Web Theory

Overview

Goals

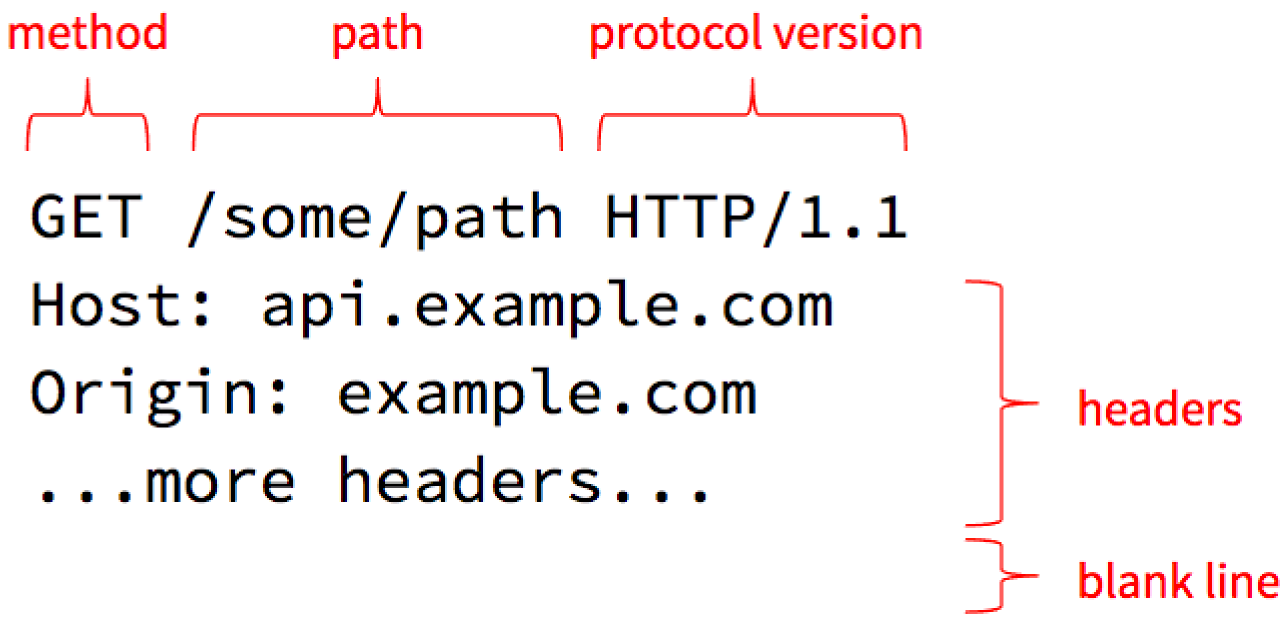
- Be able to explain how the internet works
- Focus on front-end web
- Learn about the purpose of the “semantic web”

How the Web Works

The Big Question

“What happens when I visit `https://google.com`?”

Request is sent



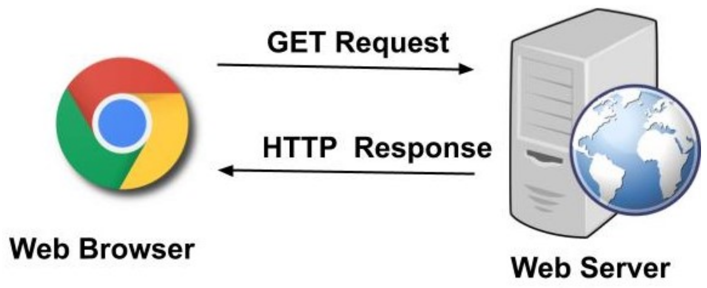
```
GET /some/path HTTP/1.1
Host: api.example.com
Origin: example.com
...more headers...

```

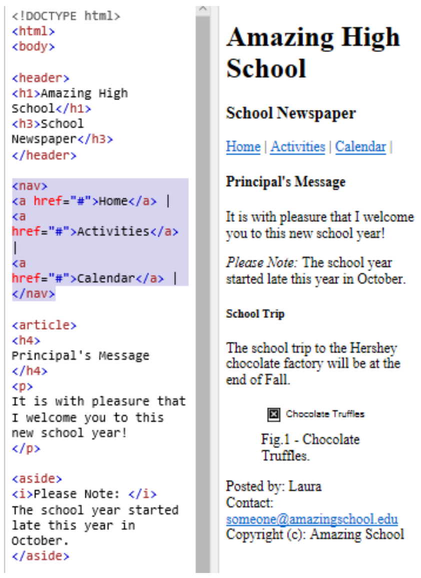
Server receives request

Server produces HTML in response

Response (HTML) arrives



Browser renders HTML



```
<!DOCTYPE html>
<html>
<body>

<header>
<h1>Amazing High
School</h1>
<h3>School
Newspaper</h3>
</header>

<nav>
<a href="#">Home</a> |
<a
href="#">Activities</a>
|
<a
href="#">Calendar</a> |
</nav>

<article>
<h4>
Principal's Message
</h4>
<p>
It is with pleasure that
I welcome you to this
new school year!
</p>

<aside>
<i>Please Note: </i>
The school year started
late this year in
October.
</aside>
```

Amazing High School

School Newspaper

Home | Activities | Calendar |

Principal's Message

It is with pleasure that I welcome you to this new school year!

Please Note: The school year started late this year in October.

School Trip

The school trip to the Hershey chocolate factory will be at the end of Fall.

Chocolate Truffles

Fig.1 - Chocolate Truffles.

Posted by: Laura
Contact:
someone@amazingschool.edu
Copyright (c): Amazing School

Requests and Responses

Types of Requests

- GET: most basic request, ask for data
- POST: **change the world** in some way, some database is likely to be updated
- PUT: create a new resource at the specified place
- DELETE: remove the specified resource
- …and more

Idempotency

- Stateless - each request is **independent** of the next
- If you make the **same request** multiple times, the server is left in the exact same state as if you made it only once
- The request contains everything the server/browser need to know
- Don’t need to know what came before to do the proper thing
- GET, PUT, and DELETE are idempotent
- POST is not

Types of Responses

- HTML: most common, contains “markup” to be rendered
- JS: Javascript code to be executed in the browsser
- CSS: styling modifications to apply on top of HTML structure
- PNG or JPEG: images! separate response for each.
- JSON: key-value data, usually from a database, to be used within Javascript code
- XML: key-value data, less common nowadays, also usually used within Javascript code
- …and more!

HTTP Theory

URLs

URL stands for Uniform Resource Locator

`http://google.com/search?input=cats&filter=False`

Parts of a URL

`http://google.com/search?input=cats&filter=False`

- Protocol `http://`
- Hostname `google.com`
- Resource `/search`
- Query arguments `?input=cats&filter=False`

DNS

- Hostnames are just for humans
    - The real address of a web server is called the **IP Address**
- Before any request is sent, the hostname is converted into an IP address
- `google.com` becomes something like `52.53.158.216`
- The process of converting a hostname to an IP address is called **Domain Name System**, or DNS

HTTP, a Protocol

- HTTP stands for **Hyper Text Transfer Protocol**
- Request/response pattern
- Structure of requests/responses is simple and straightforward
- Established way for browsers and computers to communicate
- Originally invented to allow groups of scientific researchers to share information

HTTPS vs. HTTP

- HTTPS is just HTTP + Secure
- It is 99% the same as HTTP, but with a layer of encryption around all requests
- Browser must **encrypt** request (no longer human readable) in a way that allows the server to **decrypt** it
  - And vice-versa
- HTTPS is the expectation nowadays
  - HTTP on its own is highly insecure and vulnerable to crypto-attacks

Browsers & Accessibility

What can browsers do?

- Read and render HTML
- Send web requests
- Receive web responses
- Run Javascript code
- And so much more!

Multiple Responses

- A single response can contain both HTML AND Javascript code
- Multiple responses can be assembled together to create a single page
- A single page usually involves > 5 request/responses, for HTML, CSS, Javascript, Images, and any other data

Semantic Web

- Also known as “Web 3.0”
- Generally refers to the use of common data formats and exchange protocols that have meaning across applications, communities, and enterprises
- For coders, the use of **Semantic HTML elements** is encouraged
    - Use `<article>` or `<nav>` instead of just `<div>`

Accessibility

- Many users of the web use assistive technologies, such as **screen readers**
- Allow someone to navigate the contents of a webpage via keyboard and audio or braille output
- Assistive technologies only work if the proper HTML elements and attributes are used
    - Machine cannot derive meaning if data is not properly labelled

The End