REST and some other stuff like that

Lauri Jutila
Matias Korhonen

kisko LABS

First you have to know a bit about **HTTP**

Request - Response

Server

Client

kiskolabs.com,
I want **/**

HTTP 200 - OK
Here you go:
<html>

HTTP servers communicate success or failure with status codes.

Status code types:

- 1xx Informational  You can ignore these
- 2xx Success  Everything went as expected
- 3xx Redirection  You should look over here: <new location>
- 4xx Client error  You screwed up
- 5xx Server error  We screwed up

Most common/important status codes to know

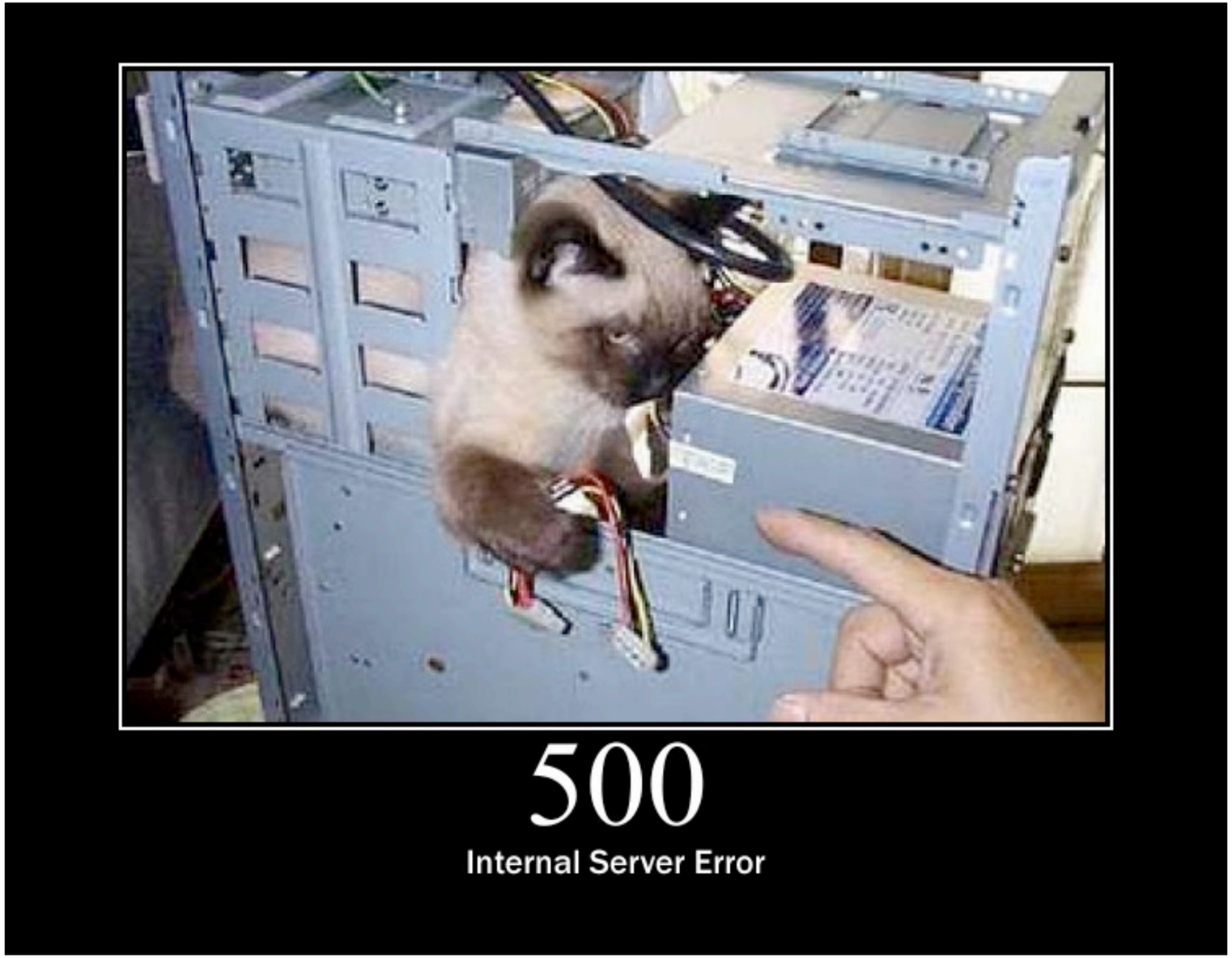
http://www.flickr.com/photos/girliemac/6512628175/in/set-72157628409467125

http://www.flickr.com/photos/girliemac/6519540231/in/set-72157628409467125/

http://www.flickr.com/photos/girliemac/6508023829/in/set-72157628409467125

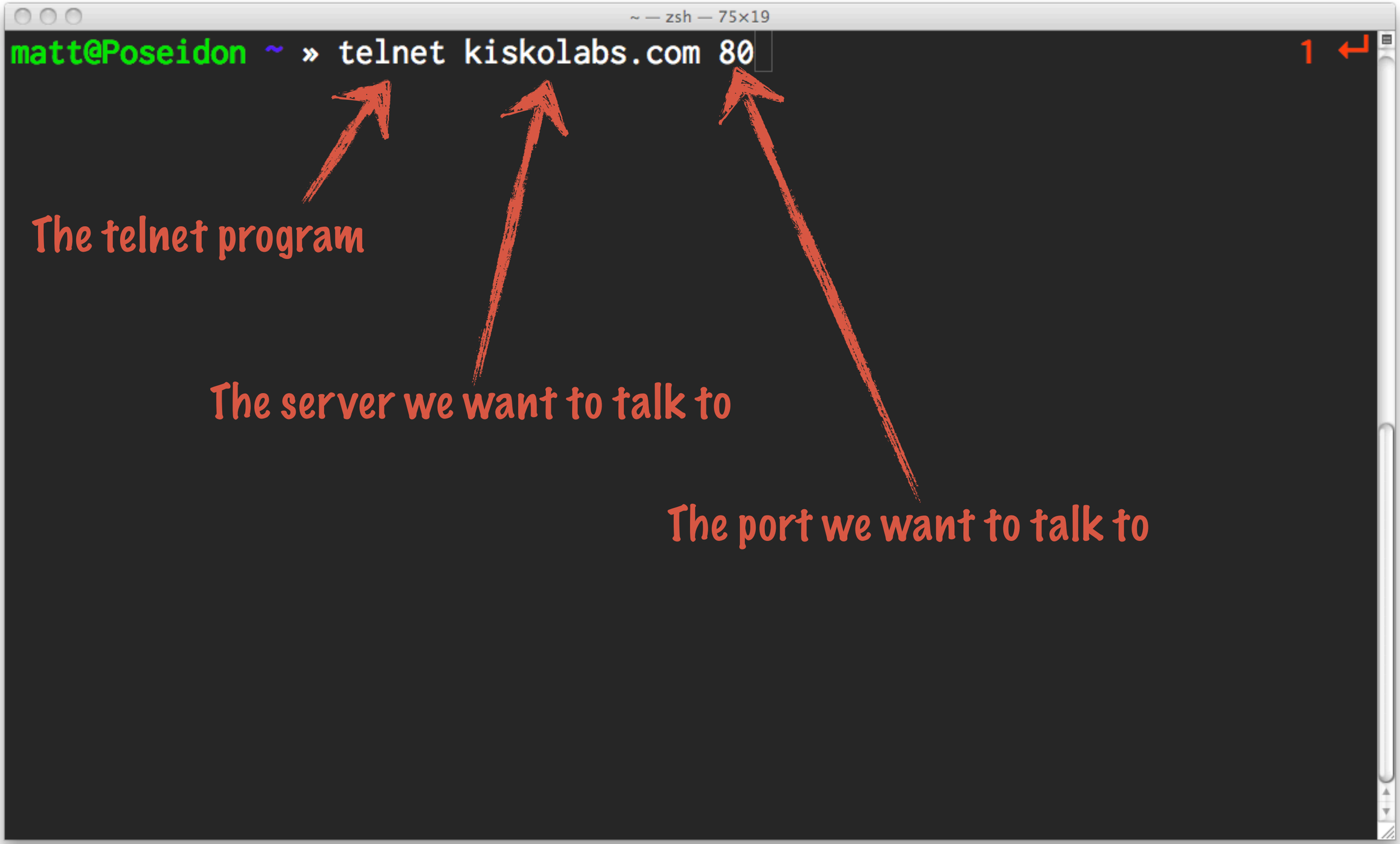
http://www.flickr.com/photos/girliemac/6508023617/in/set-72157628409467125

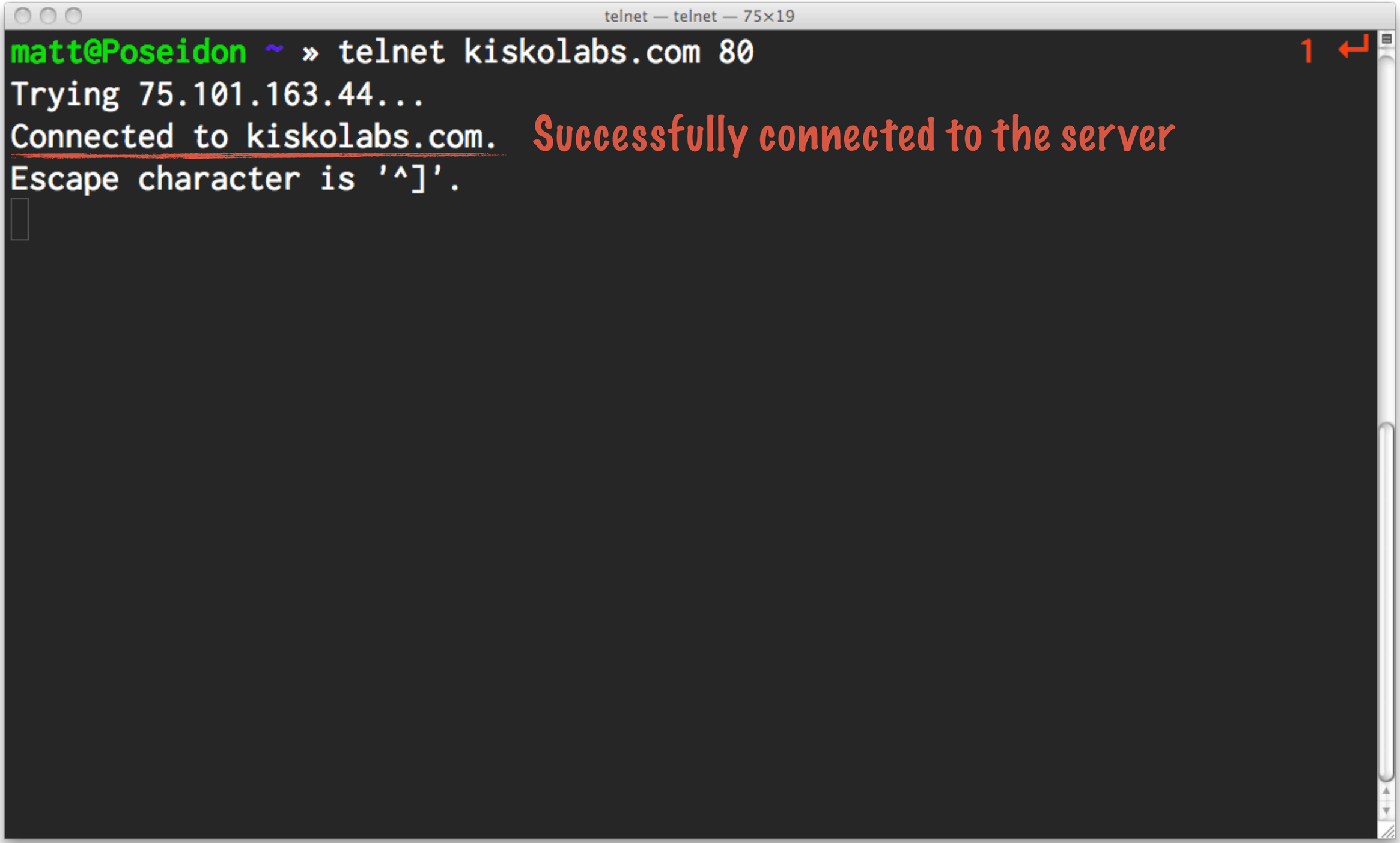
http://www.flickr.com/photos/girliemac/6508022985/in/set-72157628409467125

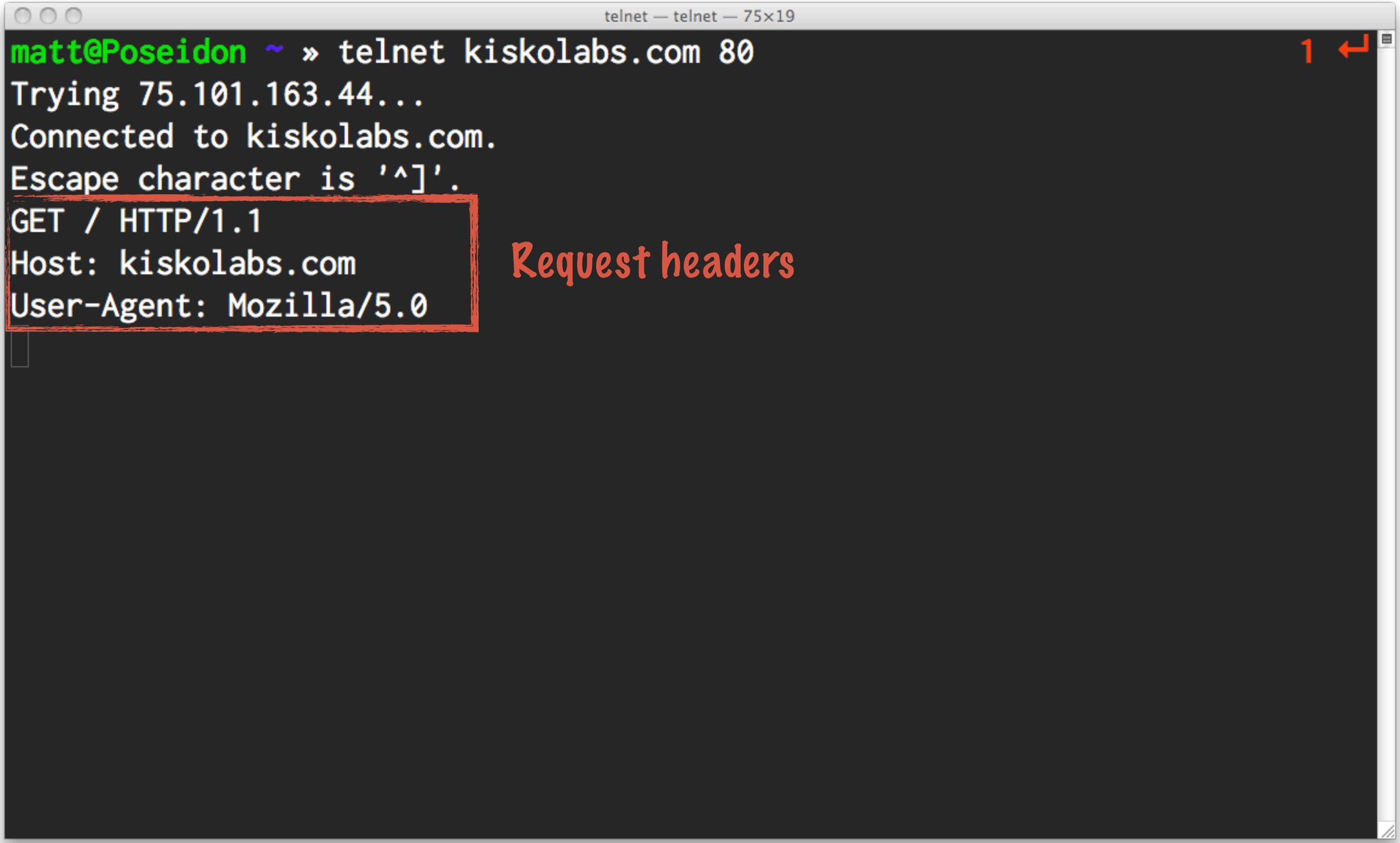


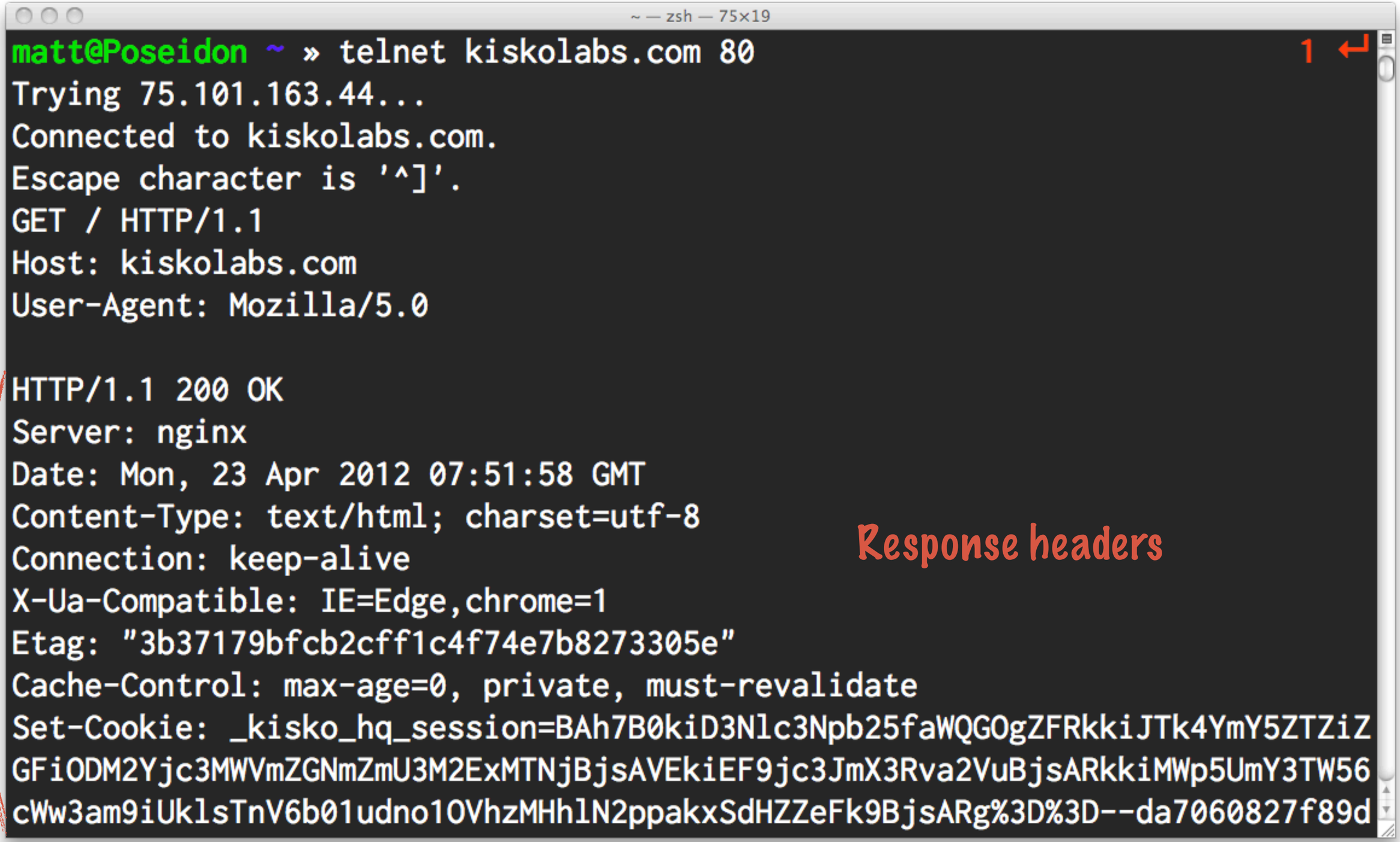
http://www.flickr.com/photos/girliemac/6509400855/in/set-72157628409467125

Making a request with **telnet**


~ — zsh — 75×19
matt@Poseidon ~ » telnet kiskolabs.com 80
1
The telnet program
The server we want to talk to
The port we want to talk to


telnet — telnet — 75×19
matt@Poseidon ~ » telnet kiskolabs.com 80
1
Trying 75.101.163.44...
Connected to kiskolabs.com.
Escape character is '^]'.
Successfully connected to the server


telnet — telnet — 75×19
matt@Poseidon ~ » telnet kiskolabs.com 80
1
Trying 75.101.163.44...
Connected to kiskolabs.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: kiskolabs.com
User-Agent: Mozilla/5.0
Request headers


~ — zsh — 75×19
matt@Poseidon ~ » telnet kiskolabs.com 80
1
Trying 75.101.163.44...
Connected to kiskolabs.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: kiskolabs.com
User-Agent: Mozilla/5.0
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 23 Apr 2012 07:51:58 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
X-Ua-Compatible: IE=Edge,chrome=1
Etag: "3b37179bfcb2cff1c4f74e7b8273305e"
Cache-Control: max-age=0, private, must-revalidate
Set-Cookie: _kisko_hq_session=BAh7B0kiD3Nlc3Npb25faWQGOgZFRkkiJTk4YmY5ZTZiZ
GFiODM2Yjc3MWVmZGNmZmU3M2ExMTNjBjsAVEkiEF9jc3JmX3Rva2VuBjsARkkiMWp5UmY3TW56
cWw3am9iUklsTnV6b01udno1OVhzMHhlN2ppakxSdHZZeFk9BjsARg%3D%3D--da7060827f89d
Response headers


~ — zsh — 75×19
Cache-Control: max-age=0, private, must-revalidate
Set-Cookie: _kisko_hq_session=BAh7B0kiD3Nlc3Npb25faWQGOgZFRkkiJTVmMzIyMzEwM
mJmMWVlZGQ1NTA0OTJjYjgyNmUxNzAxBjsAVEkiEF9jc3JmX3Rva2VuBjsARkkiMURVd1p2cXVI
UEFuajI0R2pTa1laV0l0MFFhWkRVQm51WVd1a2JMSXZldDQ9BjsARg%3D%3D--890d9c287cfaf
f60e2b63900a75c66a7ea824b5d; path=/; HttpOnly
X-Request-Id: 4b0cf171560de7bab6dc84f567bc0a83
X-Runtime: 0.029851
X-Rack-Cache: miss
Content-Length: 19203
X-Varnish: 1371030090
Age: 0
Via: 1.1 varnish
Response headers
Line feed
<!DOCTYPE html>
<html lang='en'>
<head>
<title>Kisko Labs</title>
<meta charset='utf-8'>
<meta content='Kisko Labs' name='author'>
HTTP body

REST

Representational state transfer

Defined by Roy Fielding

The term representational state transfer was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation.

http://en.wikipedia.org/wiki/File:Roy_Fielding.jpg

There is no REST specification

Rather it is a set of guiding principles and constraints

Conforming to the REST principles is generally referred to as being **"RESTful"**.

REST architecture consists of **servers** and **clients**

Clients initiate requests to servers

Servers process and respond to requests from clients

Requests and responses are built around the transfer of representations of resources.

A resource can be essentially any coherent and meaningful concept that may be addressed.

Resources often do, but don’t need to, correspond to your data models

RESTful interfaces rely on HTTP verbs

Mostly: **GET**, **POST**, **PUT**, **DELETE**, and **PATCH**

Idempotency & Safety

- The **PUT** and **DELETE** methods are idempotent methods.
- The **GET** method is a safe method.
- The **POST** method may have side-effects and sending a **POST** request multiple times may have adverse effects

http://www.bocaratontribune.com/wp-content/uploads/2011/08/Traffic_safety_effort_picture_no_caption.jpg

Guiding principles for RESTful interfaces

HATEOAS

Hypermedia as the engine of application state

HTTP session state

HTTP is a stateless protocol.

Identification of resources

Individual resources are identified in requests, for example using URIs

Manipulation of resources through these representations

Hypermedia as the engine of application state

Self-descriptive messages

Each message includes enough information to describe how to process the message.

More resources

- **How I Explained REST to My Wife** by *Ryan Tomayko*: http://tomayko.com/writings/rest-to-my-wife
- Implementing a RESTful API w/Ruby - “The Right Way”: http://j.mp/RESTfulAPITheRightWay
- REST for the Rest of Us: http://developer.mindtouch.com/REST/REST_for_the_Rest_of_Us
- Rails Guides - Rails Routing from the Outside In: http://guides.rubyonrails.org/routing.html

RESTful Web Services

Web services for the real world

http://shop.oreilly.com/product/9780596529260.do


kisko
LABS

- Established in **2007**
- We only do **Ruby**
- There are now 9 of us:
  - 3 designers
  - 3 developers + the 2 founders
- Everybody knows how to code.

lauri@kiskolabs.com
@ljuti

matias@kiskolabs.com
@matiaskorhonen

Helsinki Ruby Brigade

http://rubybrigade.fi

Frozen Rails 2012

September 20 – 21

http://2012.frozenrails.eu