

DRAFT

REST API Design

HTTP & REST Principles

Hany ahmed
namozag.com/hany

Outline

HTTP

HTTP Features

Web services

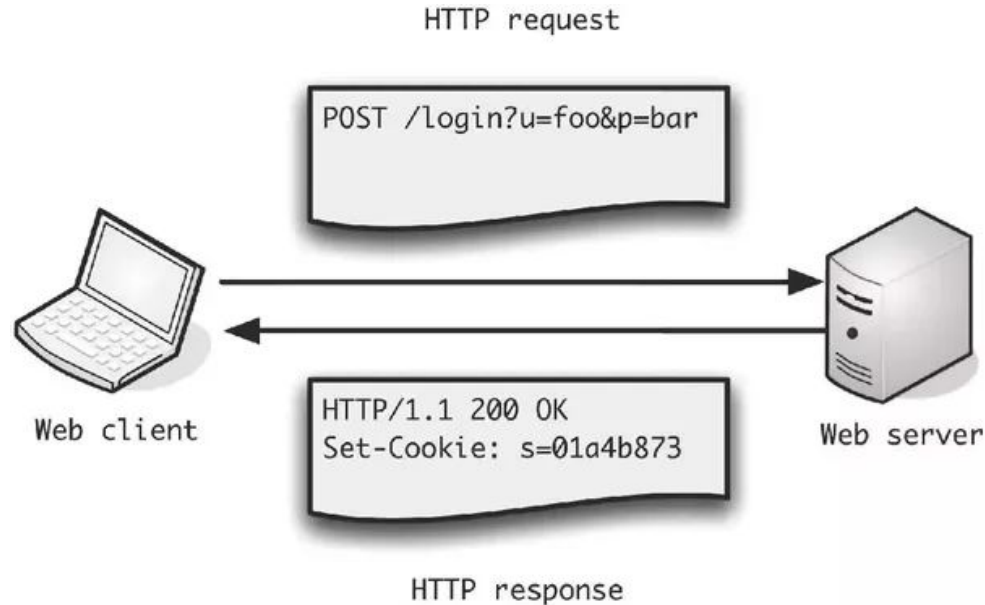
REST

Demos

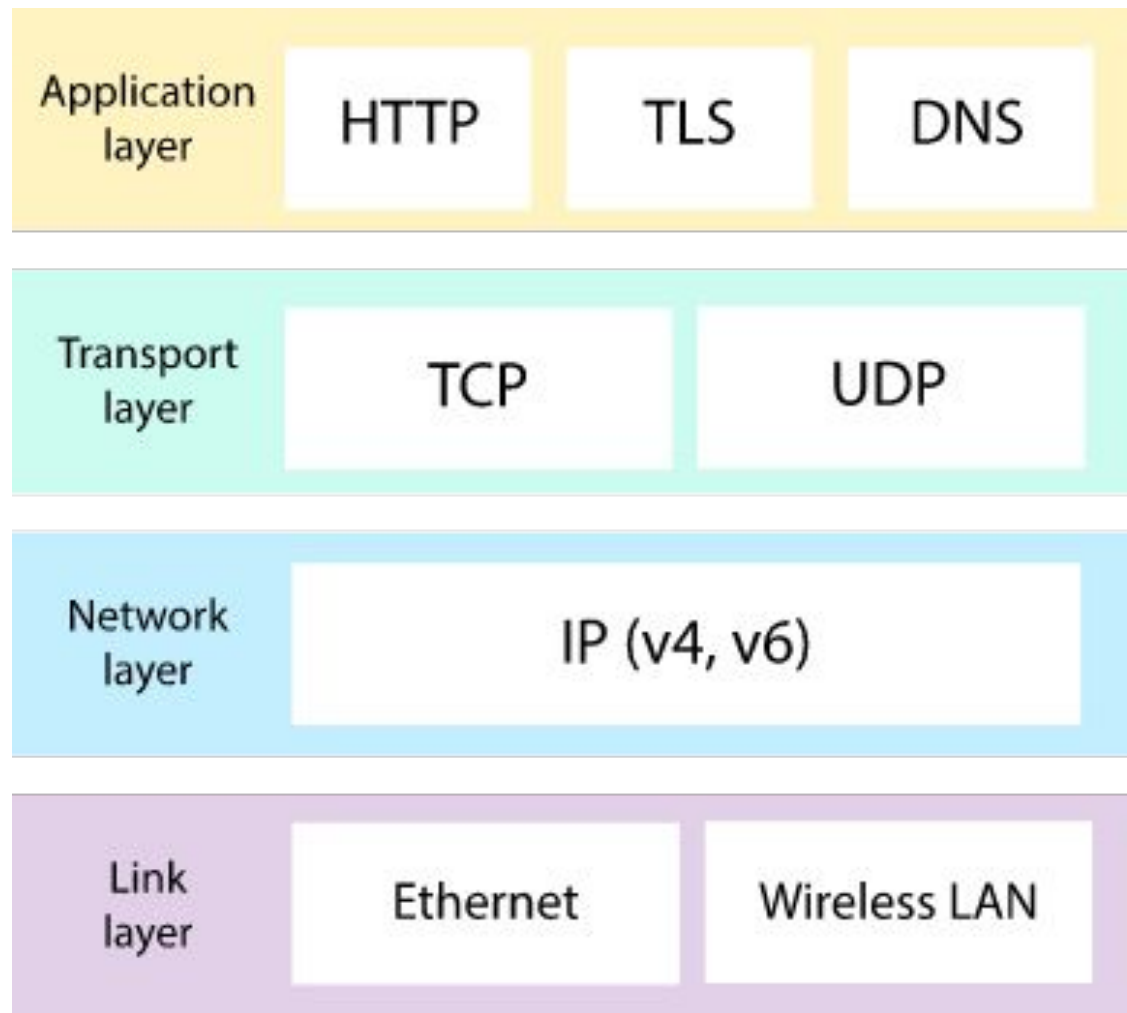
HTTP Protocol

HTTP

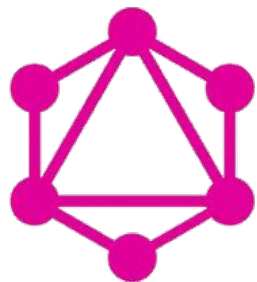
Application-level Communication protocol



Internet protocols



Where is HTTP?

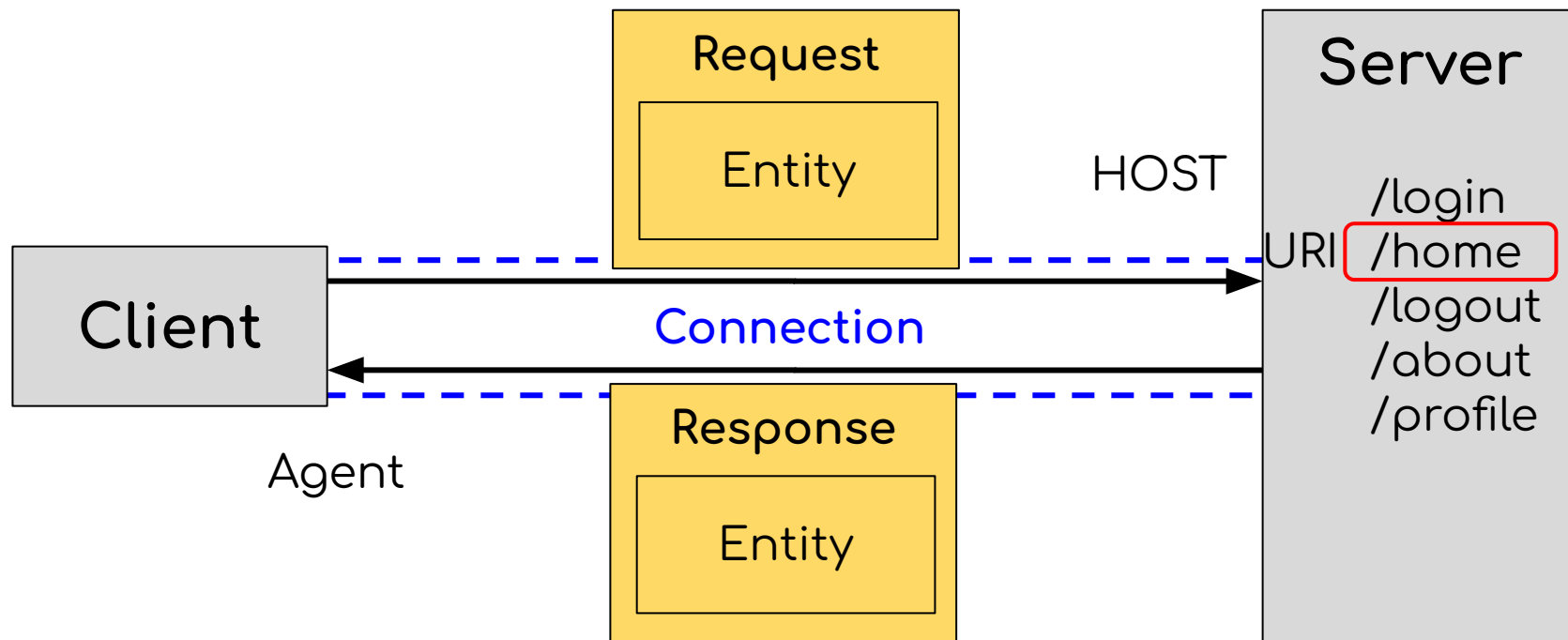


GraphQL



gRPC

HTTP Terminology



HTTP

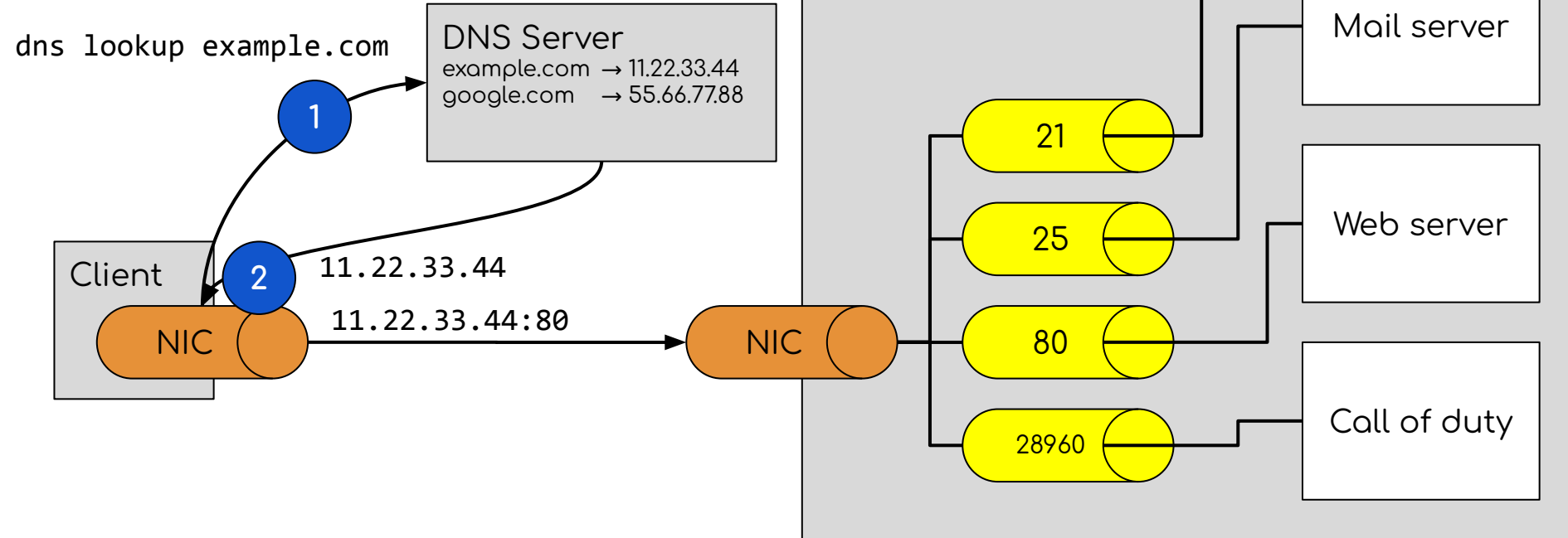
Stateless

Half-duplex

Client-driven

Verbose

Client to Server Communication



URL



HTTP Request

```
GET /httpgallery/introduction/  
HTTP/1.1  
Accept: */*  
Accept-Language: en-gb  
Accept-Encoding: gzip, deflate  
User-Agent: Mozilla/5.0 (Windows NT  
6.3; WOW64; Trident/7.0; rv:11.0)  
like Gecko  
Host: www.httpwatch.com  
Connection: Keep-Alive
```

HTTP Response

```
HTTP/1.1 200 OK  
Server: Microsoft-IIS/8.0  
Date: Mon, 04 Jan 2015 12:04:43 GMT  
X-Powered-By: ASP.NET  
X-AspNet-Version: 4.0.30319  
Cache-Control: no-cache, no-store  
Expires: -1  
Content-Type: text/html;  
charset=utf-8  
Content-Length: 14990
```

```
<!DOCTYPE html> <html>...
```

Status code

Explains the HTTP response status

1xx - Informational

2xx - Successful

3xx - Redirection

4xx - Client Error

5xx - Server Error

2xx - Successful

200 OK

201 Created

202 Accepted

204 No content

3xx - Redirection

301 Moved permanently

302 Moved temporarily

303 Not modified → read from local cache

4xx - Client Error

400 Bad request

401 Unauthorized

403 Forbidden

404 Resource not found

405 Method not allowed

415 Unsupported media type

5xx - Server Error

500 Internal server error

503 Service Unavailable

HTTP Methods

GET

HEAD

POST

DELETE

PUT

PATCH

OPTIONS

—

Safe vs. Idempotent

Safe	return a	(Cacheable)
Idempotent	a=1	
	[a, b, c] - c	
Not safe/idempotent	a++	

Idempotence



☐ Label

☒ Label

My name

Register



☒ Option A

☐ Option B

My name

Update

Safe

Read only

Doesn't change the representation of the resource

Idempotence

Making multiple identical requests has the same effect as making a single request

It's about the effect produced on the representation of the resource and not the response status code received

May lead to side effects

GET

Safe✓

Idempotent✓

Cacheable✓

HTML✓

URL

GET /products/2105

200 OK

Headers

Content-Type: application/json

Body

```
{  
  id: 2105,  
  name: "Lenovo thinkpad",  
  price: 8500  
}
```

The same for **HEAD**, **OPTIONS**

POST

Safe ❌

Idempotent ❌

Cacheable ❌

HTML ✅

POST /products

201 CREATED

Content-Type: application/json

Location: /products/11

```
{  
  name: "X Box",  
  price: 5200  
}
```

POSTing the same request again would result on /products/12

PUT

Safe ✗

Idempotent ✓

Cacheable ✗

HTML ✗

URL

PUT /products/4811

204 No Content

Headers

Content-Type: application/json

Body

```
{  
  name: "X Box special",  
  price: 5000  
}
```

* May result in 201 Created, 200 OK if response has body

PUT vs. PATCH

PUT

=Replace

Affects the whole
entity

PATCH

=Update

Affects only some
fields

PATCH

Safe ~~x~~

Idempotent!

Cacheable ~~x~~

HTML ~~x~~

URL

PATCH /products/4811

200 OK

Headers

Content-Type: application/json

Location: /products/4811

Body

```
[  
  { "op": "replace", "path": "/price", "value": 1500 },  
  { "op": "add", "path": "/tags", "value": ["sale"] }  
]
```

PATCH may have different way of implementation

* May return 204

DELETE

Safe ~~x~~

Idempotent ✓

Cacheable ~~x~~

HTML ~~x~~

DELETE /products/4811

204 No Content

* May result in 200 OK if response has body, 404 when not found

HTTP Headers

Nodes send them to each other to know more about each other and the sent messages

Request headers

User-Agent

Accept: */*

Accept-Language:
en-gb

Accept-Encoding: gzip,
deflate

Host: www.abc.com

Response Headers

Cache-Control:
no-cache

Content-Length: 2748

Content-Type:
image/gif

Date: Wed, 4 Oct 2004
12:00:00 GMT

HTTP Capabilities

HTTP Capabilities

Authentication

Caching

Conditionals

Content negotiation

CORS

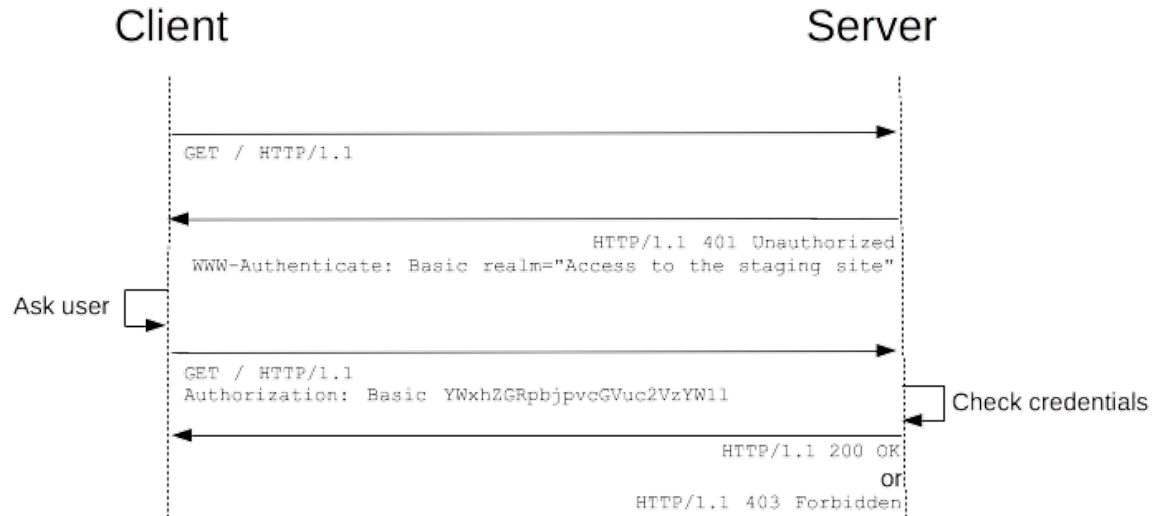
Redirection

...

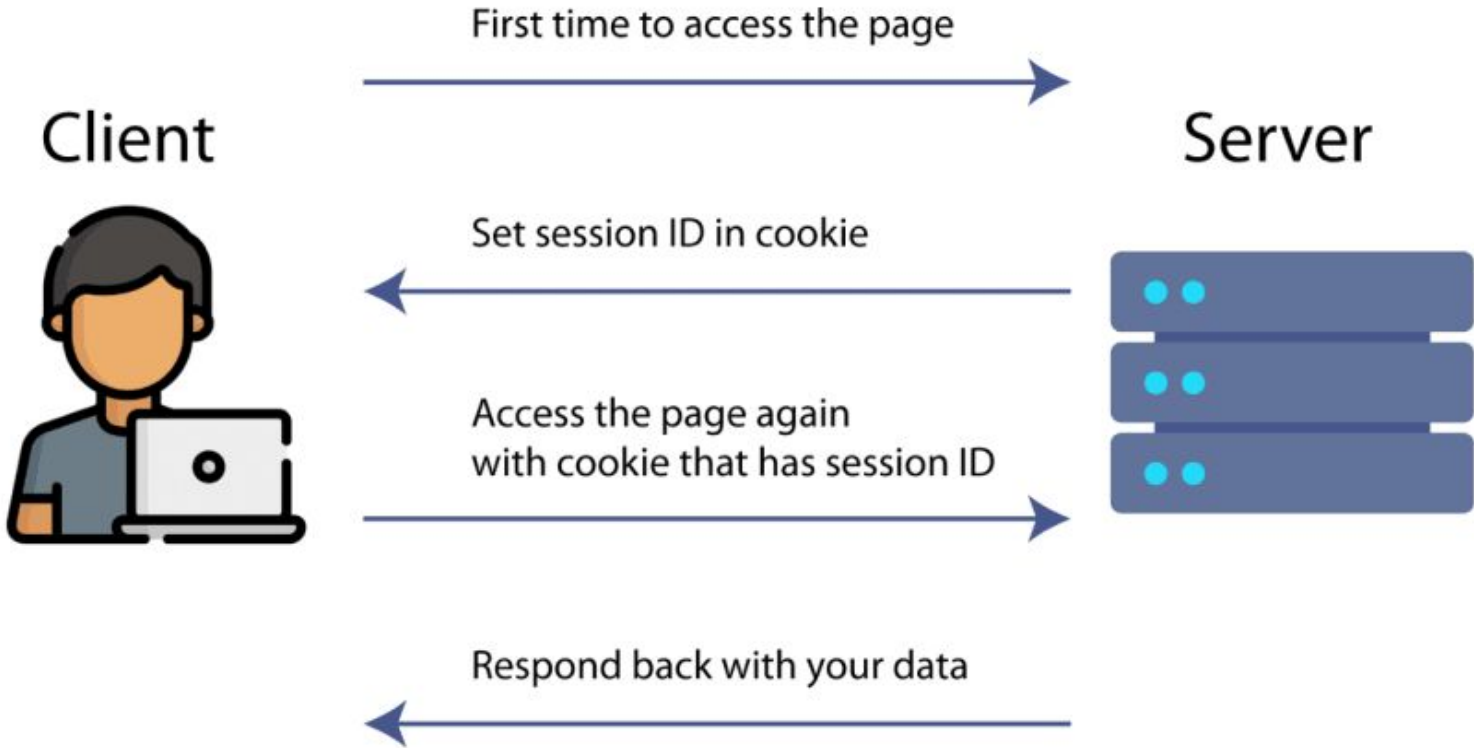
Authentication

WWW-Authenticate

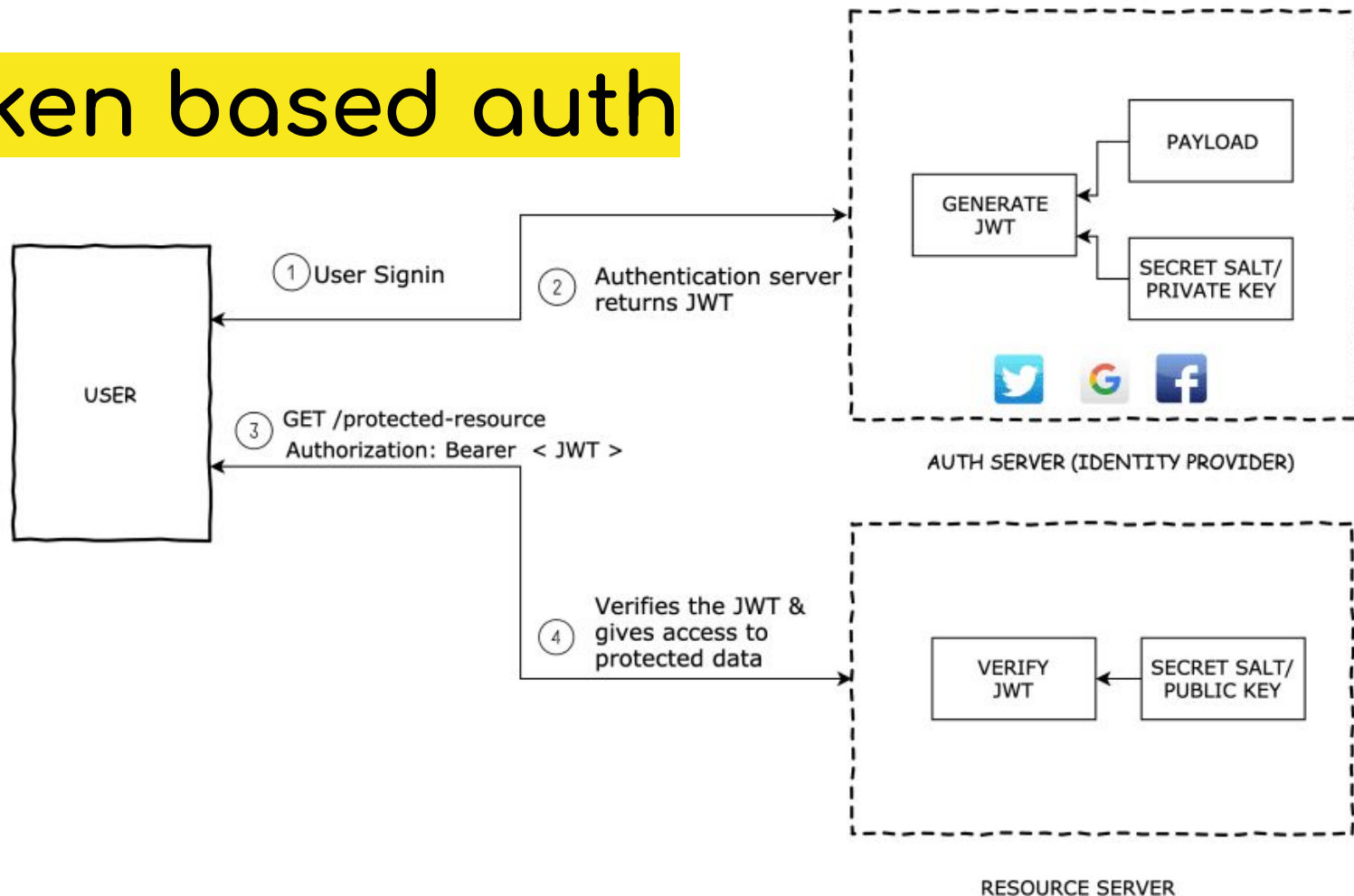
Authorization



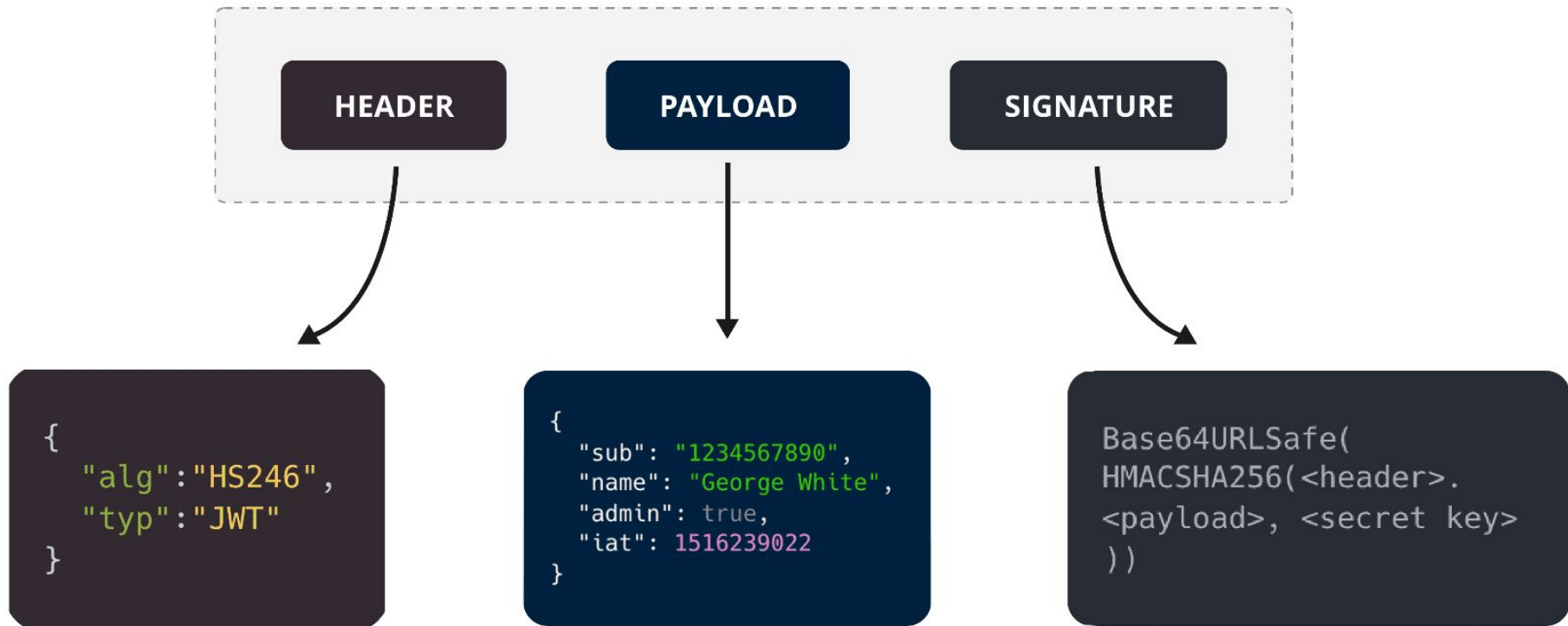
Session based auth



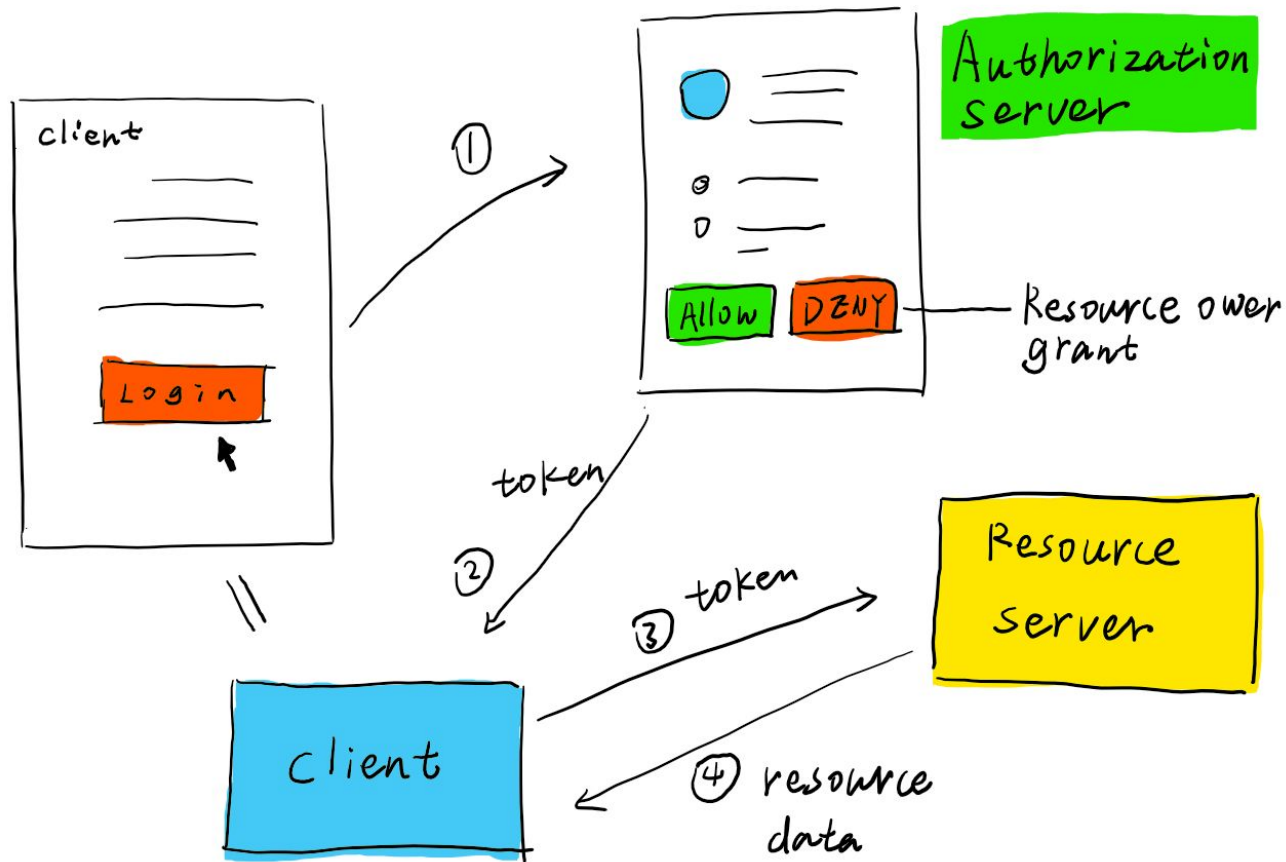
Token based auth



JWT



OAuth



Cookies

Set-Cookie

Cookie



Caching



URL	Response
https://example.com/index.html	<!doctype html>...
https://example.com/style.css	body { ...
https://example.com/script.js	function main () { ...

Cache-Control: max-age=31536000

Cache-Control: no-cache

Expires: Tue, 28 Feb 2022 22:22:22 GMT

Conditionals

Last-Modified

If-Modified-Since

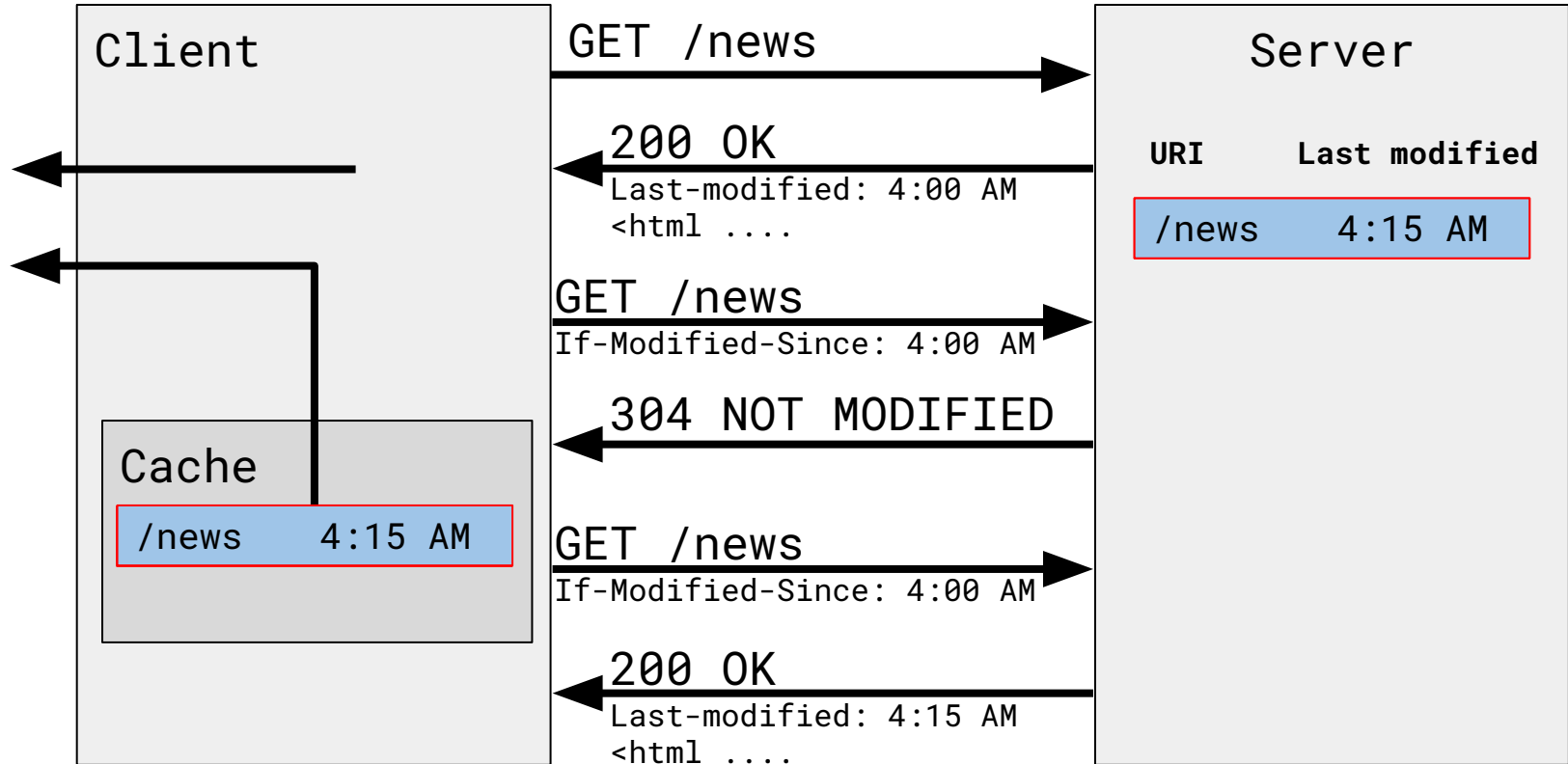
If-Unmodified-Since

ETag

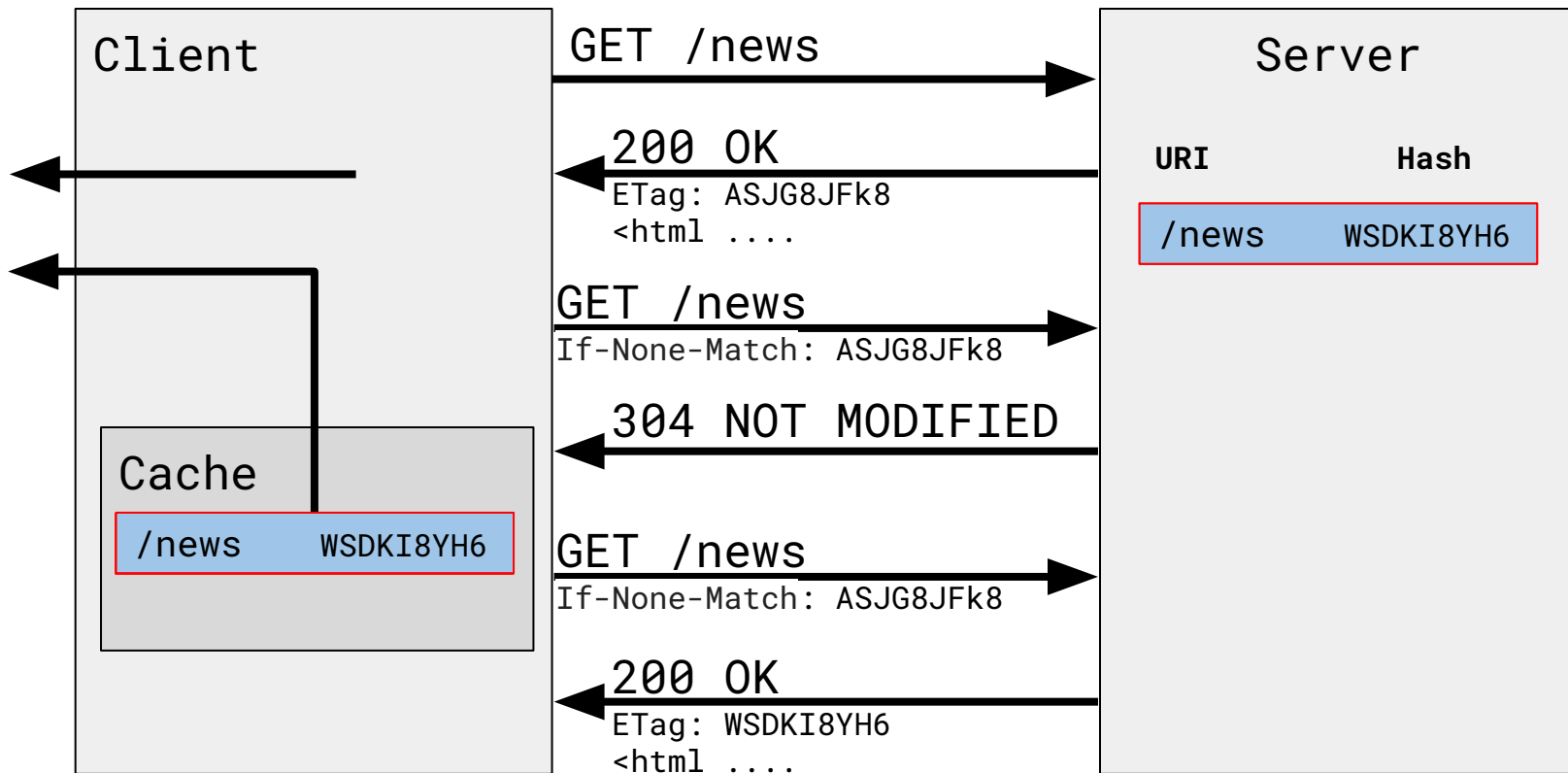
If-Match

If-None-Match

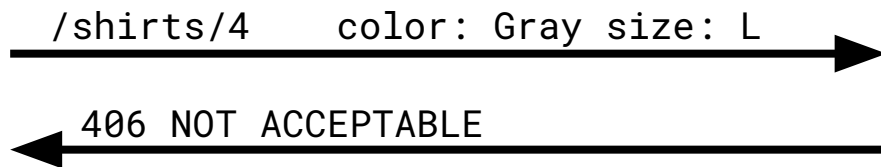
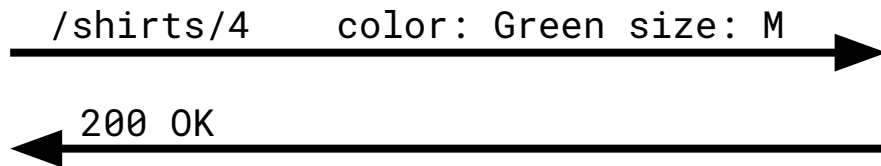
Last-modified



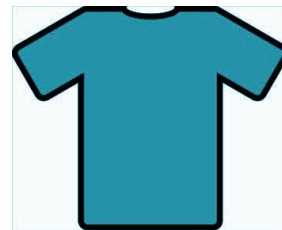
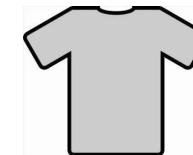
ETag



Content negotiation



Resource representations



Content negotiation headers

Accept

Content-Type

Accept-Charset

Accept-Encoding

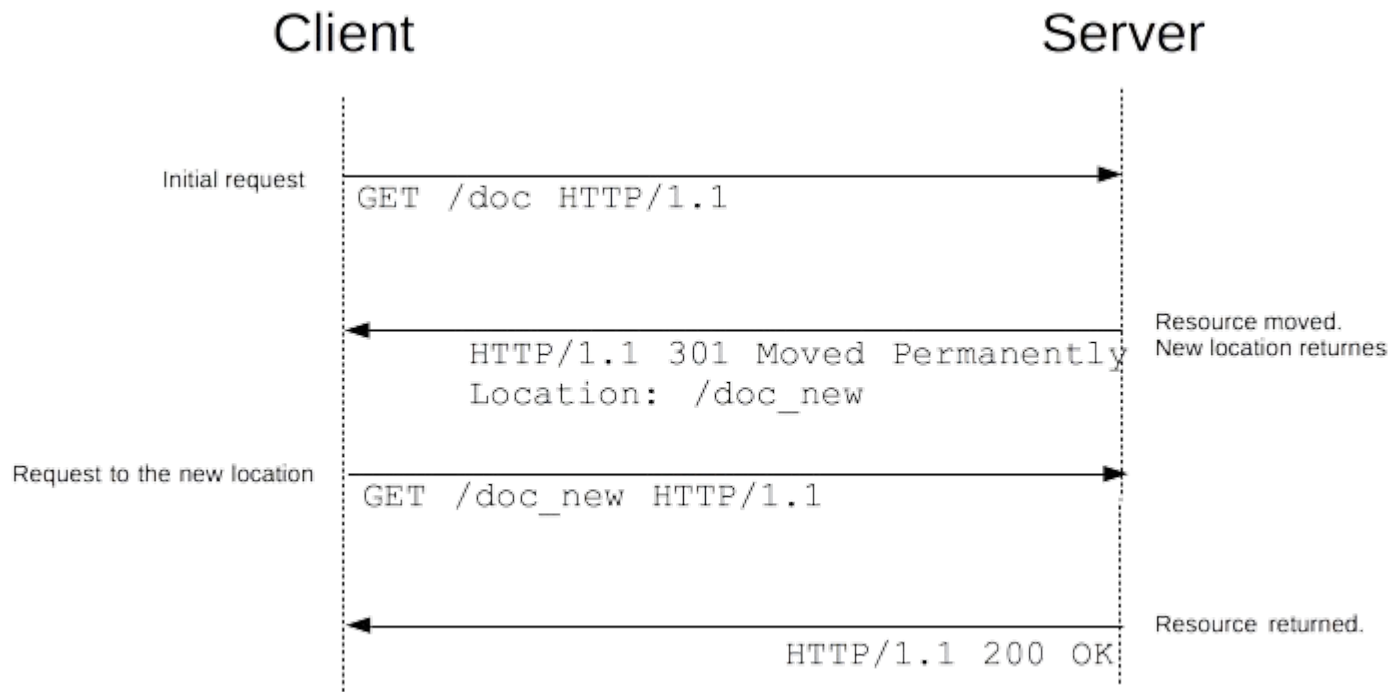
Content-Encoding

Accept-Language

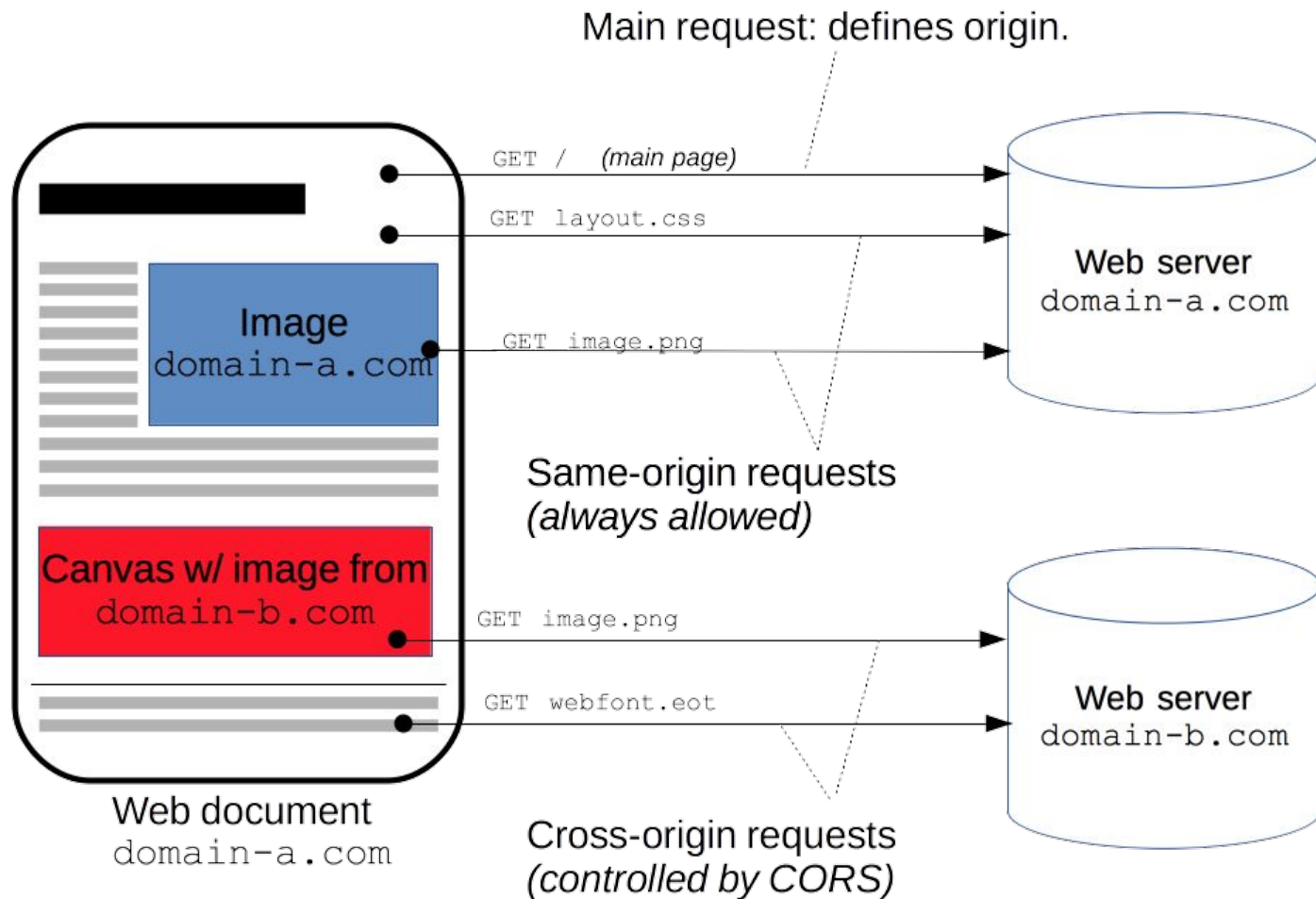
Content-Language

Content-Length

Redirection



CORS



CORS

Origin

Access-Control-Allow-Origin

Access-Control-Allow-Credentials

Access-Control-Allow-Methods

Access-Control-Allow-Headers

CORS Simple requests

Client

Server

Simple request

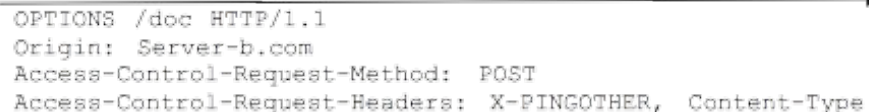
GET /doc HTTP/1.1
Origin: Server-b.com

HTTP/1.1 200 OK

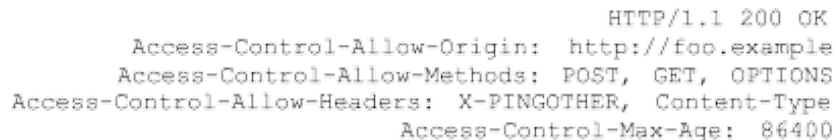
Access-Control-Allow-Origin: *

CORS Preflighted requests

1. Preflight request

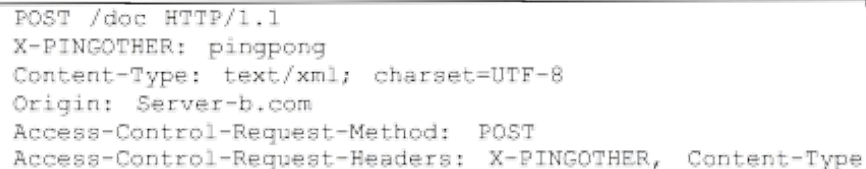


```
OPTIONS /doc HTTP/1.1
Origin: Server-b.com
Access-Control-Request-Method: POST
Access-Control-Request-Headers: X-PINGOTHER, Content-Type
```



```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: http://foo.example
Access-Control-Allow-Methods: POST, GET, OPTIONS
Access-Control-Allow-Headers: X-PINGOTHER, Content-Type
Access-Control-Max-Age: 86400
```

2. Main Request



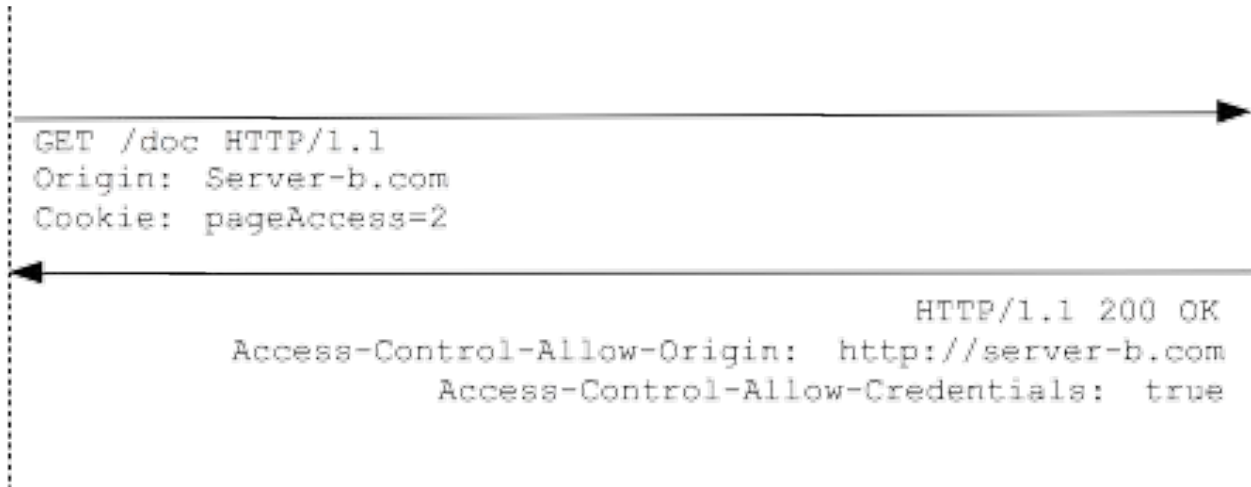
```
POST /doc HTTP/1.1
X-PINGOTHER: pingpong
Content-Type: text/xml; charset=UTF-8
Origin: Server-b.com
Access-Control-Request-Method: POST
Access-Control-Request-Headers: X-PINGOTHER, Content-Type
```



```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: http://foo.example
```

Requests with credentials

Requests that are aware of HTTP cookies and HTTP Authentication information





User-Agent

Referer

Host

Date

Forwarded

X-Forwarded-For

Location

Date

Web services

Web service

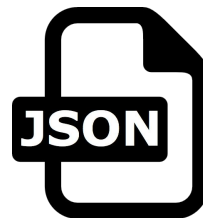
Machine to machine communication over a network.



Data transfer protocol
+
Standard data exchange format
=

Web service

http://



Web service

Loosely coupled

Interoperable

Platform / OS /

Language

independent

SOAP

A protocol.

Message oriented.

Message Envelope is XML.

WSDL as definition language.



Sample SOAP Request

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: nnn

```
<?xml version="1.0"?>
```

```
<soap:Envelope
```

```
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
```

```
  soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
```

```
    <soap:Body xmlns:m="http://www.example.org/stock">
```

```
      <m:GetStockPrice>
```

```
        <m:StockName>IBM</m:StockName>
```

```
      </m:GetStockPrice>
```

```
    </soap:Body>
```

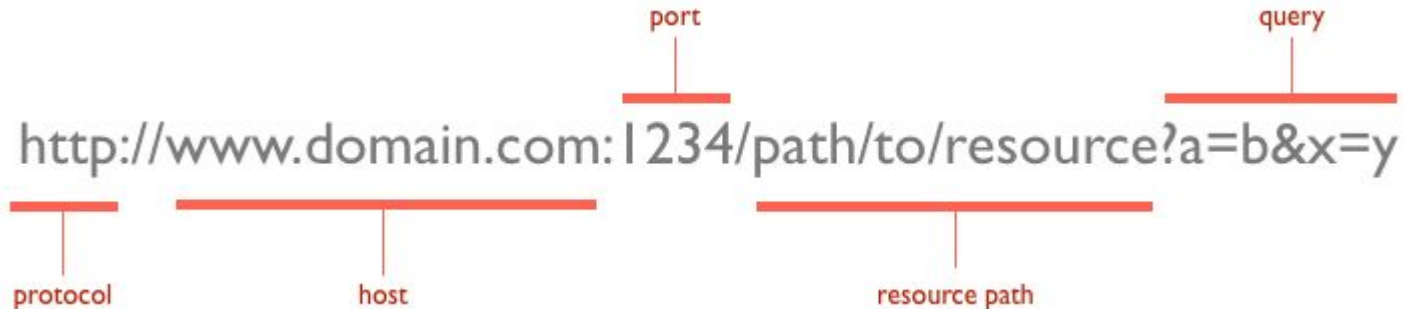
```
</soap:Envelope>
```

REST

Architectural style.

Make use of HTTP protocol features:

URI, Methods, Headers



SOAP

Protocol

HTTP POST only

Message oriented

XML Envelope

WSDL definition

REST

Architectural style

HTTP GET, POST, ...

Resource oriented

No Envelope

OpenApi definition

REST

HTTP Representation

- Entity
 - Payload information of a request or response
- Content negotiation
 - The mechanism for selecting the appropriate representation when servicing a request
- Representation
 - An entity included with a response that is subject to content negotiation

REST Terminology

Resource: an object or representation of something

Collections: set of resources

Media/Content type: the way a resource can be represented

REST Maturity Levels

#0

Single URL

#1

Resources

#2

HTTP verbs

#3

HATEOAS

—

Level 0 Single URL

Single URL for all requests

Much like SOAP

POST /api

```
{ "code": "updateUser", "data": { ... } }
```


Level 1 Resources

Identify your application resources

Resources may be mapped but not limited to Entities

End point

Entity

Table

/users/add

User

user

/products

Product

product

Level 2

HTTP Verbs

GET /users /11 → Retrieve

POST /users → Create

PUT /users/11 → Update (Replace)

PATCH /users/11 → Update (Partial)

DELETE /users/11 → Delete

Data-oriented Approach

SOAP → functions

REST → Domain entities

Standard way to access HTTP resources,
like the standard way you access DB data

Designing representation

Resource Representation

Users can view the representation of a resource in different formats, called media types.

JSON

2 Data structures: Object, Array

Limited data types: Null, Boolean,
Number, String

No standard way to represent date

shorter than XML and faster in parsing

Relations in JSON

```
{  
    "name": "Ahmed",  
    "company": "abc"  
}
```

```
{  
    "name": "Ahmed",  
    "company-link":  
        "http://api.com/companies/abc"  
}
```

Links Example

```
{
  "id": 1,
  "url": "https://api.github.com/repos/octocat/Hello/issues/1347",
  "repository_url": "https://api.github.com/repos/octocat/Hello",
  "comments_url": "https://api.github.com/repos/octocat/Hello/issues/1347/comments"
  "user": {
    "login": "octocat",
    "id": 1,
    "avatar_url": "https://github.com/images/error/octocat_happy.gif",
    "url": "https://api.github.com/users/octocat",
    "repos_url": "https://api.github.com/users/octocat/repos",
    "gists_url": "https://api.github.com/users/octocat/gists{/gist_id}"
  }
}
```


Resource Common properties

Type/Kind

Self link/URL

Navigation links: next, previous, first, last

Envelope

Use it when needed only

May be needed to

- Represent errors

- wrap a paginated response

- not able to use some headers

Designing Resources URLs

Naming convention

Noun

products

Plural (collections)

products/1/buyer

kebab-case:

Lowercase

action-items

hyphen-separated

Query parameters

Filtering

GET /products?state=active&category=tv

Sorting

GET /products?sort=-price,created_at

Paging

GET /products?offset=10&limit=10 //Facebook

?page=2&rpp=10 //Twitter

?offset=10&limit=10 //Linkedin

Subresources

Path parameter

`/users/11/products`

Matrix parameter

`/users;11/products`

`/users;id=11/products`

`/users;name=ali/products`

* Each path segment represents one relationship traversal

Actions

A service that calculate, translate, convert

POST /translate?value=cat&from=en&to=ar 😐

GET /translations?value=cat&from=en&to=ar 😊

POST /products/:id/star / unstar 😐

POST /products/:id/star 😊

DELETE /products/:id/star

Partial response

Get exactly what you need in the response

Linked In

`/people:(id,first-name,last-name,industry)`

Facebook (And Google)

`/joe.smith/friends?fields=id,name,picture`

Idempotency

You can make idempotent calls any number of times without concern that the server creates or completes an action on a resource more than once

You can retry idempotent calls

Idempotency on POST request

On Paypal

To enforce idempotency on REST API POST calls, use the PayPal-Request-Id request header

a unique user-generated ID that the server stores for a period of time.

API Design Guidelines

API First approach

File ▾ Preferences ▾ Generate Server ▾ Generate Client ▾ Help ▾

✓ Processed with no error

```
1 # this is an example of the Uber API
2 # as a demonstration of an API spec in YAML
3 swagger: '2.0'
4 info:
5   title: Uber API
6   description: Move your app forward with the Uber API
7   version: '1.0.0'
8   # the domain of the service
9 host: api.uber.com
10 # array of all schemes that your API supports
11 schemes:
12   - https
13   # will be prefixed to all paths
14 basePath: /v1
15 produces:
16   - application/json
17 paths:
18   /products:
19     get:
20       summary: Product Types
21       description: |
22         The Products endpoint returns information about the "Uber" products
23         offered at a given location. The response includes the display name
24         and other details about each product, and lists the products in the
25         proper display order.
26       parameters:
27         - name: latitude
28           in: query
29           description: Latitude component of location.
30           required: true
31           type: number
32           format: double
33         - name: longitude
34           in: query
35           description: Longitude component of location.
36           required: true
37           type: number
38           format: double
39       tags:
40         - Products
41       responses:
42         200:
43           description: An array of products
44           schema:
```

Uber API

Move your app forward with the Uber API

Version 1.0.0

Filter operations by a tag:

Products Estimates User

Paths

/products

GET /products

Products

Summary

Product Types

Description

The Products endpoint returns information about the Uber products offered at a given location. The response includes the display name and other details about each product, and lists the products in the proper display order.

Parameters

Name	Located in	Description	Required	Schema
latitude	query	Latitude component of location.	Yes	⇔ number (double)
longitude	query	Longitude component of location.	Yes	⇔ number (double)

Responses

Code	Description	Schema
200	An array of products	⇔ { Product { }

Evolving APIs

Two versions running concurrently

Embed version in the url

`/api/v2/trips`

Robustness principle

Tolerate Unrelated Changes (minor changes)



Permalink

use UUID if name is changeable

Real life Examples

GitHub Repository

PayPal Payment

JIRA Issue

FaceBook Photo

YouTube Video

GitHub Repositories resource

GET /user/repos

POST /user/repos

GET /repos/:owner/:repo/contributors

DELETE /repos/:owner/:repo

POST /repos/:owner/:repo/transfer

PetStore Pet Resource

POST

/pet Add a new pet to the store



PUT

/pet Update an existing pet



GET

/pet/findByStatus Finds Pets by status



GET

/pet/{petId} Find pet by ID



POST

/pet/{petId} Updates a pet in the store with form data



DELETE

/pet/{petId} Deletes a pet



POST

/pet/{petId}/uploadImage uploads an image



PetStore User Resource

POST

/user Create user

GET

/user/login Logs user into the system

GET

/user/logout Logs out current logged in user session

GET

/user/{username} Get user by user name

PUT

/user/{username} Updated user

DELETE

/user/{username} Delete user

PayPal Payment Resource

POST /v1/payments/payment

GET /v1/payments/payment

GET /v1/payments/sale/{sale_id}

POST /v1/payments/sale/{sale_id}/refund

JIRA Issue Resource

POST /rest/api/2/issue

GET /rest/api/2/issue/{issueIdOrKey}

PUT /rest/api/2/issue/{issueIdOrKey}

DELETE /rest/api/2/issue/{issueIdOrKey}

PUT /rest/api/2/issue/{issueIdOrKey}/assignee

GET /rest/api/2/issue/{issueIdOrKey}/comment

Facebook Photo Resource

POST /v2.12/{page_id}/photos

GET /v2.12/{photo-id}

GET /v2.12/{photo-id}/likes

GET /v2.12/{object-id}/comments

POST /v2.12/{photo-id}

DELETE /v2.12/{photo_id}

GMail Message Resource

GET /userId/messages

GET /userId/messages/id

POST /userId/messages/id/modify

DELETE /userId/messages/id

POST /userId/messages/send

POST /userId/messages/id/trash

YouTube Video Resource

GET /videos?id=#

POST /videos

DELETE /videos?id=#

PUT /videos?id=#

POST /videos/rate?id=#

Woocommerce Product Resource

GET /v2/products

POST /v2/products

GET /v2/products/<id>

PUT /v2/products/<id>

DELETE /v2/products/<id>

GET /v2/products/<product_id>/reviews

WordPress Post Resource

GET /wp/v2/posts

POST /wp/v2/posts

GET /wp/v2/posts/<id>

POST /wp/v2/posts/<id>

DELETE /wp/v2/posts/<id>

Main references

- Web API Design: The Missing Link, apigee.com
- <https://developer.mozilla.org/kab/docs/Web/HTTP>
- <https://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api>
- https://en.wikipedia.org/wiki/Representational_state_transfer
- <https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design>
- <https://restfulapi.net/>