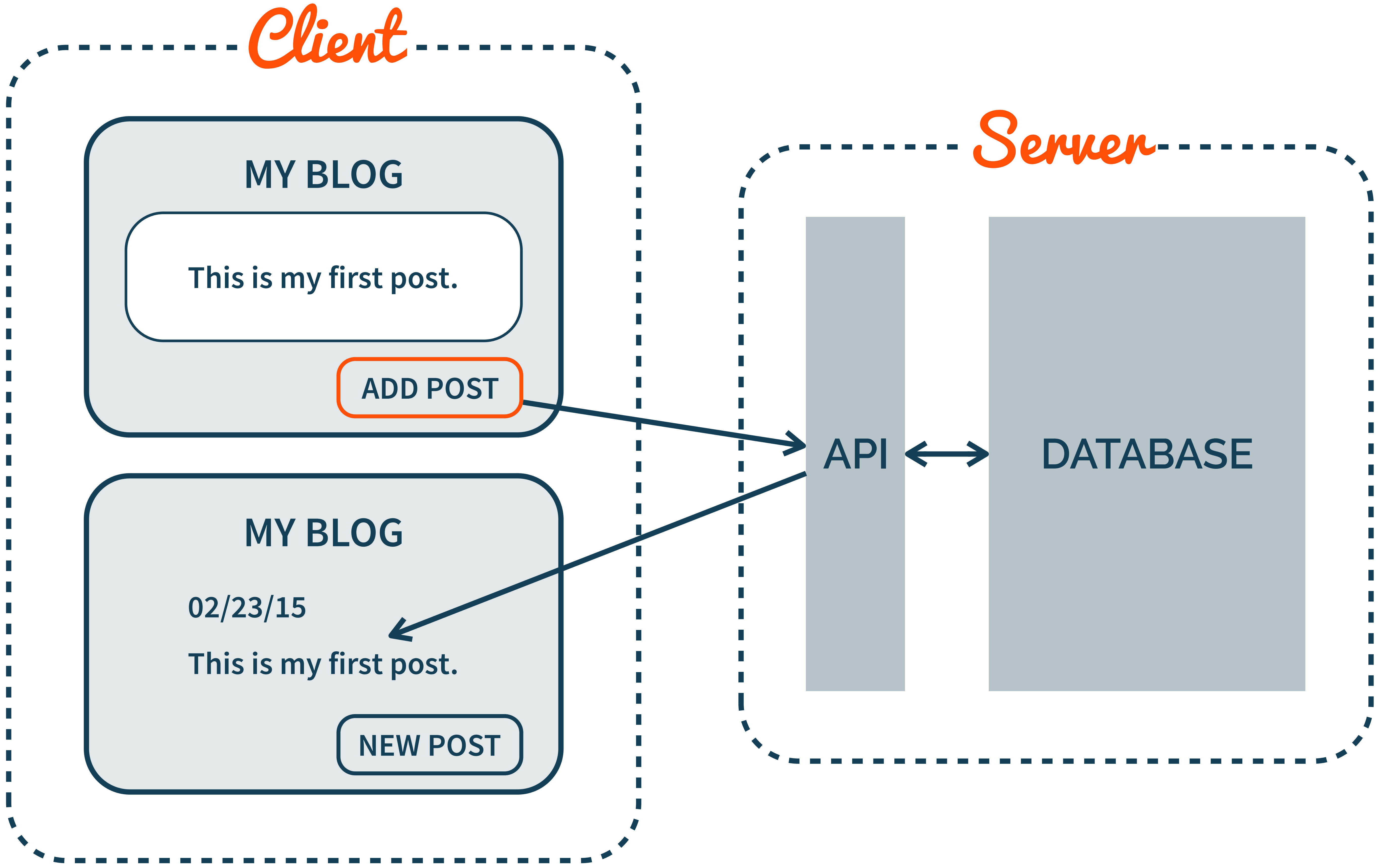
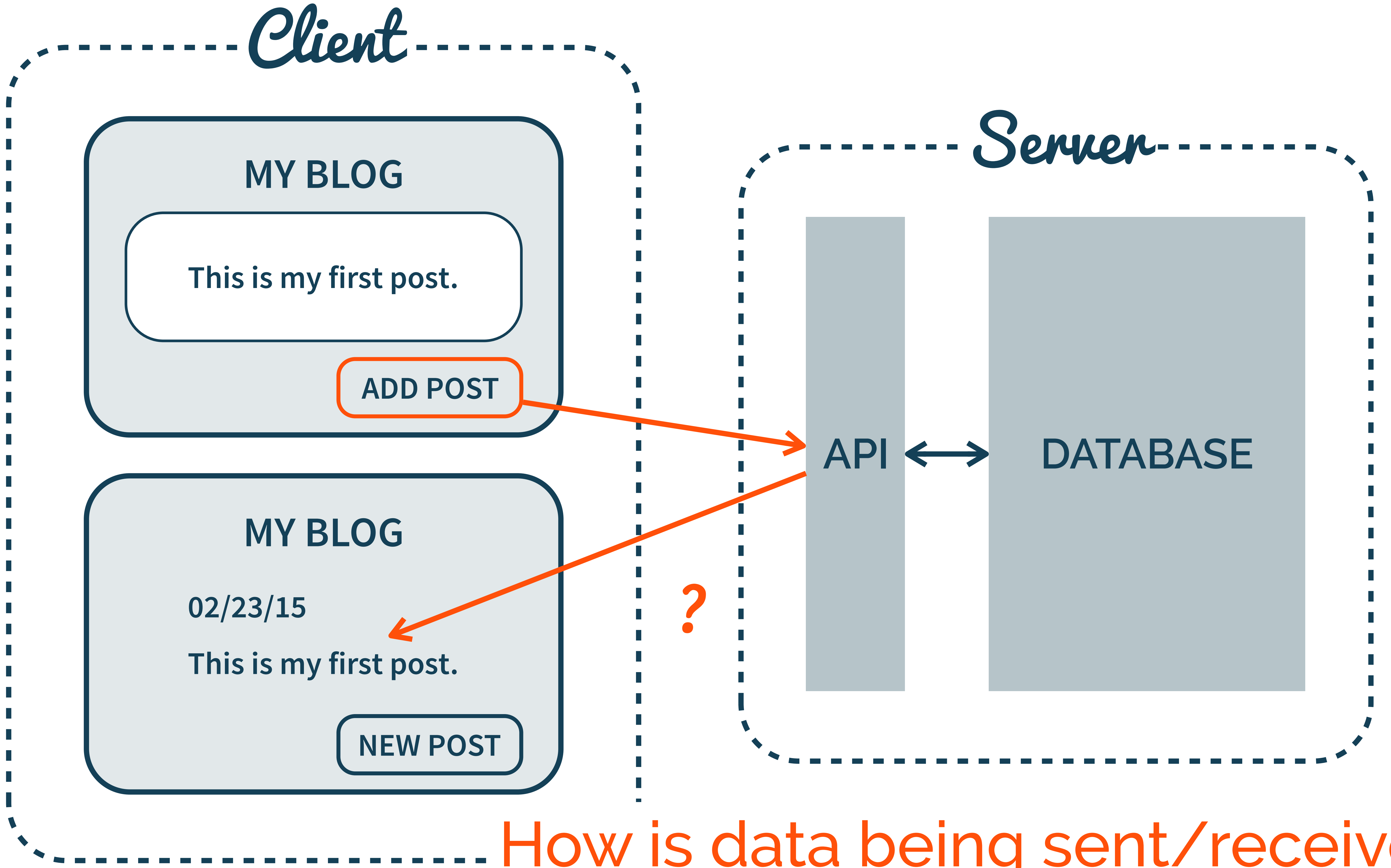
DATA BINDING

Client-side View of Data

So many frameworks, so little time...

www.smashingmagazine.com/2012/07/27/journey-through-the-javascript-mvc-jungle/


Client
MY BLOG
This is my first post.
ADD POST
MY BLOG
02/23/15
This is my first post.
NEW POST
Server
API
DATABASE



How is data being sent/received?

Http

HTTP

`Hypertext Transfer Protocol`

request-response protocol

sent using TCP/IP sockets

“all about applying verbs to nouns”

nouns: resources (*i.e.*, concepts)

verbs: GET, POST, PUT, DELETE

web.archive.org/web/20130116005443/http://tomayko.com/writings/rest-to-my-wife

URL

Uniform Resource Locator

type of URI (Identifier)

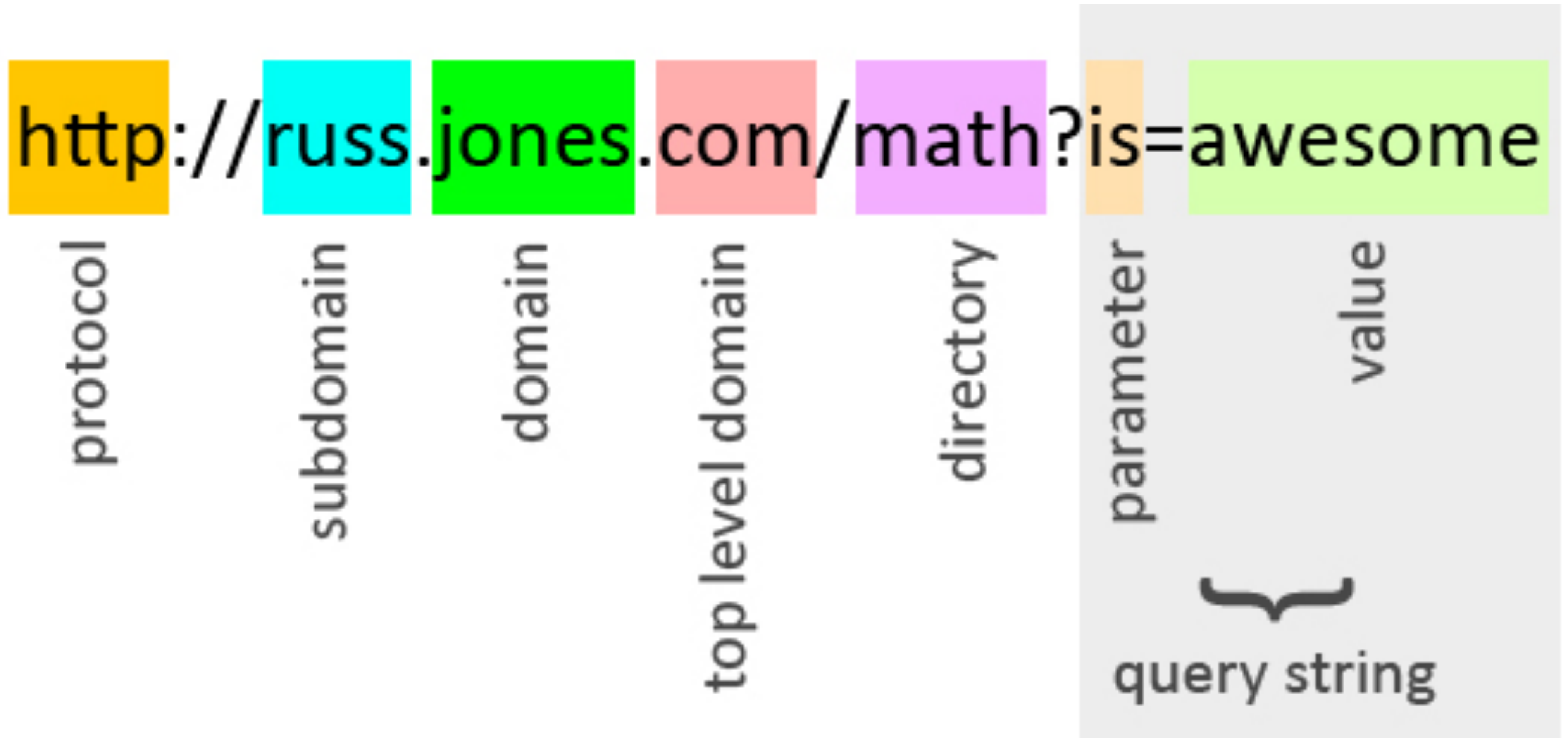
specifies the location of a resource on a network

server responds with **representations** of resources and not the resources themselves

→ Rest lecture

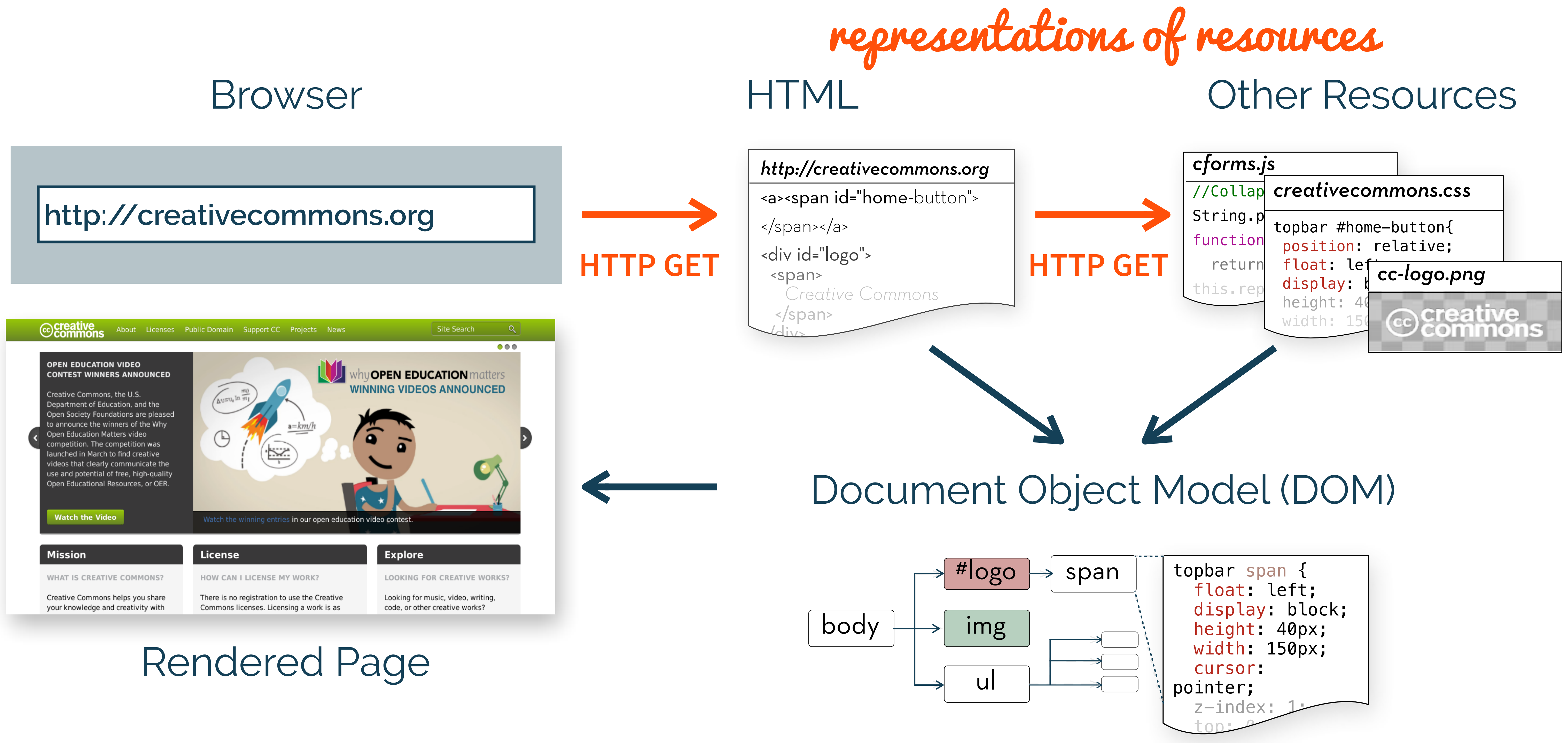
web.archive.org/web/20130116005443/http://tomayko.com/writings/rest-to-my-wife

URL ANATOMY



moz.com/blog/solving-the-subdomain-equation-predicting-traffic-and-value-when-merging-subdomains

LOADING A PAGE IN A BROWSER



```
<a><span id="home-button">
</span></a>
<div id="logo">
  <span>
    Creative Commons
  </span>
</div>
```

```
//Collap
String.p
function
  return
this.rep
```

```
topbar #home-button{
 position: relative;
 float: lef
 display: b
 height: 40
 width: 150
```

```
topbar span {
  float: left;
  display: block;
  height: 40px;
  width: 150px;
  cursor:
pointer;
  z-index: 1;
  top: 0
```

creative commons About Licenses Public Domain Support CC Projects News Site Search

OPEN EDUCATION VIDEO CONTEST WINNERS ANNOUNCED

Creative Commons, the U.S. Department of Education, and the Open Society Foundations are pleased to announce the winners of the Why Open Education Matters video competition. The competition was launched in March to find creative videos that clearly communicate the use and potential of free, high-quality Open Educational Resources, or OER.

Watch the Video

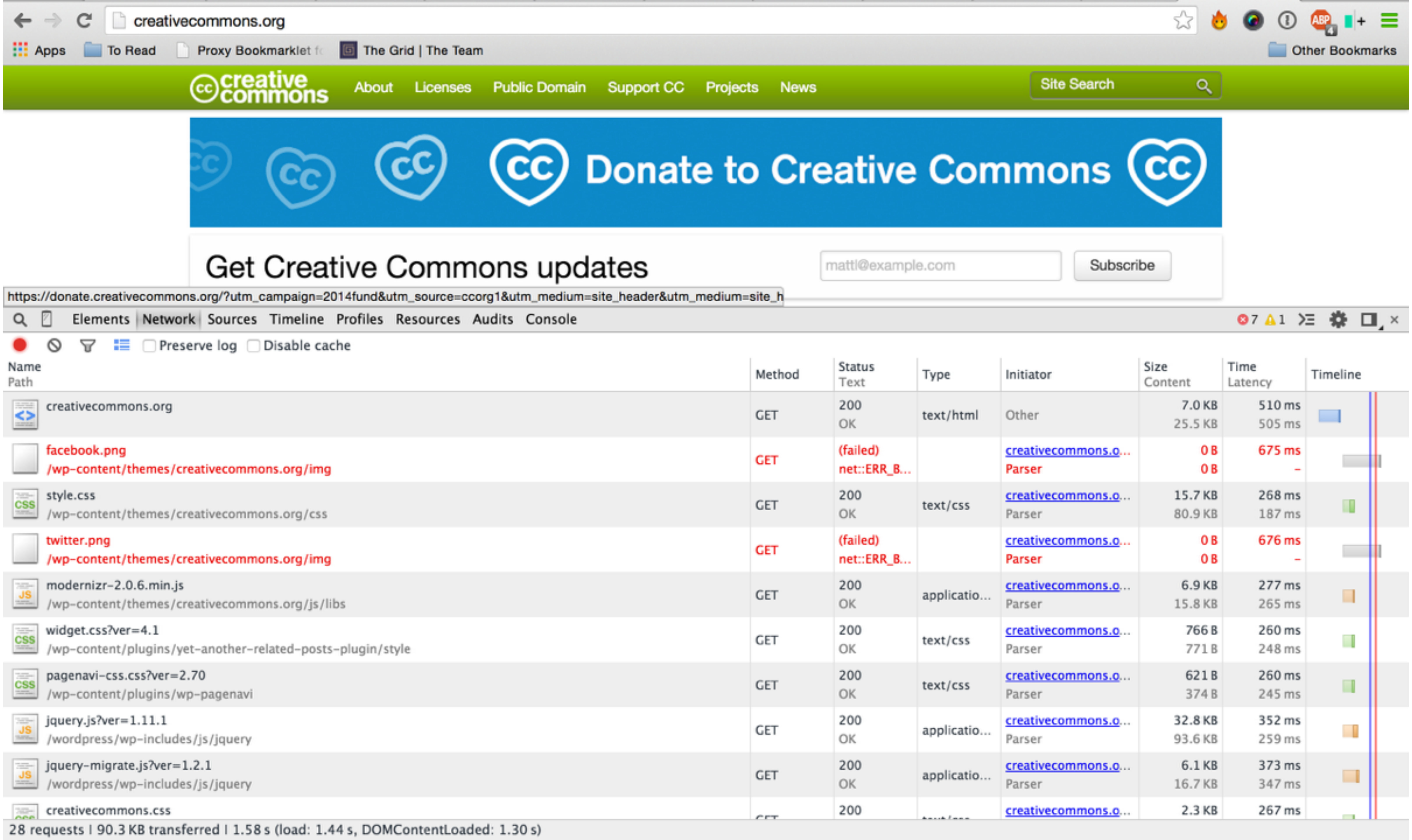
why OPEN EDUCATION matters
WINNING VIDEOS ANNOUNCED

Watch the winning entries in our open education video contest.

Mission
WHAT IS CREATIVE COMMONS?
Creative Commons helps you share your knowledge and creativity with

License
HOW CAN I LICENSE MY WORK?
There is no registration to use the Creative Commons licenses. Licensing a work is as

Explore
LOOKING FOR CREATIVE WORKS?
Looking for music, video, writing, code, or other creative works?



creativecommons.org

Apps | To Read | Proxy Bookmarklet | The Grid | The Team | Other Bookmarks

creative commons | About | Licenses | Public Domain | Support CC | Projects | News | Site Search

Donate to Creative Commons

Get Creative Commons updates

mattl@example.com | Subscribe

https://donate.creativecommons.org/?utm_campaign=2014fund&utm_source=ccorg1&utm_medium=site_header&utm_medium=site_h

Elements | Network | Sources | Timeline | Profiles | Resources | Audits | Console | 7 | 1

Preserve log | Disable cache

Name / Path	Method	Status / Text	Type	Initiator	Size / Content	Time / Latency
creativecommons.org	GET	200 OK	text/html	Other	7.0 KB / 25.5 KB	510 ms / 505 ms
facebook.png /wp-content/themes/creativecommons.org/img	GET	(failed) net::ERR_B...		creativecommons.o... Parser	0 B / 0 B	675 ms / –
style.css /wp-content/themes/creativecommons.org/css	GET	200 OK	text/css	creativecommons.o... Parser	15.7 KB / 80.9 KB	268 ms / 187 ms
twitter.png /wp-content/themes/creativecommons.org/img	GET	(failed) net::ERR_B...		creativecommons.o... Parser	0 B / 0 B	676 ms / –
modernizr-2.0.6.min.js /wp-content/themes/creativecommons.org/js/libs	GET	200 OK	applicatio...	creativecommons.o... Parser	6.9 KB / 15.8 KB	277 ms / 265 ms
widget.css?ver=4.1 /wp-content/plugins/yet-another-related-posts-plugin/style	GET	200 OK	text/css	creativecommons.o... Parser	766 B / 771 B	260 ms / 248 ms
pagenavi-css.css?ver=2.70 /wp-content/plugins/wp-pagenavi	GET	200 OK	text/css	creativecommons.o... Parser	621 B / 374 B	260 ms / 245 ms
jquery.js?ver=1.11.1 /wordpress/wp-includes/js/jquery	GET	200 OK	applicatio...	creativecommons.o... Parser	32.8 KB / 93.6 KB	352 ms / 259 ms
jquery-migrate.js?ver=1.2.1 /wordpress/wp-includes/js/jquery	GET	200 OK	applicatio...	creativecommons.o... Parser	6.1 KB / 16.7 KB	373 ms / 347 ms
creativecommons.css		200		creativecommons.o...	2.3 KB	267 ms

28 requests | 90.3 KB transferred | 1.58 s (load: 1.44 s, DOMContentLoaded: 1.30 s)

HTTP GET Request

method url version

```
GET /index.html HTTP/1.1
```

request headers

```
Host: www.example.com

User-Agent: Mozilla/5.0

Accept: text/xml,application/
xml,application/xhtml+xml,text/html*/*

Accept-Language: en-us

Accept-Charset: ISO-8859-1,utf-8

Connection: keep-alive

<blank line>
```

en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Example_session

HTTP GET Response

version status code text explanation

```
HTTP/1.1 200 OK
```

response headers

```
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Content-Type: text/html; charset=UTF-8
Content-Length: 131
```

entity-body/body

```
<!DOCTYPE html>
<html>
…
</html>
```

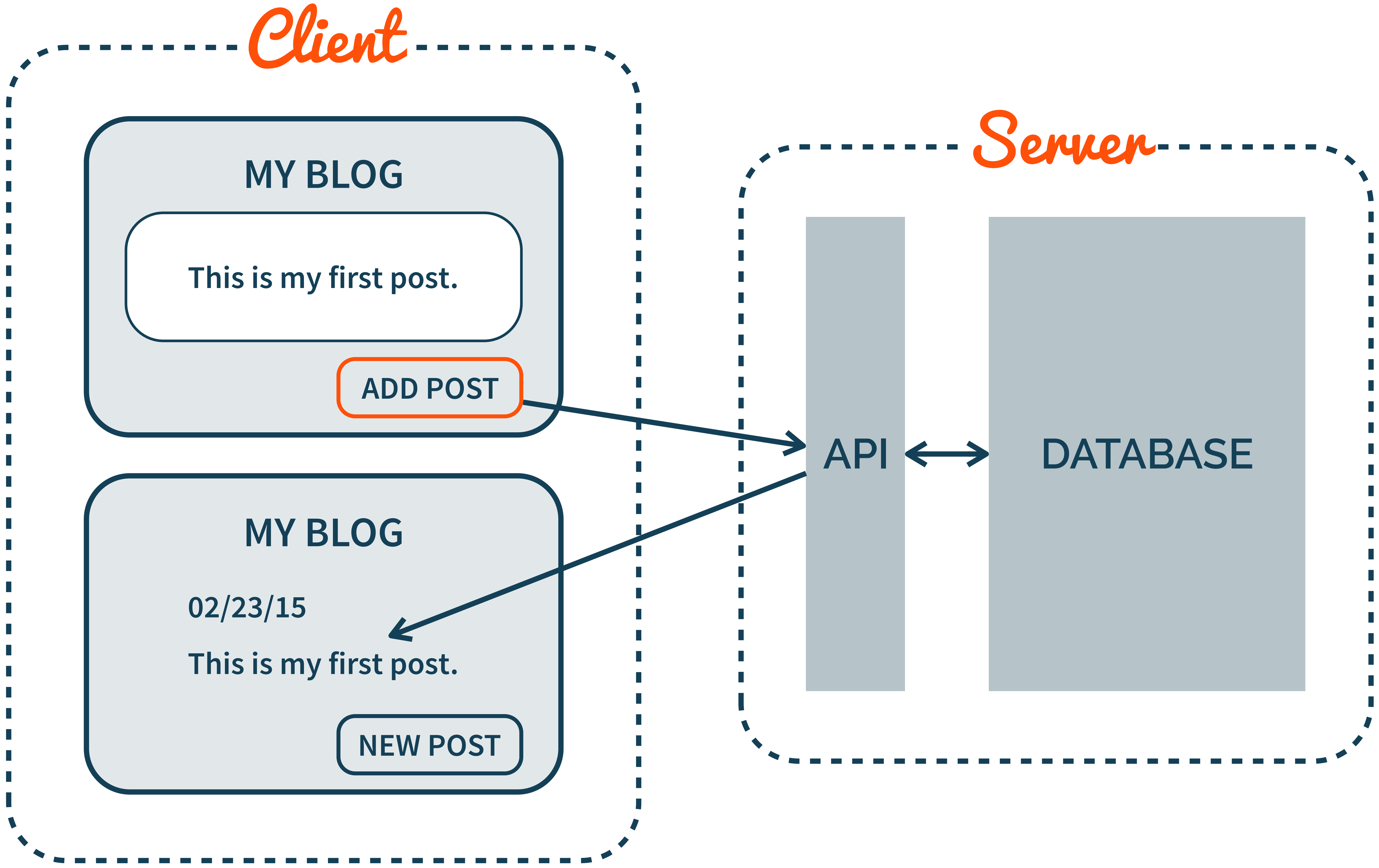
en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Example_session

HTTP GET Response

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Content-Type: text/html; charset=UTF-8
Content-Length: 131

<!DOCTYPE html>
<html>
…
</html>
```

MIME Type

en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Example_session

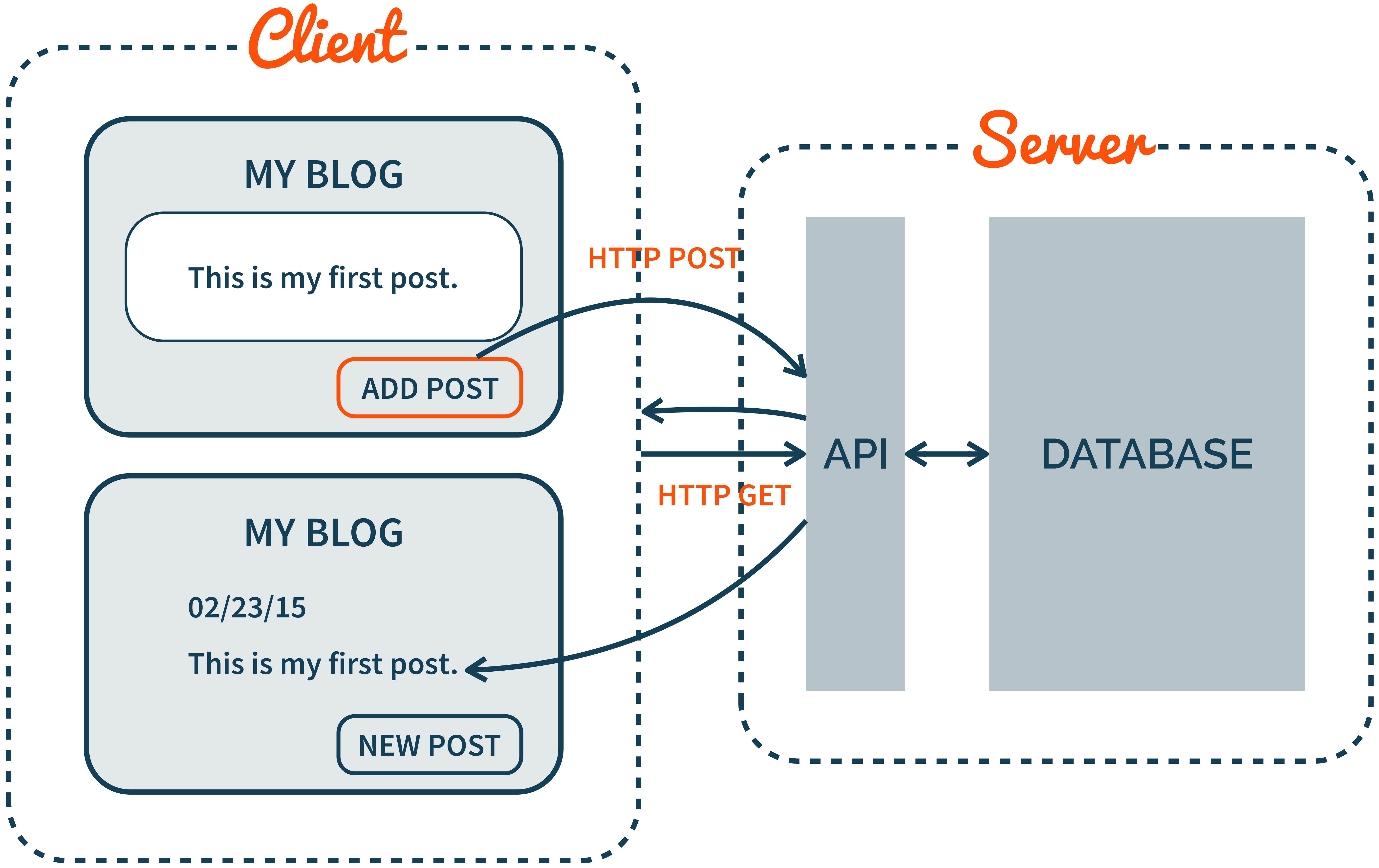

Client
MY BLOG
This is my first post.
ADD POST
MY BLOG
02/23/15
This is my first post.
NEW POST
Server
API
DATABASE

HTTP POST Request

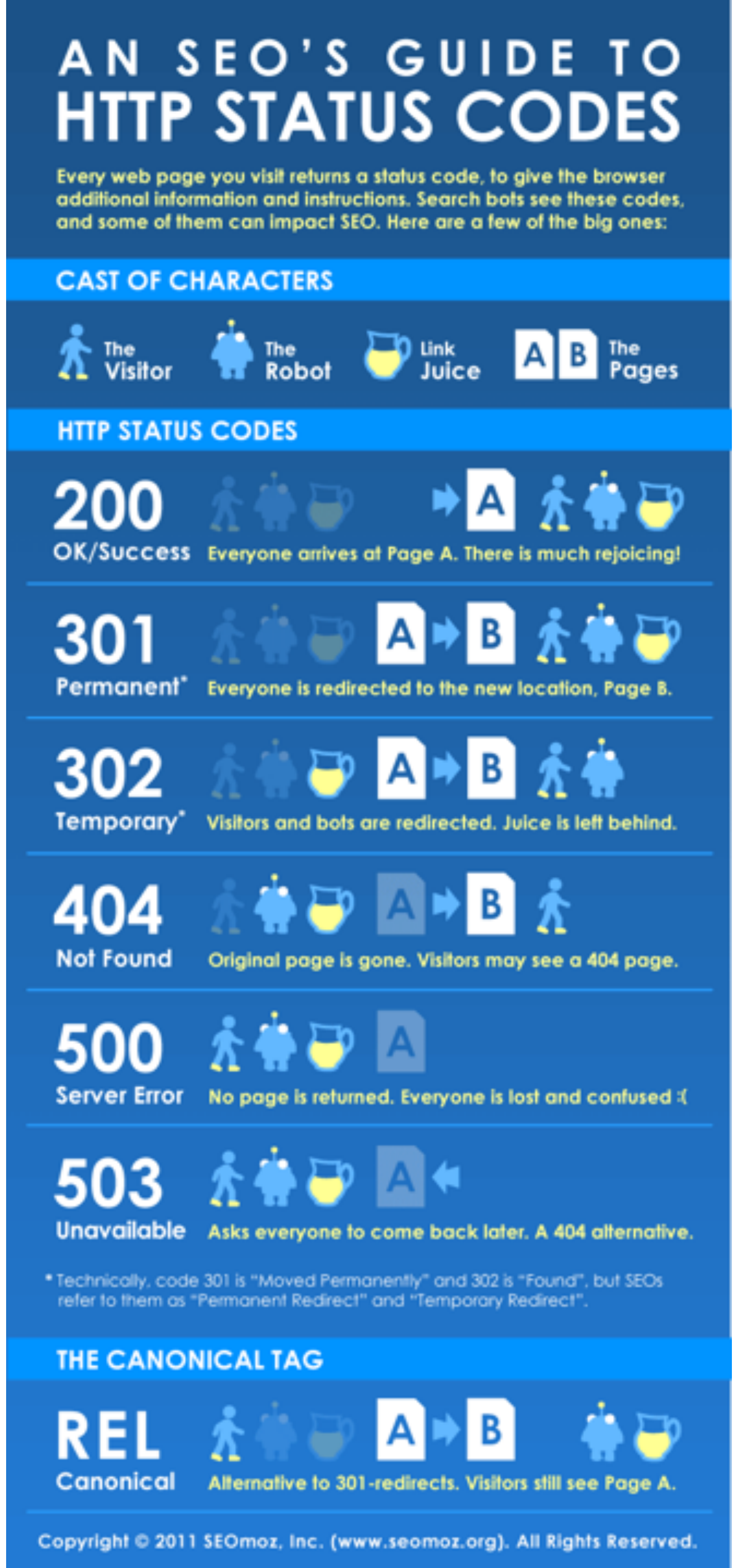
```
POST /messages HTTP/1.1

Host: www.anotherblogpost.com

Content-type: application/x-
www-form-urlencoded

<blank line>

entity-body
```

HTTP POST Response

```
HTTP/1.1 303 See Other

Content-type: text/html

Location: http://
www.anotherblogpost.com/
messages/3486152
```


Client
MY BLOG
This is my first post.
ADD POST
MY BLOG
02/23/15
This is my first post.
NEW POST
HTTP POST
HTTP GET
Server
API
DATABASE

GET	*vs*	POST
retrieve representations of resources		upload data from the browser to server
no side effects		returns information from the server
no data in request body		side effects are likely
		data contained in request body


AN SEO'S GUIDE TO HTTP STATUS CODES

Every web page you visit returns a status code, to give the browser additional information and instructions. Search bots see these codes, and some of them can impact SEO. Here are a few of the big ones:

CAST OF CHARACTERS

The Visitor · The Robot · Link Juice · The Pages

HTTP STATUS CODES

200 OK/Success	Everyone arrives at Page A. There is much rejoicing!
301 Permanent*	Everyone is redirected to the new location, Page B.
302 Temporary*	Visitors and bots are redirected. Juice is left behind.
404 Not Found	Original page is gone. Visitors may see a 404 page.
500 Server Error	No page is returned. Everyone is lost and confused :(
503 Unavailable	Asks everyone to come back later. A 404 alternative.

* Technically, code 301 is "Moved Permanently" and 302 is "Found", but SEOs refer to them as "Permanent Redirect" and "Temporary Redirect".

THE CANONICAL TAG

REL Canonical	Alternative to 301-redirects. Visitors still see Page A.

HTTP STATUS CODES

moz.com/learn/seo/http-status-codes

HTTPS

More details in Security lecture

request and response messages are transmitted securely using encryption

Ajax

AJAX

`Asynchronous JavaScript and XML`

send and receive data without reloading page

Before, every user interaction required the complete page to be reloaded

AJAX

Issue HTTP request to the server from Javascript

Process response with Javascript in the browser

AJAX TECHNOLOGIES

HTML and CSS

DOM

XML

`XMLHttpRequest` object

JavaScript

JSON

AJAX doesn’t require XML

JSON has become de facto standard data interchange format

lightweight and simple format

types: Number, String, Boolean, Array, Object, **`null`**

objects are key/value pairs

JSON CODE EXAMPLE

```
{
  “camelids”: [
    {
      “name”: “llama”,
      “height”: 1.8
    },
    {
      “name”: “alpaca”,
      “height”: 0.9
    }
  ]
}
```

Look familiar?

XMLHttpRequest

```
var xhr = new XMLHttpRequest();
xhr.onreadystatechange = xhrHandler;
xhr.open('get', 'llama.json');
xhr.send(null);
```

http://www.w3.org/TR/XMLHttpRequest/

XMLHttpRequest

```
function xhrHandler() {
  if (xhr.readyState == 4
      && xhr.status == 200) {
    var data = JSON.parse(xhr.responseText);
    myFunction(data);
  }
};
```

CODEPEN

AJAX CHALLENGES

hard to go back to a particular state

URL fragment identifier

content retrieved by AJAX not easily indexable

The same origin policy prevents some Ajax techniques from being used across domains

JSONP

callback-style programming is hard to maintain/test

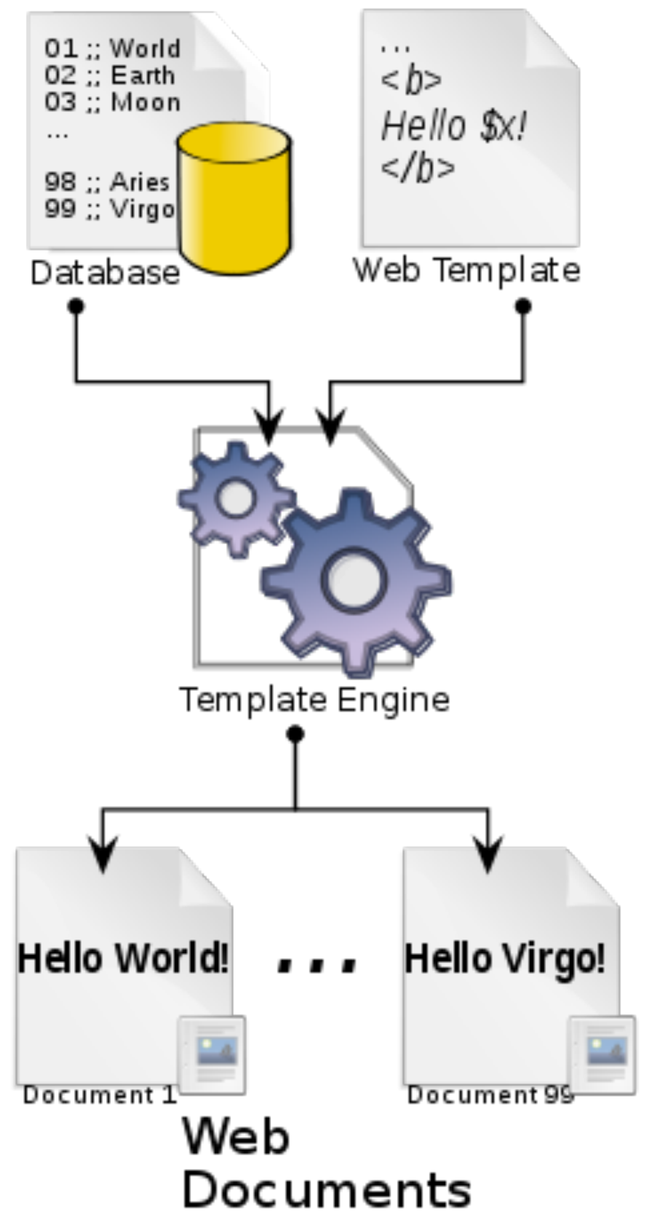
Client-side Templating

TEMPLATES

common way to generate dynamic HTML for multi-page web sites and apps

separation of markup and data (content)

SERVER-SIDE TEMPLATES



server puts HTML and data together and sends it to the browser

platforms like Rails, PHP, JSP

http://www.w3.org/TR/XMLHttpRequest/

CLIENT-SIDE TEMPLATES

React

browser receives HTML and data and puts it together

server serves templates and data required by the templates

made popular by AJAX

Resources - Nouns

RESOURCES

Nouns

If your users might “want to create a hypertext link to it, make or refute assertions about it, retrieve or cache a representation of it, include all or part of it by reference into another representation, annotate it, or perform other operations on it”, make it a resource

can be anything: a document, a row in a database, the result of running an algorithm, etc.

www.w3.org/TR/2004/REC-webarch-20041215/

REPRESENTATION OF RESOURCES

when a client issues a GET request for a resource, server responds with **representations** of resources and not the resources themselves

any machine-readable document containing any information about a resource

server may send data from its database as HTML, XML, JSON, etc.

web.archive.org/web/20130116005443/http://tomayko.com/writings/rest-to-my-wife

REST

`Representational State Transfer`

architectural style, set of design constraints

coined in Roy T. Fielding's dissertation (2000)

the Web is the largest implementation

three important technologies: HTTP, URL, HTML

http://shop.oreilly.com/product/0636920028468.do

www.w3.org/TR/2004/REC-webarch-20041215/

REPRESENTATIONAL STATE TRANSFER

representations are transferred back and forth from client and server

server sends a representation describing the state of a resource

client sends a representation describing the state it would like the resource to have

MULTIPLE REPRESENTATIONS

a resource can have more than one representation: different languages, different formats (HTML, XML, JSON)

client can distinguish between representations based on the value of Content-Type (HTTP header)

A resource can have multiple representations—one URL for every representation

Http Methods - Verbs

Verbs

GET Get a representation of resource

DELETE Destroy resource

POST Create a new resource based on the given representation

PUT Replace resource state with the one described in the given representation

HEAD Get the headers that would be sent with a representation, but not the representation itself

OPTIONS Discover which HTTP methods this resource responds to

PATCH Modify part of the state of this resource based on the given representation

GET

retrieve representations of resources

no side effects: not intended to change any resource state

no data in request body

response codes: 200 (OK), 302 (Moved Permanently), 404 (Not Found)

safe method

DELETE

destroy a resource on the server

success response codes: 200 (OK), 204 (No Content), 202 (Accepted)

not safe, but idempotent

POST

upload data from the browser to server

usually means “create a new resource,” but can be used to convey *any* kind of change: PUT, DELETE, etc.

side effects are likely

data contained in request body

success response codes: 201 (Created), **`Location`** header contains URL for created resource; 202 (Accepted), new resource will be created in the future

Not safe or idempotent

PUT

request to modify resource state

success response codes: 200 (OK), 204 (No Content)

can also be used like POST

idempotent

PATCH

representations can be big: PUTs can be inefficient

send the server the parts of the document you want to change

neither safe nor idempotent

Rest Constraints

CLIENT-SERVER

separation between clients from servers

servers and clients be replaced and developed independently as long as the interface between them is not altered

en.wikipedia.org/wiki/Representational_state_transfer#Stateless

STATELESSNESS

server doesn’t know about client’s application state

client has no direct control over resource state

pass representations around to change state

UNIFORM INTERFACE

Identification of resources

manipulation of resources through these representations

self-descriptive messages

hypermedia as the engine of application state (HATEOAS)

OTHER CONSTRAINTS

cacheable

layered system

code-on-demand (optional)

NEXT CLASS: APIs

courses.engr.illinois.edu/cs498rk1/