHl h TTs7z

a

E

wh2mc7Mx9NBW0Vt 830DmyKLcbgpgTU G Pzz8fHsTwUuAt1N9E2pV30XZ0lfdi8PPBGyJYKeKGY
 wn YbdvW5xXgBvrFTzt bg YKeV Iu nSFgEVvCrFy69WdmjT5KPT5q
 5
 pnww6oAh5vjtG3xi8ACXFtSAW5BCKcRv7mMt0CktX1 0cXW4 QDaPLdz54oLx1TwTeKbChinhaTSvVY2Q7NjwX06l6474VByKouTAt5dIz
 g
 eVYSZupi3lbZ8eNvYlCwe Pm2nFuu9Rm9pG7mWjy T nGrFsuuNF3IOWE3Rf6PjZc7beTyf6wP9wghZWQY1VGCKZb5bZw5Xwv
 NZ qjBY2YyVqweJEWwSIn5 pm N0JYk
 t
 S3 LX7UrLs1VXpZbxb4daS2mrEXylYw
 1
 h7IqiNRlP1oLKVV0YRluxE7FmvmMGcmgYGceIBw31oUV7d
 4viBhyfEw ye7i1 J BQaLVlgTu4ilc3LCJSZjwT70
 7M lp3NArKisZHMr8oQFIK5ZBGj
 xfxcEHf545RVNGVGCSpq7V1FgaEp19HHi7jAR4zqFF
 3y hv3Z4bE8s3cb3Y1iuPiK51cJ
 D
 OGG6ljdh7QyYNchphy02NTzwR g UY5qgbU6LEc1djB
 a
 1
 vSCm
 u
 Zv2UVuaQ38T I nQTMu
 7
 x
 k8ApaW5BUXYsmHORGWmH5 ilpHPBcqwprrZ9CIj1 U iczpkXi9bVpMjXud3QQdEr5CLg86C1zBAmgkloD0ZLmEFf3Ifb2Ety
 cx nWwRqdJa0Ridqc7doNO 4V 86szo
 u
 V0 TuxCUAujtZne7UI7zx4pjgpCtZYZ
 0
 5AlV78SNb069YMjNWFICAwE k q4pxnTcLTpBJpS7tj2tdWEF1JHwSL
 Y
 TV jZDU2F 8k0JkiJ

G

ZZmG5YoGntIsL83n0C6uAfa0Lpx aUhIZKF5VHiC8FqWih6fqQhMBh1 w 15N7ScFw01H0dLSKLh6AAC58rJAzkWLtw6XSkD33g2CAHBCP5scNy8FIXyk3aPV5KH
sw sLLCKieQ4F0PCKdsumor3nnlRdSc iz Xq9Jc

A

aW DFXejRsl2lSGNj0p44wCxImvfidY0Wv1KW6AQ

t

c2SjHeZAvz57jfUnLvP64tmwdjg9qGbw0s5QRgyGApAALuKWM9TZqJ rAoqI 2MlgFnfjwB0c54uN2P9TNvZIPXQIvBo5I03hTC6uqfmGoWJxo0cgQfDMO
ST958hcmo5MMY04mWAgUDEh xU gG0z 82 6NaRZhLkSvefD0rANoYIypF0Ze90E06nhQ dk tWpcN

V

pR uh RK 6vVj xMq wkuvM1cYZhu 4kQ gBwCTP vF Zn QyiE
rTDDca JA6dZzk l aJEC4MiQTS0ep9bdRGKnIl3qjipGRLE6Zm03qTnRaEFGtgXmdwSCsKHlNgEUmi4RAzqUT0m80XG013XUyCUWBVLw
qq PrnBxqR7 Pe Js2AR

a

hBE0DMJdqsYIVas1zAqLqtXRJyvXWD5rFyKniSlPSTGD4RCEQOTvh0S42DRqhFWNLvMq82QH1 OAA8vd1 BSb0ADUShJISe0XpPYm02Ihs9EQI

I

q

o1pP

J

oJ Qp1 VA AKDt znx 6hpTg
gokDxm czRiigu R 04K1UxF66hPrmYXpah3wuPdEMANZrvXKTlWUwgDGJmPJl4V8BKlPFkoZ8PWUyV9lUmRNupgZ3GoyA3AH4Eb10
2v 8z3bv9Bh Oa Zs4WZ

8

TtMfPugl5bATEjMurqYF N P0Z 5o1TfqTLKYZMPciRkt
q10pZhrLoiwWD4qyIgVEc20gXFWpxveupo9Jj64HAQsqBAef433tGXhXPcnQ527gmwwsP6jx6 GHVdjVNX nKayuSWdlr8dqDab0go13YC1R0l

o

R

Z

QSgx

R

kuynWS7Nnb8 q e2T9t

K

3

LbnM5JVLX0x2oQDqCm807qEj2PYzbT6i9HIM

R8qM5TaiiJiv QhHti70CyyIgDQIwuo0sAf2MDMJVlxm vzTcvFaCEXWftr0ED

ZH

0

e7

The same in ql.io (randomized)

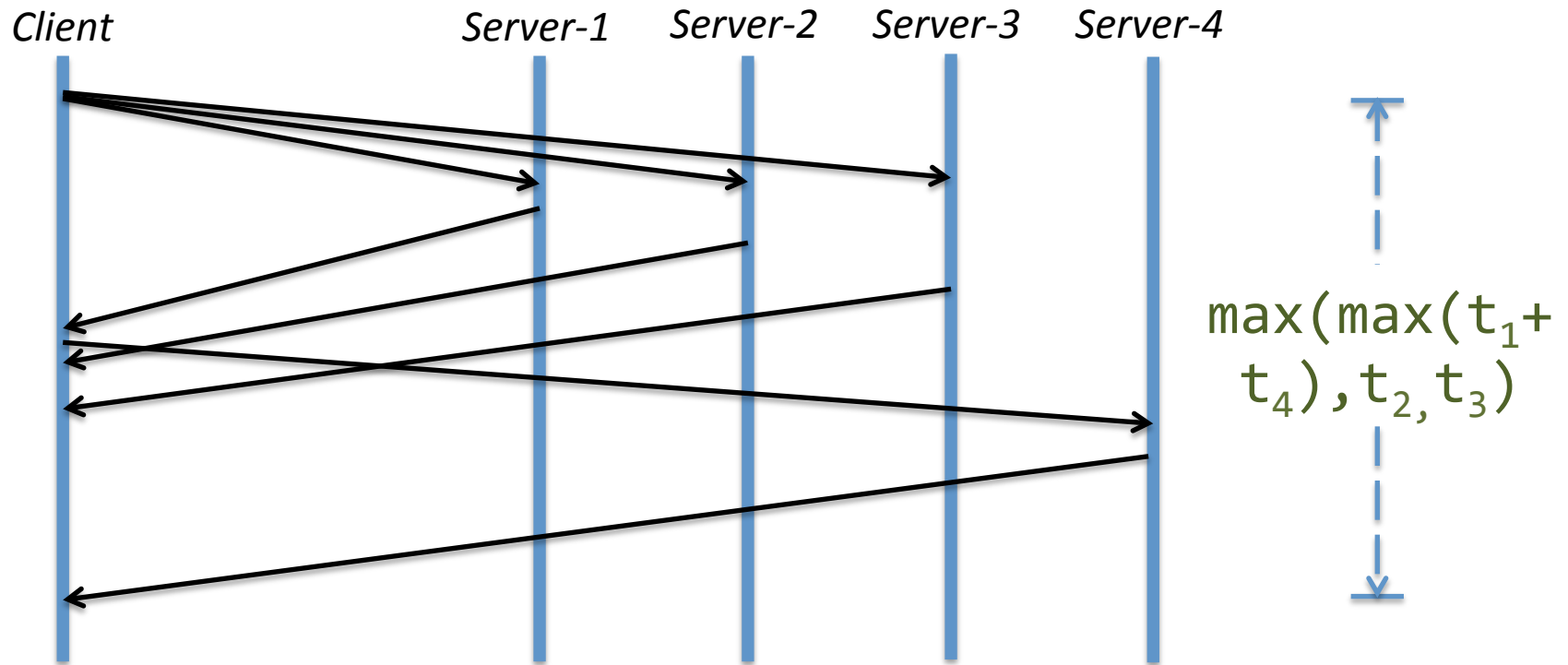
HFRwni G NlXGNS TSMeB7 A9On vtwZ
93TbEBy6 Q ocpBxgpH3 Pu4ju fi
ZsKb W RkIs5b z UAsS QK3nyJ68IhT
tUH 8i4fwR S Hut69mnCHAO
ufyx w CZLOtN 9 PvTU sPd2lMVDV42tRAfIoPM56H1hE tGz5s
kmekNeyrai5Sk1C 5 TstTKDhFb
OLy 5KQ5oz A MiZzQJSCbEv
aLr068KLleE X q8cwPm 5 nZpH 3jpeWcIpkTTIjGsZovq7 fR4Hn
dz3Lhl o MfdTDqpFVdh
aiPOsj2fO9 w fWD3mv p ORHX Bq4xIMvLGjMrgnC6JpBw1S5
HDwoI CwhI09 z 742rMEqx626
ZH0qwtN g boU4fU W QYKf F24BKGrFfg0sfhkc8U4aZfL4bn
kUNmG vm6odt 6 YaC6b0Ff4gG
Ox4Jh0 6
aXtsEg G LULJL3k2O
WeRAMe d 9GlF1XJM8
9oicQwaHnMp7n U Pjnojj5kdhD0sZzh
Pz3HHpnBy L OlVQMpHAILCH
RF3vwaFHarZR Q i2Ofa38U9ylvve
cE

Mobile and front-end

5+

requests per UI paint

[Jan 2011 ad hoc testing on an iPhone] Bing – 4, Google - 13, Mint – 26, Netflix – 13, Amazon – 2, LinkedIn 5, Facebook - 9



Fork/join dance

I/O

Parallellizing
Sequencing
Joining
Normalizing

Writing such code once or twice is fun – writing tens of times is not.

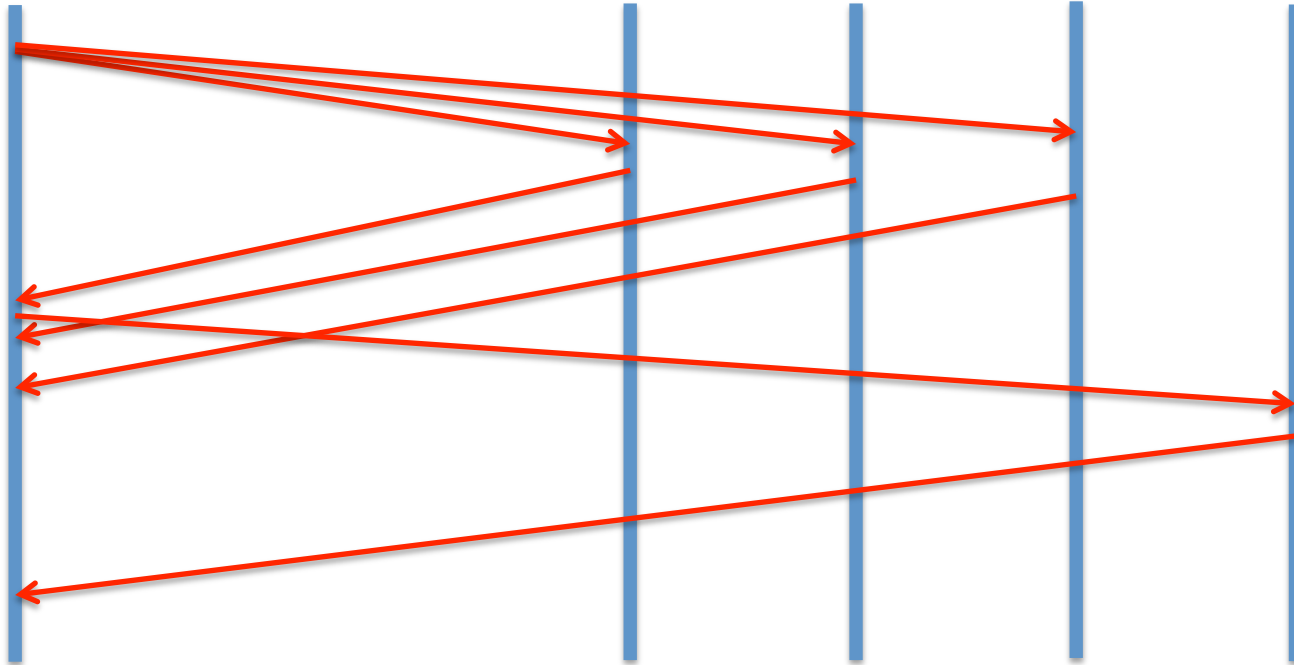
*Native, mobile
Single page apps*

Server-1

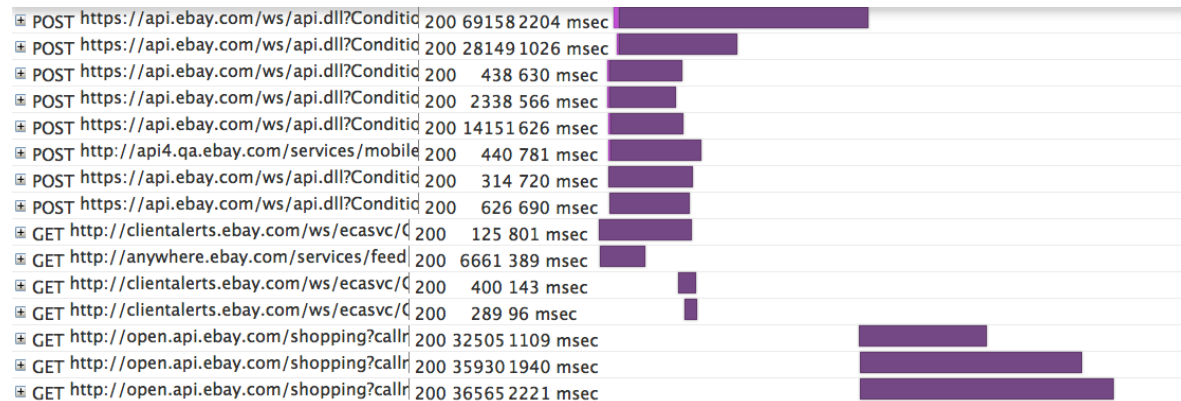
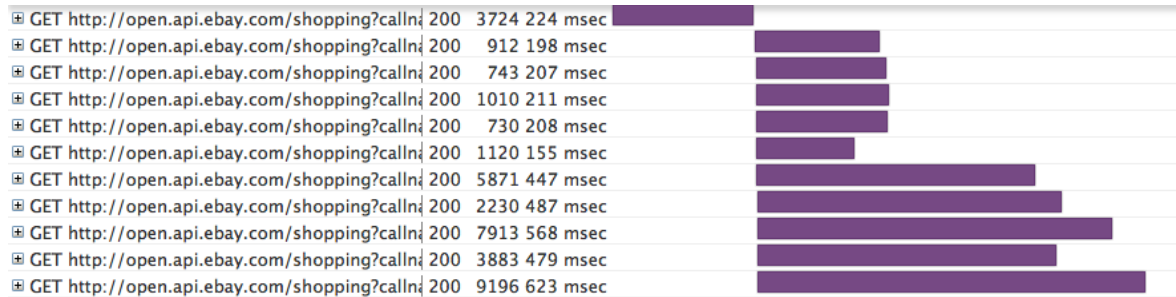
Server-2

Server-3

Server-4



Bad for far-away clients



Many requests and connections, high bandwidth use, high latency, low reliability

Easy Button

HTTP CRUD to SQLish CRUD

- create table for each resource
- select to read
- insert into to add or create
- update to update
- delete to delete

HTTP requests with one line of code

```
select long_url from bitly.shorten  
  where shortUrl = 'http://bit.ly/uZIvmY'
```

```
select Url from bing.soap.search  
  where q = "ql.io"
```

Designed for the lazy

```
// GET on a resource bound to SELECT
create table bing.search
  on select get from 'http://api.bing.net/xml.aspx?
Appid={appid}&query={q}&sources=web'
  using defaults appid = '{config.bing.appid}'
  resultset
'SearchResponse.web:Web.web:Results.web:WebResult';

// POST on a resource bound to SELECT
create table bing.soap.search
  on select post to 'http://api.bing.net/soap.asmx'
  using defaults appid = '{config.bing.appid}'
  using bodyTemplate 'bing.soap.xml.mu'
  type 'application/xml'
  resultset
'soapenv:Envelope.soapenv:Body.SearchResponse.parameters.
Web.Results.WebResult';
```

No Async Mind-Bending

```
# Sequential
minis = select * from finditems where
    keywords = 'mini cooper' limit 10;
return select PictureURL from details where
    itemId = "{minis.itemId}";

# Or parallel
keyword = "ql.io";
web = select * from bing.search where q = "{keyword}";
tweets = select id as id, from_user_name as user_name,
    text as text from twitter.search where q = "ql.io";

return {
    "keyword": "{keyword}",
    "web": "{web}",
    "tweets": "{tweets}"
}
```

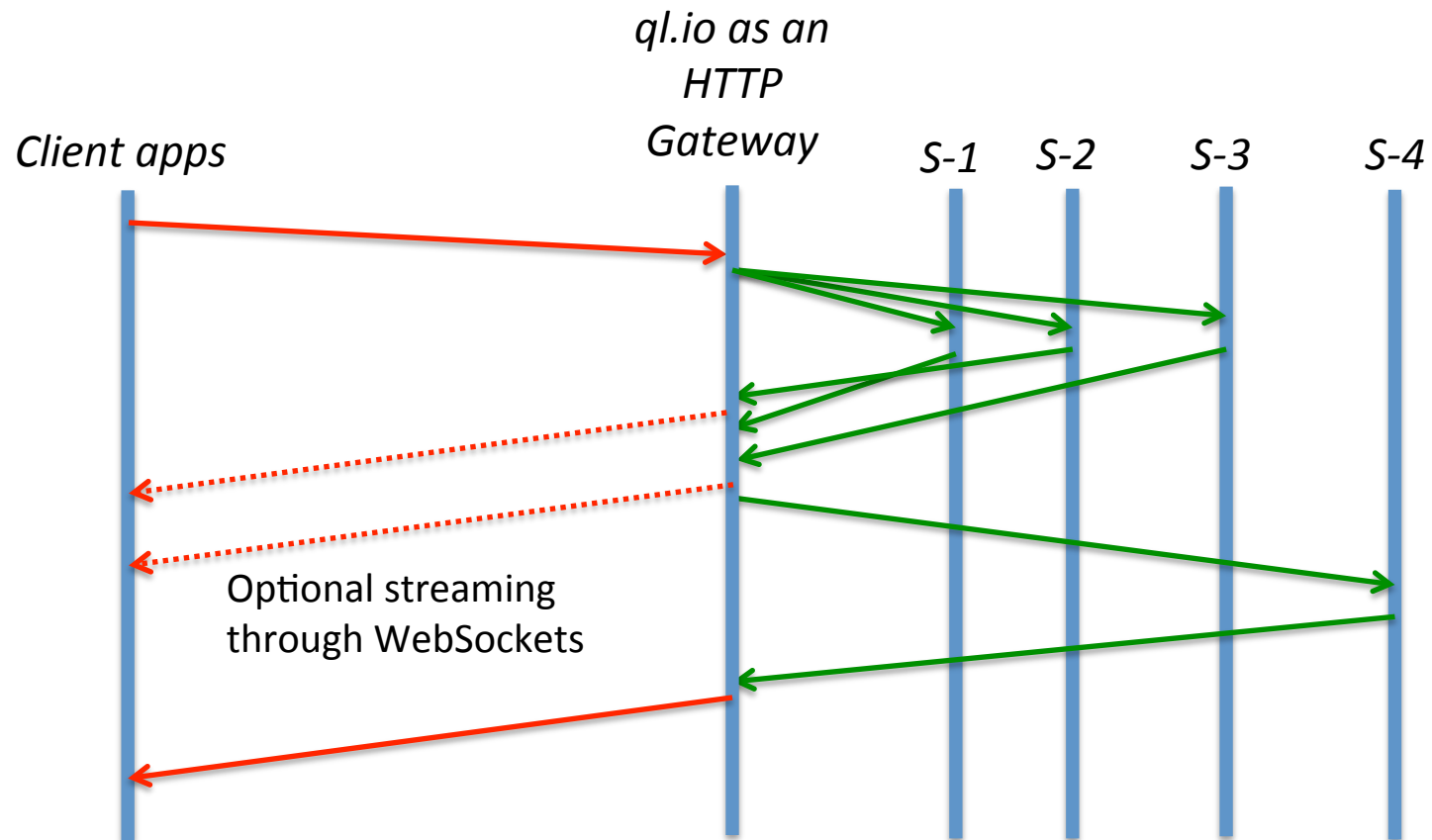
Implicit Fork-Join and Routing

```
prodid = select ProductID[0].Value from eBay.FindProducts
    where QueryKeywords = 'macbook pro';
details = select * from eBay.ProductDetails where
    ProductID in ('{prodid}') and
    ProductType = 'Reference';
reviews = select * from eBay.ProductReviews where
    ProductID in ('{prodid}') and
    ProductType = 'Reference';

return select d.ProductID[0].Value as id,
    d.Title as title, d.ReviewCount as reviewCount,
    r.ReviewDetails.AverageRating as rating
from details as d, reviews as r
    where d.ProductID[0].Value = r.ProductID.Value
    via route '/myapi' using method get;
```

How to Use

As a Gateway



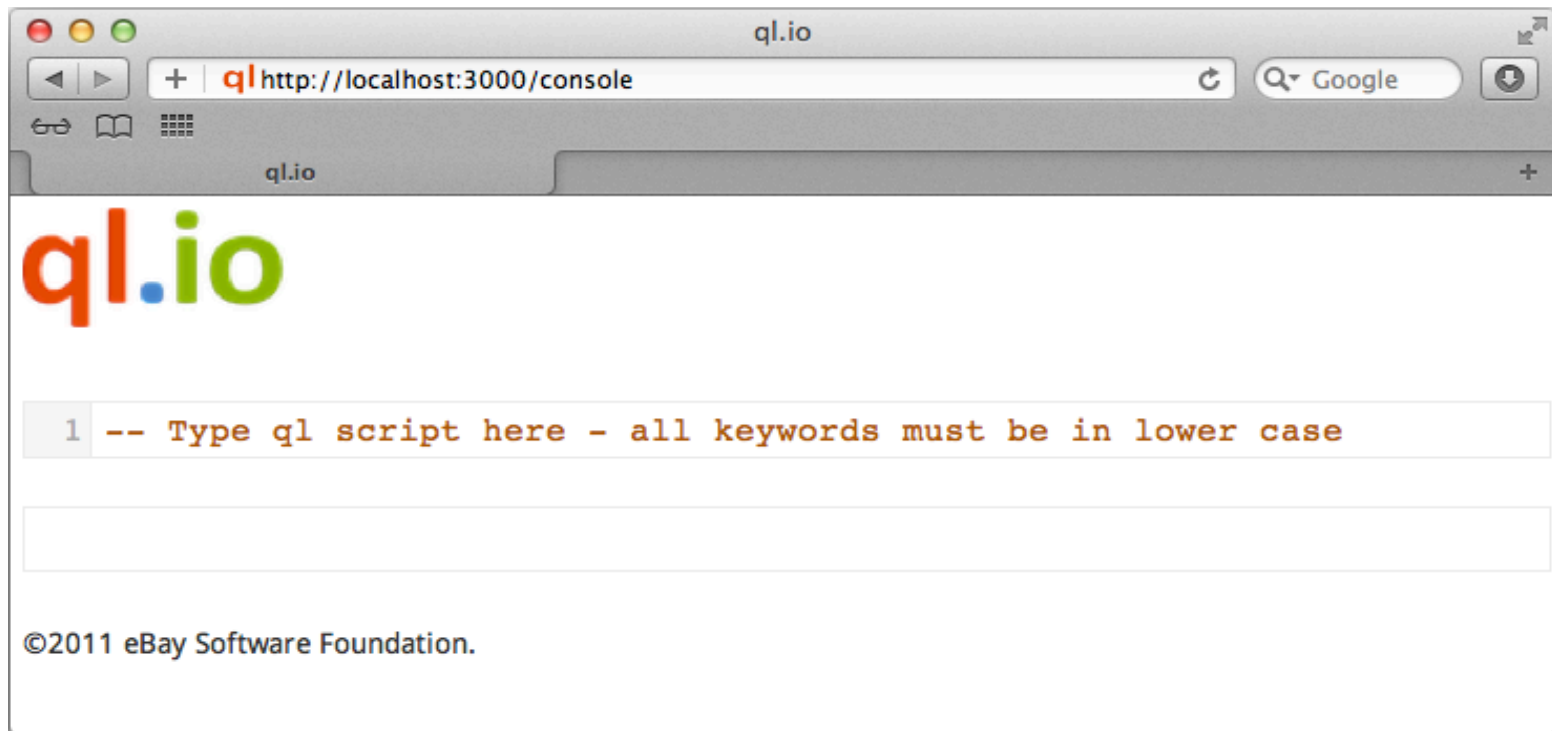
Node.js Module

```
/> npm install ql.io-engine
```

```
var Engine = require('ql.io-engine'), fs = require('fs');
var engine = new Engine({tables : __dirname + '/../tables'});
var script = fs.readFileSync(__dirname + '/myapi.ql', 'UTF-8');
engine.execute(script, function(emitter) {
  emitter.on('prodid', function(data) {
    console.log('found ' + data.length + ' product IDs');
  });
  emitter.on('details', function(data) {
    console.log('found ' + data.length + ' details');
  });
  emitter.on('reviews', function(data) {
    console.log('found ' + data.length + ' reviews');
  });
  emitter.on('end', function(err, result) {
    console.log('Got results');
  });
});
```



```
mkdir myapp  
cd myapp  
curl -L "http://tinyurl.com/7cgglby" | bash  
  
bin/start.sh
```



<http://ql.io>

ql.io