University of Sheffield

COM1001 SPRING SEMESTER

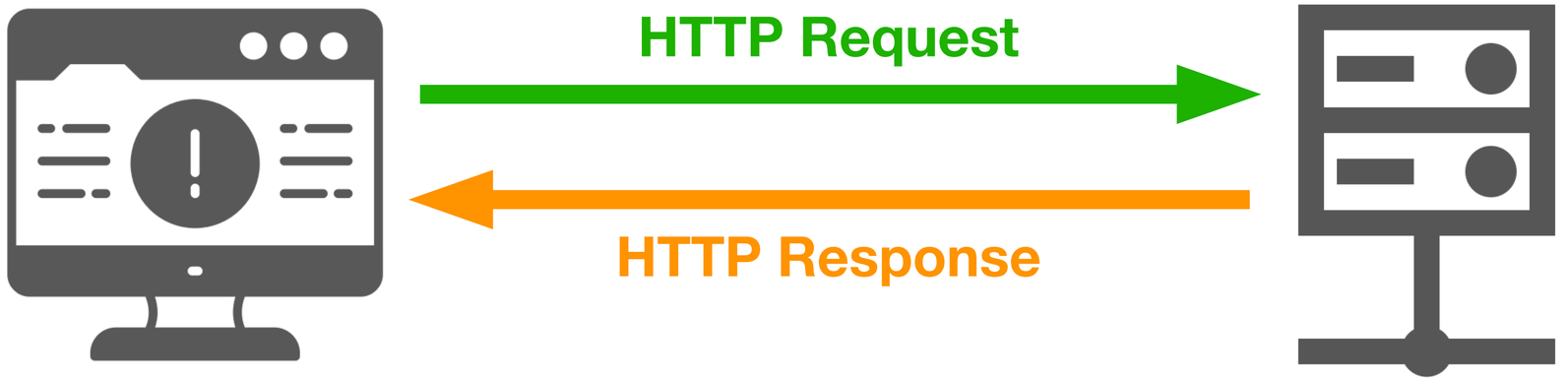
Professor Phil McMinn

p.mcminn@sheffield.ac.uk

The HyperText Transfer Protocol (HTTP)

What Happens When a User Accesses a Web Page in their Browser?

The user's **web browser** communicates with the **web server** hosting the page using an exchange of "**requests**" and "**responses**" according to the **HyperText Transfer Protocol (HTTP)**



Device with web browser

Web server hosting web site

The Path of Browser Request

1. The **browser** performs DNS Lookup for the **web server** based on its **domain name** (e.g. `www.sheffield.ac.uk`)
2. The **browser** sends an **HTTP Request** to the **web server**
3. The **web server** sends a **HTTP Response**, with the requested HTML file
4. The **browser** begins to render HTML
5. The **browser** sends additional requests for objects embedded in the HTML file (CSS files, images, JavaScript, etc.)

1 DNS Lookup

A **domain name** like “`www.sheffield.ac.uk`” is easy for humans to remember.

An **Internet Protocol (IP)** address like `143.167.2.102` is not, but this is how servers on the Internet are located.

So the first step in requesting a page is converting the domain to an IP address.

This information is managed by one of the 13 “root” **domain name servers (DNS)** that are located around the world.

2 Browser sends HTTP Request

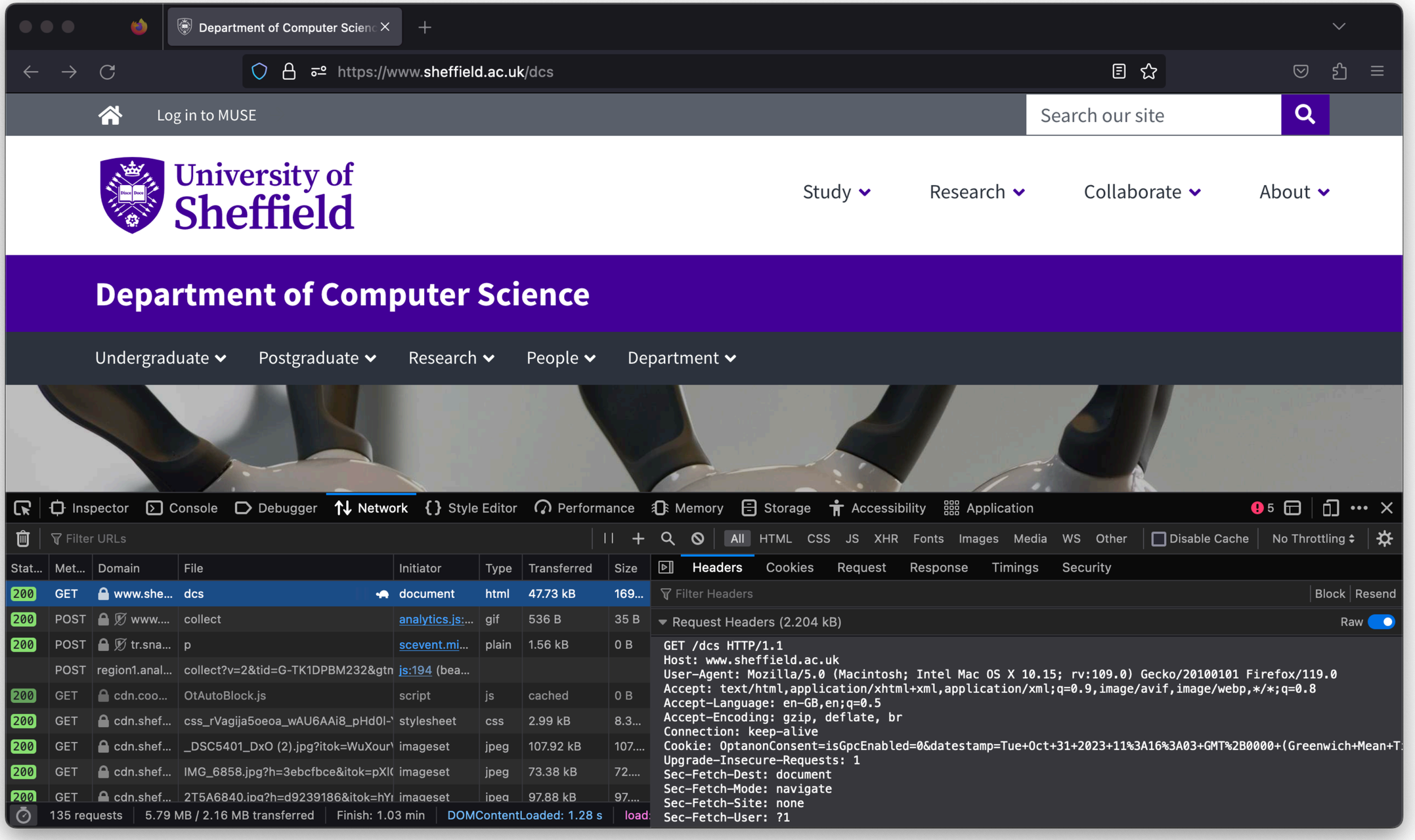
Once it has the IP address of the web server, a browser can now send that server a **HTTP Request**.

It doesn't have to literally be **HTTP** – it could be **HTTPS**, or **HTTP/2**, or even **HTTP/3**.

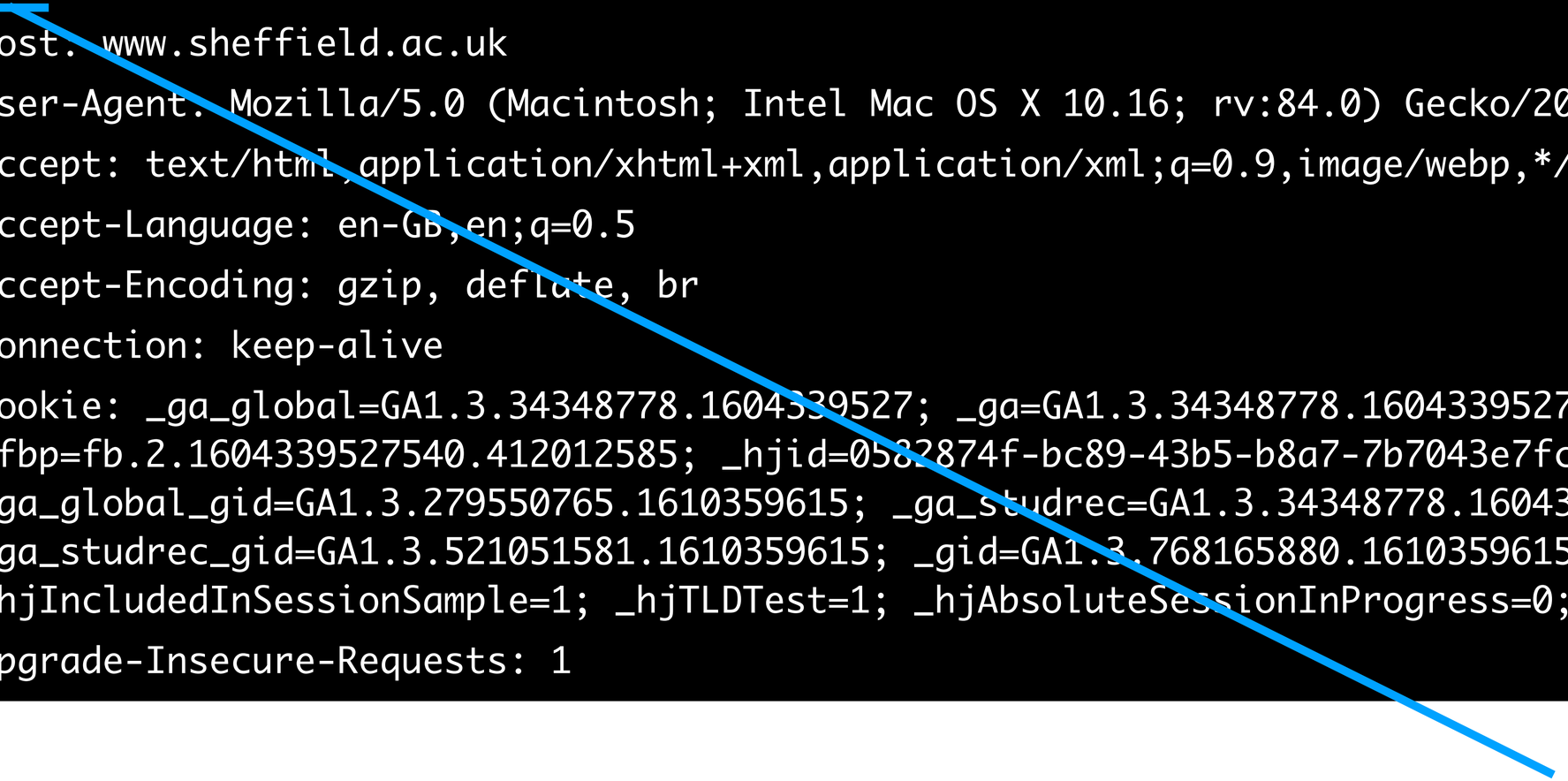
These later versions of the HTTP protocol following HTTP/1.1 do not really have any implications for web applications, they just make HTTP…

… more secure (HTTPS)

… faster (HTTP/2 and HTTP/3)


Department of Computer Scienc
https://www.sheffield.ac.uk/dcs
Log in to MUSE
Search our site
University of Sheffield
Study
Research
Collaborate
About
Department of Computer Science
Undergraduate
Postgraduate
Research
People
Department
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Filter URLs
All HTML CSS JS XHR Fonts Images Media WS Other
Disable Cache
No Throttling
Stat... Met... Domain File Initiator Type Transferred Size
200 GET www.she... dcs document html 47.73 kB 169...
200 POST www.... collect analytics.js:... gif 536 B 35 B
200 POST tr.sna... p scevent.mi... plain 1.56 kB 0 B
POST region1.anal... collect?v=2&tid=G-TK1DPBM232&gtm js:194 (bea...
200 GET cdn.coo... OtAutoBlock.js script js cached 0 B
200 GET cdn.shef... css_rVagija5oeoa_wAU6AAi8_pHd0l- stylesheet css 2.99 kB 8.3...
200 GET cdn.shef... _DSC5401_DxO (2).jpg?itok=WuXour imageset jpeg 107.92 kB 107....
200 GET cdn.shef... IMG_6858.jpg?h=3ebcfbce&itok=pXI imageset jpeg 73.38 kB 72....
200 GET cdn.shef... 2T5A6840.jpg?h=d9239186&itok=hY imageset jpeg 97.88 kB 97....
135 requests
5.79 MB / 2.16 MB transferred
Finish: 1.03 min
DOMContentLoaded: 1.28 s
Headers Cookies Request Response Timings Security
Filter Headers
Block Resend
Request Headers (2.204 kB)
Raw
GET /dcs HTTP/1.1
Host: www.sheffield.ac.uk
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/119.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: OptanonConsent=isGpcEnabled=0&datestamp=Tue+Oct+31+2023+11%3A16%3A03+GMT%2B0000+(Greenwich+Mean+T
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

```
GET /dcs HTTP/1.1
Host: www.sheffield.ac.uk
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:84.0) Gecko/20100101 Firefox/84.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: _ga_global=GA1.3.34348778.1604339527; _ga=GA1.3.34348778.1604339527;
_fbp=fb.2.1604339527540.412012585; _hjid=0582874f-bc89-43b5-b8a7-7b7043e7fc80;
_ga_global_gid=GA1.3.279550765.1610359615; _ga_studrec=GA1.3.34348778.1604339527;
_ga_studrec_gid=GA1.3.521051581.1610359615; _gid=GA1.3.768165880.1610359615;
_hjIncludedInSessionSample=1; _hjTLDTest=1; _hjAbsoluteSessionInProgress=0; _gat_UA-2341502-28=1
Upgrade-Insecure-Requests: 1
```

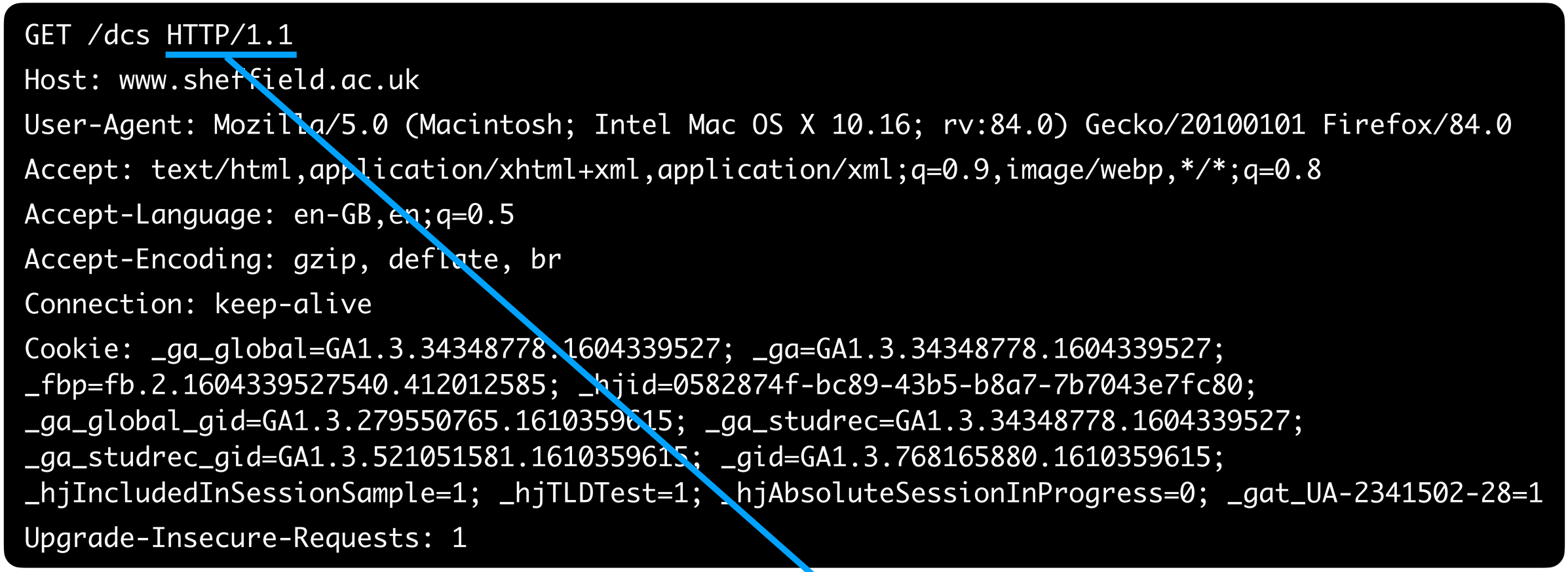


The first part of the first line of a **HTTP Request** is the **HTTP method**.

The **HTTP method** defines the type of request being made and therefore how the server will interpret it. The most important **HTTP methods** are `GET` and `POST`.

```
GET /dcs HTTP/1.1
Host: www.sheffield.ac.uk
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:84.0) Gecko/20100101 Firefox/84.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: _ga_global=GA1.3.34348778.1604339527; _ga=GA1.3.34348778.1604339527;
_fbp=fb.2.1604339527540.412012585; _hjid=0582874f-bc89-43b5-b8a7-7b7043e7fc80;
_ga_global_gid=GA1.3.279550765.1610359615; _ga_studrec=GA1.3.34348778.1604339527;
_ga_studrec_gid=GA1.3.521051581.1610359615; _gid=GA1.3.768165880.1610359615;
_hjIncludedInSessionSample=1; _hjTLDTest=1; _hjAbsoluteSessionInProgress=0; _gat_UA-2341502-28=1
Upgrade-Insecure-Requests: 1
```

Secondly, we have the **resource identifier** of the **resource** being requested, which could be a web page, or any type of file (such an image, script, or document).



The resource identifier is followed by the **HTTP protocol** being used.

```
GET /dcs HTTP/1.1
Host: www.sheffield.ac.uk
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:84.0) Gecko/20100101 Firefox/84.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: _ga_global=GA1.3.34348778.1604339527; _ga=GA1.3.34348778.1604339527;
_fbp=fb.2.1604339527540.412012585; _hjid=0582874f-bc89-43b5-b8a7-7b7043e7fc80;
_ga_global_gid=GA1.3.279550765.1610359615; _ga_studrec=GA1.3.34348778.1604339527;
_ga_studrec_gid=GA1.3.521051581.1610359615; _gid=GA1.3.768165880.1610359615;
_hjIncludedInSessionSample=1; _hjTLDTest=1; _hjAbsoluteSessionInProgress=0; _gat_UA-2341502-28=1
Upgrade-Insecure-Requests: 1
```

Finally, the **HTTP Request** contains a series of **Request Headers**

Most of these are not particularly important for this module.

```
GET /dcs HTTP/1.1
Host: www.sheffield.ac.uk
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:84.0) Gecko/20100101 Firefox/84.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: _ga_global=GA1.3.34348778.1604339527; _ga=GA1.3.34348778.1604339527;
_fbp=fb.2.1604339527540.412012585; _hjid=0582874f-bc89-43b5-b8a7-7b7043e7fc80;
_ga_global_gid=GA1.3.279550765.1610359615; _ga_studrec=GA1.3.34348778.1604339527;
_ga_studrec_gid=GA1.3.521051581.1610359615; _gid=GA1.3.768165880.1610359615;
_hjIncludedInSessionSample=1; _hjTLDTest=1; _hjAbsoluteSessionInProgress=0; _gat_UA-2341502-28=1
Upgrade-Insecure-Requests: 1
```

Finally, the **HTTP Request** contains a series of **Request Headers**

Most of these are not particularly important for this module.

Note the `Cookie` header, however – we will be returning to the topic of **cookies** in a later lecture.

3 Web Server sends HTTP Response

Once the server receives a **HTTP Request**, it will respond in the form of a **HTTP Response**. How it does depends on a number of things.

For example, the resource (i.e., a web page) may not actually exist.

The response consists of a series of headers (like the request did), and the **body**, which contains the resource requested (e.g., the HTML of a web page).

Here's the initial part of the **HTTP response** Sheffield University's web server sent for the **HTTP request** for `http://www.sheffield.ac.uk/dcs`

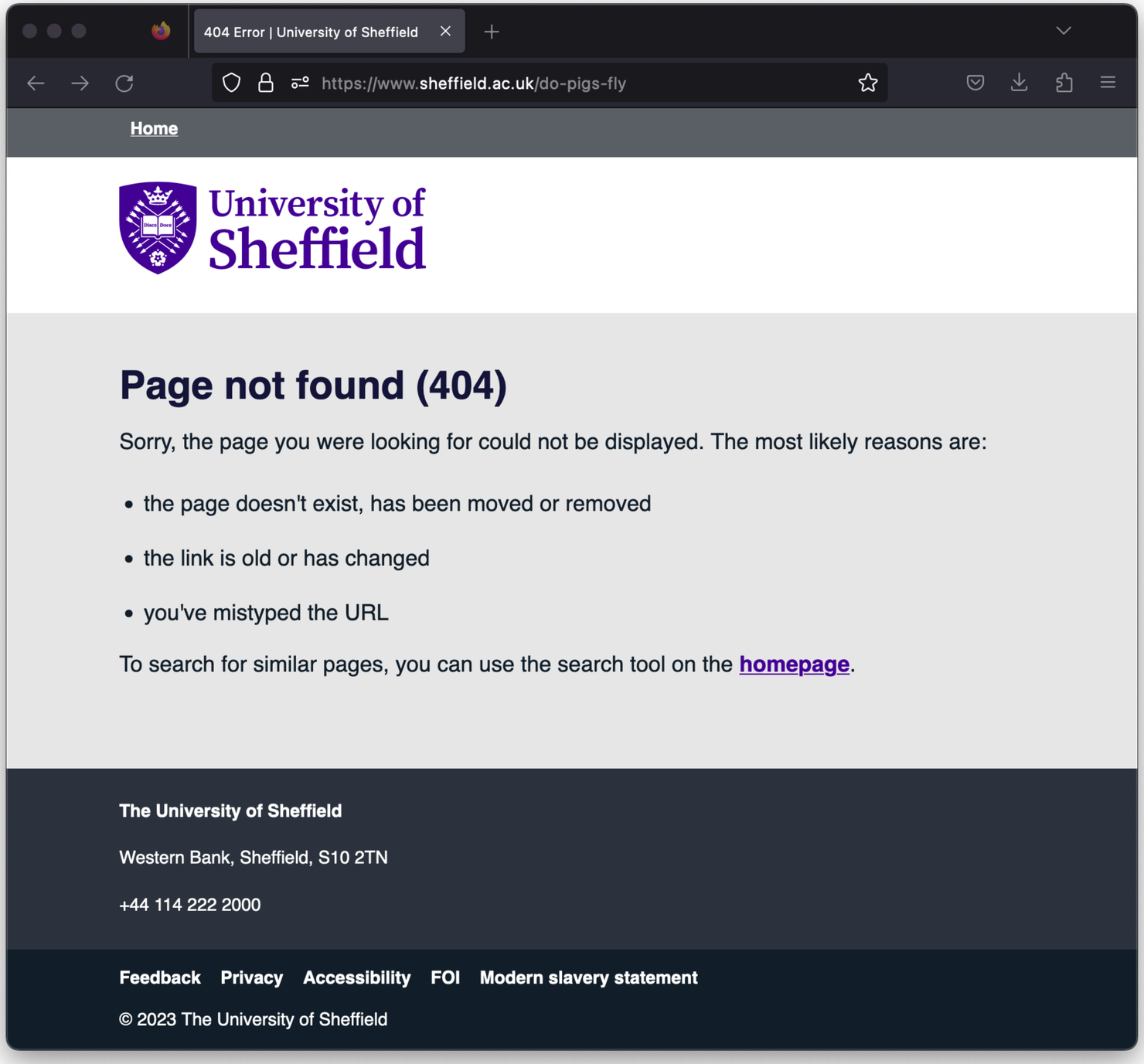
```
HTTP/1.1 200 OK
X-Drupal-Dynamic-Cache: MISS
X-Varnish: 341153 402883
X-Grace: none
Vary: Accept-Encoding,X-Forwarded-Host
Cache-Control: max-age=900, public
X-Cache: HIT (drupal-cache-live1)
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Strict-Transport-Security: max-age=31536000; preload
Link: <https://www.sheffield.ac.uk/dcs>; rel="canonical"
…
```

The most important part of the response headers is the **status code**.
Ideally it sends a `200 OK`, which means success.

But it may send a `404 Not Found` or a `500 Internal Server Error`.

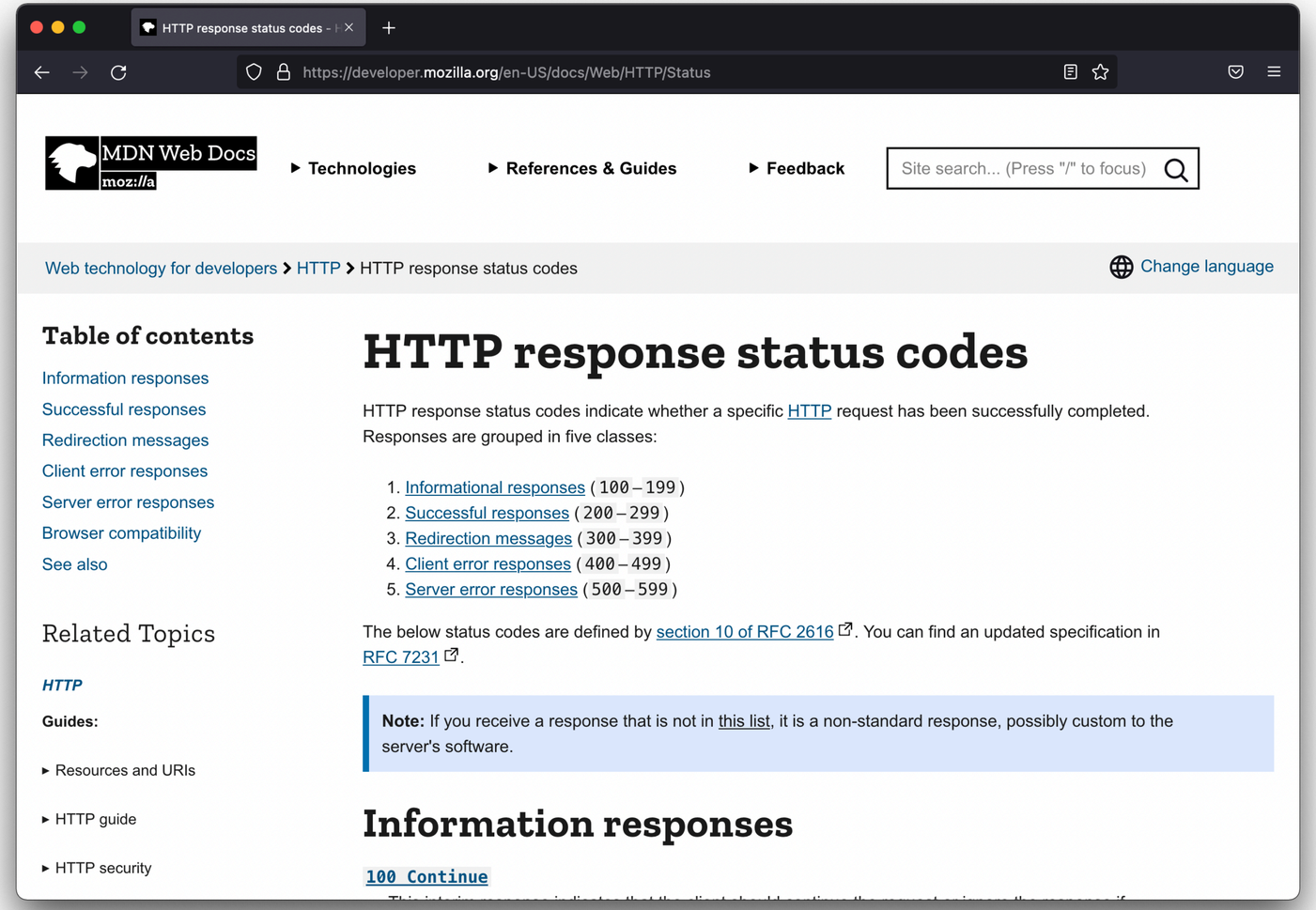
Both of these mean the requested resource cannot be sent.

```
HTTP/1.1 200 OK
X-Drupal-Dynamic-Cache: MISS
X-Varnish: 341153 402883
X-Grace: none
Vary: Accept-Encoding,X-Forwarded-Host
Cache-Control: max-age=900, public
X-Cache: HIT (drupal-cache-live1)
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Strict-Transport-Security: max-age=31536000; preload
Link: <https://www.sheffield.ac.uk/dcs>; rel="canonical"
…
```


404 Error | University of Sheffield
https://www.sheffield.ac.uk/do-pigs-fly
Home
University of Sheffield
Page not found (404)
Sorry, the page you were looking for could not be displayed. The most likely reasons are:
the page doesn't exist, has been moved or removed
the link is old or has changed
you've mistyped the URL
To search for similar pages, you can use the search tool on the homepage.
The University of Sheffield
Western Bank, Sheffield, S10 2TN
+44 114 222 2000
Feedback Privacy Accessibility FOI Modern slavery statement
© 2023 The University of Sheffield

More on Status Codes

developer.mozilla.org/en-US/docs/Web/HTTP/Status



Important HTTP Status Codes You Need to Know

200 (OK) – if the web application successfully processes the request

302 (Redirect) – if the web application re-directed the request (e.g., to an alternative resource ID)

404 (Not Found) – if the web application could not find the resource requested.

500 (Internal Server Error) – if the web application encountered an error while trying to process the request (e.g., its code contained a bug)

The Body of the HTTP Response: Static vs Dynamic Resources

Depending on the nature of the resource, there may be more work for the web server to do in generating the body of the HTTP response.

Static resources ***already exist*** before the request is made – e.g., an image files. Sometimes whole websites are static – the HTML pre-exists too. In this case, the server just needs to locate the file and send it to the browser.

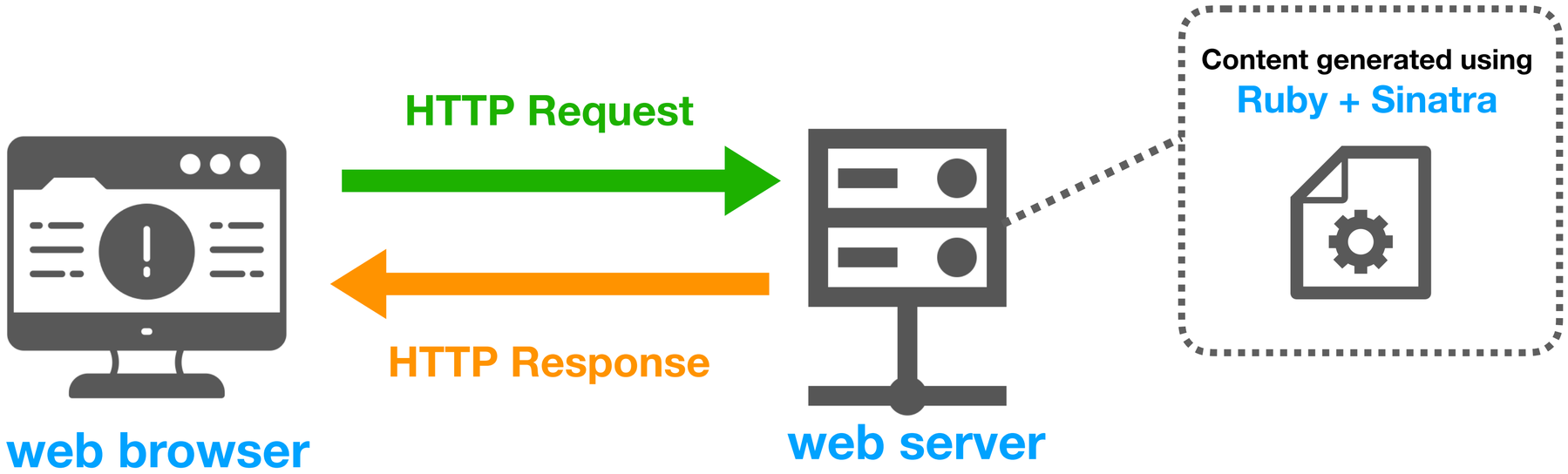
Dynamic resources ***are generated as a result of a request for them***. If a resource is dynamically generated, the **web server needs to execute some code to generate the content before it can send it.**

Dynamic Content Generation

If we can dynamically generate content, the web pages can respond to user actions and change the information appearing in the web page. This information could come from a **database**, for example.

In other words, we can write **web applications**.

In this module, we will be dynamically generating web content using **Ruby** and the help of a domain specific language called **Sinatra**.



4 The Browser Renders the HTML

Once the browser receives an HTML file, as part of the body of the HTTP Response, it can then process it and render it onto the screen.

5 The Browser Sends Additional Requests

During processing of the HTML file, the browser may find that it needs to request additional files (e.g., images and scripts, etc.) and will send additional HTTP Requests for those.

Why is all of this important?

Our **Sinatra applications** will need to respond to **HTTP requests** and generate content to form the **HTTP response**.

So its important to know a bit about what constitutes a HTTP request and a response – although we need not be concerned with all the details.

HTTP – Summary

To understand how to write a web application we need to have an understanding of HTTP (hypertext transfer protocol).

- Browsers send **HTTP requests** to a web server, which a web application processes, sending back an appropriate **HTTP response**.

Two important parts of the HTTP request are the HTTP method being used and the identifier of the resource being requested.

- The most important HTTP methods are `GET` and `POST`.

Two important parts of the HTTP response include its body (typically the HTML of a web page) and its status code.

- Important codes include **200 (OK)**, **302 (Redirect)**, **404 (Not Found)**, **500 (Internal Server Error)**.