**Header 1 Request and Response (GET):**

**Request**

Standard GET Request to / directory, using HTTP 1.1

GET / HTTP/1.1

The host and port that the request is being made to ( as the client sees it )

Host: localhost:9000

This tells the server the client wants to keep the tcp connection alive and reuse it for future responses.

Connection: keep-alive

This is the client telling the server it wants a fresh response, not a cached version (I had remove cache turned on for these requests), this is only actually needed for HTTP 1.0

Pragma: no-cache

HTTP 1.1 equivalent of the Pragma

Cache-Control: no-cache

This is chrome telling the server it would like to upgrade to HTTPS if possible

Upgrade-Insecure-Requests: 1

This is information about the client and its host – to help the server send the appropriate version of the page back, despite it being from Chrome, the Mozilla rendering engine is listed due to some older servers sending only bare bones HTML during the first browser war.

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_11\_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36

This is the client telling the server what types of content-type responses its capable of supporting.

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

This is the client telling the host where it came from originally, ie where it was referred from.

Referer: <http://localhost:9000/>

This lists the acceptable encoding that the client can handle.

Accept-Encoding: gzip, deflate, sdch

This is the client telling the server what types of language it will accept back, in this case US English.

Accept-Language: en-US,en;q=0.8

This is the cookie key value pair that indicates the user is logged in.

Cookie: PLAY\_SESSION=16f9065769d9c89dd4f8ffb4be6553ca719695dc-mySession=57cd5b2f3774db72fe040862

**Response**

This is the server telling the client it has successfully accepted the connection 200 return code of OK, and that it is using HTTP/1.1

HTTP/1.1 200 OK

This is the server telling the client the size of the response body, in bytes, in this case, 2821 bytes

Content-Length: 2821

This is the server telling the client the content type of the response is text/html using the utf-8 charset.

Content-Type: text/html; charset=utf-8

This is response time of the server in GMT time.

Date: Mon, 12 Sep 2016 10:05:15 GMT

**Header 2 Request and Response (Web Socket):**

**Request**

This is the client making a GET request to a web socket connection on localhost:9000/websocket with the query string topic=Day using HTTP 1.1

GET ws://127.0.0.1:9000/websocket?topic=Day HTTP/1.1

The host and port that the request is being made to ( as the client sees it )

Host: 127.0.0.1:9000

The client is asking to upgrade to a newer version of the protocol to support web sockets in this case.

Connection: Upgrade

This is the client telling the server it wants a fresh response, not a cached version (I had remove cache turned on for these requests), this is only actually needed for HTTP 1.0

Pragma: no-cache

HTTP 1.1 equivalent of the Pragma

Cache-Control: no-cache

The client is saying it wants to upgrade to a websocket connection

Upgrade: websocket

Initiates a request for cross-origin resource sharing. This is particularly important as this field can be validated on the server side to prevent hijacking.

Origin: <http://localhost:9000>

The client wants to use websocket version 13

Sec-WebSocket-Version: 13

This is information about the client and its host – to help the server send the appropriate version of the page back, despite it being from Chrome, the Mozilla rendering engine is listed due to some older servers sending only bare bones HTML during the first browser war.

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_11\_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36

This is the client telling the server what types of content-type responses its capable of supporting.

Accept-Encoding: gzip, deflate, sdch

This is the client telling the server what types of language it will accept back, in this case US English.

Accept-Language: en-US,en;q=0.8

This key is generated by the client randomly, when the server responds it will return a hashed version of this key

Sec-WebSocket-Key: ufYl2mnEpjL6pby0HwVGJw==

This is the client telling the server what types of web socket extensions it will support, the server will reply and negotiate these.

Sec-WebSocket-Extensions: permessage-deflate; client\_max\_window\_bits

**Response**

This server responds with a 101 response code, to acknowledge it will be switching protocols.

HTTP/1.1 101 Switching Protocols

This is the server telling the client that it is upgrading to the websocket protocol

Upgrade: websocket

This is the server saying that the connection has been upgraded.

Connection: Upgrade

This is the response to to the Sec-Websocket-Key sent by the client, it is the key + a constant GUI ID and then runs the SHA-1 hashing function to produce the key to complete the websocket upgrade handshake.

Sec-WebSocket-Accept: 9FJTnkr59RbZN37kkCqpDsUqvLg=

**Header 3 Request and Response (POST):**

**Request**

This is the client making a POST request to /remoteLogout over HTTP/1.1

POST /remoteLogout HTTP/1.1

The host and port that the request is being made to ( as the client sees it )

Host: localhost:9000

This tells the server the client wants to keep the tcp connection alive and reuse it for future responses.

Connection: keep-alive

This tells the server the clients request body size is 31 bytes.

Content-Length: 31

This is the client telling the server it wants a fresh response, not a cached version (I had remove cache turned on for these requests), this is only actually needed for HTTP 1.0

Pragma: no-cache

HTTP 1.1 equivalent of the Pragma

Cache-Control: no-cache

Origin: <http://localhost:9000>

This is chrome telling the server it would like to upgrade to HTTPS if possible

Upgrade-Insecure-Requests: 1

This is information about the client and its host – to help the server send the appropriate version of the page back, despite it being from Chrome, the Mozilla rendering engine is listed due to some older servers sending only bare bones HTML during the first browser war.

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_11\_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36 Content-Type: application/x-www-form-urlencoded

This is the client telling the server what types of content-type responses its capable of supporting.

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Referer: <http://localhost:9000/>

This lists the acceptable encoding that the client can handle.

Accept-Encoding: gzip, deflate

This is the client telling the server what types of language it will accept back, in this case US English.

Accept-Language: en-US,en;q=0.8

This is the client sending the cookie with the session details up to the server, as this was part of a log out quest,this information will be used to log out the correct client.

Cookie: PLAY\_SESSION=16f9065769d9c89dd4f8ffb4be6553ca719695dc-mySession=57cd5b2f3774db72fe040862

**Response**

This is the server telling the client to redirect to another page.

HTTP/1.1 303 See Other

This is where the server redirects the client to after the logout.

Location: /

This is the server telling the client there is nothing to download beyond this header, as all of the download will occur as part of a future request.

Content-Length: 0

This is response time of the server in GMT time.

Date: Mon, 12 Sep 2016 10:16:36 GMT