HTTP Requests and Responses Reference

This is a reference sheet for your first activity but also the rest of this day's lesson.

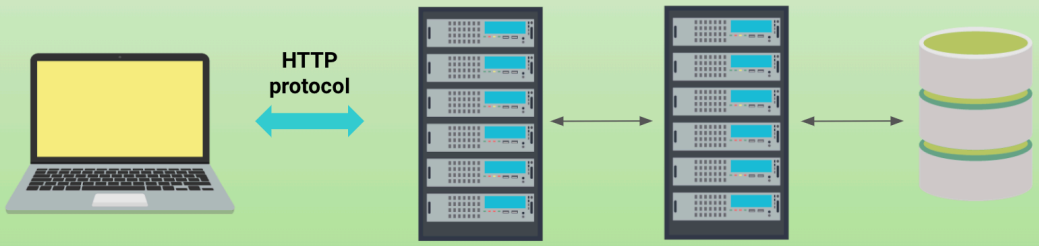
Table of Contents

Client-Server Architecture

An example of the typical web client and server architecture:

Client-Server Architecture

The client-server model is an exchange of information, a cycle of **requests and responses** between **clients and servers**.



Web client
A user's device communicates with a server to request resources from it.

Web server
The server queries the resources from internal servers it is connected to.

Web server
The server sends a response back to the client.

Database

HTTP

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers.

The HTTP Exchange

- The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.
- HTTP is based on the client-server architecture model and a stateless request/response protocol that operates by exchanging messages across a reliable TCP/IP connection.

The HTTP Client

- An HTTP "client" is a program (Web browser or any other client) that establishes a connection to a server for the purpose of sending one or more HTTP request messages.

The HTTP Server

- An HTTP "server" is a program (generally a web server like Apache Web Server) that accepts connections in order to serve HTTP requests by sending HTTP response messages.

Example of an HTTP Request

- An example HTTP GET request for the web page (http://www.example.com/hello.html):

```
GET /hello.html HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.example.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

  (this example has no request body)

Example of an HTTP Response

- An example HTTP 200 response to the request example above:

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
Content-Length: 88
Content-Type: text/html
Connection: Closed
```

- Example Response Body (HTML or HyperText Markup Language)

```
<html>
  <body>
  <h1>Hello, World!</h1>
  </body>
</html>
```

HTTP Requests

HTTP Request Methods

HTTP Method	Description
GET	The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.
HEAD	The HEAD method asks for a response identical to that of a GET request, but without the response body.
POST	The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server.
PUT	The PUT method replaces all current representations of the target resource with the request payload.
DELETE	The DELETE method deletes the specified resource.
CONNECT	The CONNECT method establishes a tunnel to the server identified by the target resource.
OPTIONS	The OPTIONS method is used to describe the communication options for the target resource.
TRACE	The TRACE method performs a message loop-back test along the path to the target resource.
PATCH	The PATCH method is used to apply partial modifications to a resource.

Common HTTP Request Headers

Header	*Usage and Paramaters*	*Description*
Authorization	{type} {credentials}	The HTTP Authorization request header contains the credentials to authenticate a user agent with a server, usually, but not necessarily, after the server has responded with a 401 Unauthorized status and the WWW-Authenticate header.
Referer	{url}	The Referer request header contains the address of the previous web page from which a link to the currently requested page was followed. The Referer header allows servers to identify where people are visiting them from and may use that data for analytics, logging, or optimized caching, for example.
Cookie	{Cookie*Name}={Cookie*Value}	The Cookie HTTP request header contains stored HTTP cookies previously sent by the server with the Set-Cookie header.
User-Agent	{product}	The User-Agent request header is a characteristic string that lets servers and network peers identify the application, operating system, vendor, and/or version of the requesting user agent.

HTTP Request Bodies

HTTP Responses

HTTP Response Status Codes

Status Code	Description
200 CODES	***SUCCESSFUL RESPONSES***
200 OK	The request has succeeded. The meaning of the success depends on the HTTP method
300 CODES	***REDIRECTS***

301 Moved Permanently	The URL of the requested resource has been changed permanently. The new URL is given in the response
400 CODES	***CLIENT ERRORS***
400 Bad Request	The server could not understand the request due to invalid syntax.
401 Unauthorized	Although the HTTP standard specifies "unauthorized", semantically this response means "unauthenticated". That is, the client must authenticate itself to get the requested response
403 Forbidden	The client does not have access rights to the content; that is, it is unauthorized, so the server is refusing to give the requested resource. Unlike 401, the client's identity is known to the server.
404 Not Found	The server can not find the requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist. Servers may also send this response instead of 403 to hide the existence of a resource from an unauthorized client. This response code is probably the most famous one due to its frequent occurrence on the web.
429 Too Many Requests	The user has sent too many requests in a given amount of time ("rate limiting".
500 CODES	***SERVER ERRORS***
500 Internal Server Error	The server has encountered a situation it doesn't know how to handle.
502 Bad Gateway	This error response means that the server, while working as a gateway to get a response needed to handle the request, got an invalid response.
503 Service Unavailable	The server is not ready to handle the request. Common causes are a server that is down for maintenance or that is overloaded. Note that together with this response, a user-friendly page explaining the problem should be sent. This responses should be used for temporary conditions and the Retry-After: HTTP header should, if possible, contain the estimated time before the recovery of the service. The webmaster must also take care about the caching-related headers that are sent along with

	this response, as these temporary condition responses should usually not be cached

HTTP Response Headers

HTTP Response Headers let the client and the server pass additional information with an HTTP request or response. An HTTP header consists of its case-insensitive name followed by a colon `:`, then by its value. Whitespace before the value is ignored.

Header	***Usage***	***Description***
Connection	keep-alive	Indicates that the client would like to keep the connection open. Having a persistent connection is the default on HTTP/1.1 requests.
Set-Cookie	The Set-Cookie HTTP response header is used to send cookies from the server to the user agent, so the user agent can send them back to the server later.	
X-XSS-Protection	The HTTP X-XSS-Protection response header is a feature of Internet Explorer, Chrome and Safari that stops pages from loading when they detect reflected cross-site scripting (XSS) attacks.	