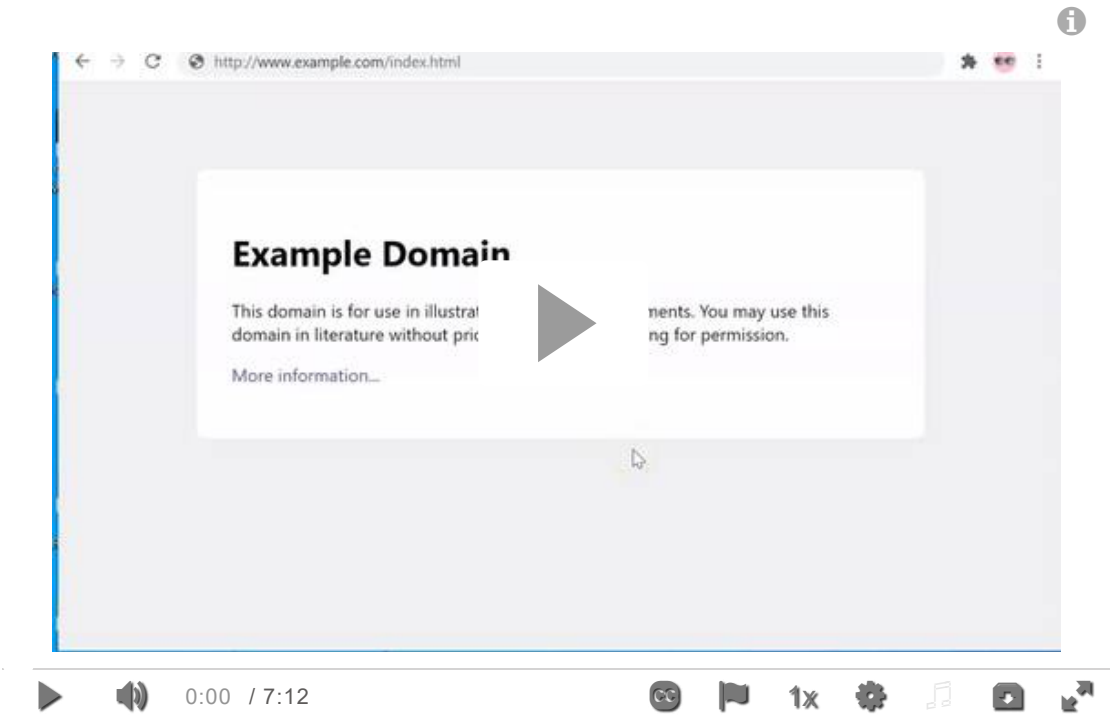
Exploration — HTTP Part 1: The Request

Introduction

As we discussed earlier, a client on the web communicates with the server using the HTTP protocol. Among other things, the HTTP protocol specifies the format for a request from a client to a server for a resource and the format for a response from the server to reply to such a request from the client. For this communication to occur, a connection needs to be created between the client and the server. The connection between the client and server is created using the **Transmission Control Protocol/Internet Protocol** or **TCP/IP** protocol. As an analogy with the telephone system, we can think of the TCP/IP protocol as the mechanism by which the telephone system establishes a phone connection between the caller and the receiver of the phone call, while the HTTP protocol as the language the caller and the receiver use once the connection has been established. Development of web applications occurs at the level of the HTTP protocol which in networking terminology is called an application level protocol. Details of the underlying TCP/IP connection are hidden by the HTTP protocol and are beyond the scope of this class.

Let's take a deeper look at the HTTP protocol by looking at the interaction that takes place to request the simple document identified by the URL `http://www.example.com/index.html`.



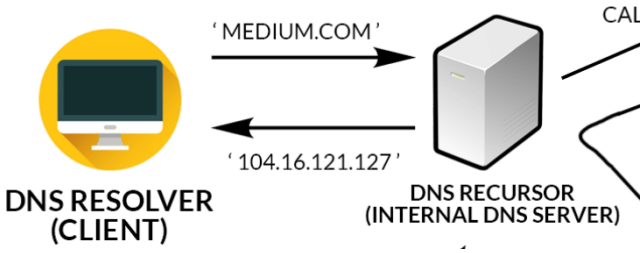
HTTP Request

Components of a URL

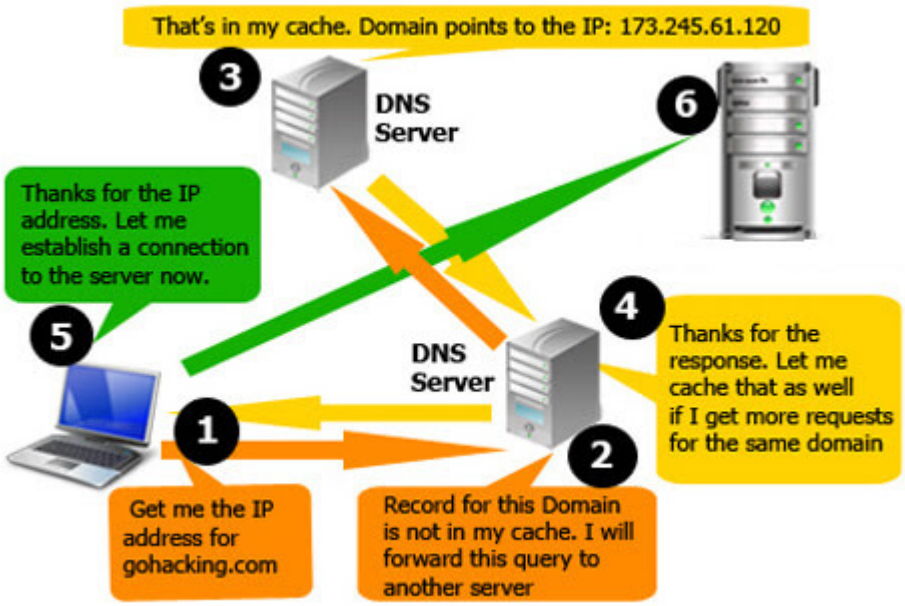
We can enter the URL of the document we want to get in our web client, typically a web browser. At the high level, the URL is just a formatted string. For now, we identify 3 parts of a URL - we will look at additional details later.

1. **Scheme or protocol:** This identifies the protocol used to send the request. In our example, this is `http`.
2. **Hostname:** This is the name of the host/server machine on which the server program is running and with which the web client needs to establish a TCP/IP connection. In our example this is `www.example.com`. Read more about the **difference between a Hostname and a Domain name (http://www.differencebetween.net/technology/difference-between-hostname-and-domain-name/)** from Sagar Khillar.
3. **Name of the resource at the server:** The server uses this to identify the specific resource that the client has requested. In our example, this is `index.html`.

Mapping Hostname to an IP Address



Based on the URL, our web client knows the hostname to talk to. However, the hostname is just a human-readable string and is insufficient to identify the machine on the internet. Every machine on the internet is identified by an **IP address (https://en.wikipedia.org/wiki/IP_address)** (Internet Protocol address). The web client needs to get the IP address in order to create a TCP/IP connection with the host machine. The internet using a naming system called **Domain Name System (https://en.wikipedia.org/wiki/Domain_Name_System)** or **DNS** to map the human-readable hostname to its IP address. To continue with our analogy with the telephone system, we can think of DNS as a phone directory which maps names of people to their phone number. In our example, the web client will make a request to a DNS server to get the IP address for `www.example.com`. The web client takes care of the communication with the DNS server and the creation of the TCP/IP connection and our web application doesn't have to deal with these details. Arindam Bhadra explains it another way in the diagram below:



HTTP Request: What is in it?

Now the web client sends the HTTP request to the server machine. For our example request `http://www.example.com/index.html`, the contents of the HTTP request are shown below:

```
GET /index.html HTTP/1.1
User-Agent: PostmanRuntime/7.26.8
Accept: * / *
Cache-Control: no-cache
Postman-Token: 0f29cc4e-bf63-40f0-959d-0f55855f6057
Host: www.example.com
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

An HTTP request has up to 4 parts:

1. **The request line.**
   - This is the first line in the request and has the following format: `Request-Method Resource HTTP/Version`
     - In our example, its value is `GET /index.html HTTP/1.1`
   - The request line starts with the `Request-Method`.
     - In our example, the `Request-Method` is `GET` which is used when we send a request to retrieve or read a resource.
     - Later, we will see other examples with different values of the `Request-Method`, including `POST` which is a common `Request-Method` to add or post a resource on the server.
   - Next the request line contains the resource on the server that is being requested.
     - In our example, the resource is `/index.html`.
   - Finally, the request line specifies the version of HTTP protocol being used. Currently both version 1.1 and version 2.0 are in wide use, with version 2.0 gaining a larger share with time.

2. **Request headers.**
   - The request line is followed by zero or more request headers.
   - These headers are in the form of `name:value` pairs.
   - The request headers are used to pass additional information to the server.
     - For example, the `Accept` header tells the server about the types of data the client can handle.
     - In the above example, the value `*/*` means that the client can accept all types of data.
     - To tell the server that the client can only handle HTML, the `Accept` header would be set to the value `text/html`.
3. **A blank line.**
4. **An optional body.**
   - Since the HTTP method `GET` is used to retrieve a document, the body is empty for a `GET` request.
   - When a client sends a `POST` request to send data to the server, the request body contains this data.

Summary

In this exploration, we studied one part of the HTTP protocol, namely, the HTTP request. In the next exploration, we continue the example and study the the generation and processing of an HTTP response.

Additional Resources

Here are some references to learn more about the topics we discussed in this exploration.

- The specifications of different web technologies are available as Request-for-Comments or **RFCs (https://en.wikipedia.org/wiki/Request_for_Comments)**. Here are the RFC for some HTTP versions: **HTTP 1.0 (https://tools.ietf.org/html/rfc1945)**, **HTTP 1.1 (https://tools.ietf.org/html/rfc2616)**
- Mozilla Developer Network or **MDN (https://developer.mozilla.org/en-US/)** is an excellent reference for all types of information about web development. A lot of information about the HTTP protocol (e.g., response codes, request and response headers) is available at MDN in a more user-friendly form than the formal RFCs.