School of Computer Science

Web and Database Computing 2019

Lecture 9: HTTP and the Client Server Model

adelaide.edu.au

seek LIGHT

The Client Server Model

What are clients and severs?

"Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients." - Wikipedia

- Clients access services and resources
  - Web browsers usually the clients in a web system.
- Servers provide those services and resources

What's involved?

1. The client establishes a connection to the server.
2. The client makes a request to the server.
3. The server sends a response to the client.
4. The client may make additional requests.
5. the connection is closed.

Establishing a connection

The client first needs to establish a TCP connection to the server.

How?

URLs

Uniform Resource Locator - An address that identifies a resource (web page) on the Internet

http://www.adelaide.edu.au:80/students

Protocol

- Application layer protocol used in network
- HTTP (Hypertext Transfer Protocol)
- HTTPS (SSL-encrypted communication HTTP)

Domain Name or Host name

- An identifier for Server/Host of the web application.
- If not already known to the client, must be resolved by the DNS to an IP address (Network layer) e.g. 192.168.1.1

URLs

Uniform Resource Locator - An address that identifies a resource (web page) on the Internet

http://www.adelaide.edu.au:80/students

Port

- An identifier of the particular process running on the server (Transport Layer)
- 65535 ports available, with 0 to 1023 reserved.
- Default port for HTTP is 80
- Default port for HTTPS is 443

Path

- Path to web page in the server
- Default path usually index.html if not given

URLs

See the full specification at https://tools.ietf.org/html/rfc1738

Establishing a connection

The client first needs to establish a TCP connection to the server.

From the URL, the client has the domain name and port number

- The client first resolves the Domain using the DNS
- Once the client knows the IP address of the server it can send data to that host, requesting the service on the given port.
- The server replies, ACKnowledging the receipt of the data
- The client sends a final ACKnowledgment of its own

Once connected, the client is ready to send a HTTP request.

HTTP Messages

- Messages exchanged between a server and a client
- Automatically generated by a web browser or web server
- HTTP Request:
  - Sent by a client to trigger an action on the server
- HTTP Response:
  - An answer to the client sent by the server
- Structure
  - Start line: the requests to be implemented or the response status
  - Headers: Meta meta information about the resource and client/server
  - Body: Data associated with the request or response

HTTP Start Line (Request)

1. Type of request
    - GET: get a resource (web page, image, etc)
    - POST: accept information related to a resource (usually form data)
    - HEAD: get information about the resource but not the resource itself
    - PUT: store this resource on the server
    - DELETE: delete a resource
2. Path of resource
    - From URL
3. Protocol Version of HTTP being used
    - Usually `HTTP/1.1` or `HTTP/2.0`

HTTP Start Line (Response)

1. Protocol Version of HTTP being used
    - Usually `HTTP/1.1` or `HTTP/2.0`
2. Status code inidcating if the request was able to be fulfilled
    - **1xx** Information (100 Continue)
    - **2xx** Success (200 OK)
    - **3xx** Instructions to client (301/302 to redirect, 304 to use cached version)
    - **4xx** Client Error (401 Unauthorized, 404 Not Found)
    - **5xx** Server Error (500 Internal Server Error)

http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

HTTP Headers

Headers provide us meta information about the resource and client/server

Example request headers:

- Accept: text/plain
- If-Modified-Since: Sat, 28 Aug 2010 19:45:22 GMT
- User-Agent e.g.
  - Host: giphy.com
  - Referer: https://google.com
  - User-Agent: Mozilla/5.0 (X11; Linux x86_64...) Gecko/20100101 Firefox/65.0

Example response headers

- Content-Type: text/html; charset=utf-8
- Date: Tue, 15 Nov 1994 08:12:31 GMT
- Last-Modified: Tue, 15 Dec 2010 12:45:31 GMT

https://tools.ietf.org/html/rfc7230

Sending a request

Demo of using Telnet to send a HTTP request

Sending a request

A better way (demo)


THE UNIVERSITY
of ADELAIDE

What's happening

Due:

- Prac Exercise 2 should be complete.
- Prac Exercise 3 now available with Websub

This week:

- Introduction to NodeJS & AJAX

Further learning:

- Try using telnet to send a HTTP request manually
- Download and install Insomnia Rest Client
- Keep using HTML and CSS in your forum posts.