

CIS 658 Web Architectures

HTTP



GRAND VALLEY
STATE UNIVERSITY®

Lecturer: **Dr. Yong Zhuang**

Practice: Sass

Use the online playground to convert the CSS code into Sass. Implement the following features:

- Inheritance: Create a `%flex-layout` placeholder to share the `display`, `align-items`, and `padding` between both boxes.
- Mixins: Create a `@mixin alert-theme($bg, $text)` to handle the background and text colors in one line.
- Nesting: Nest the `.alert-title` inside the `.alert-box`.
- Operators: Use a `$base-unit: 5px`; and set the `border-radius` and `margin-right` using multiplication.



HTTP

- HyperText Transfer Protocol
- Invented by Tim Berners Lee @ CERN
- A protocol for delivering resources over the web
- TCP/IP connections, default (server) port 80
- HTTP client & HTTP server



Other network Transfer Protocols

- FTP: File Transfer Protocol
- FTPS: Secure FTP
- SMTP: Simple Message Transfer Protocol
- NTP: Network Time Protocol

Why learn the details of HTTP?



Why learn the details of HTTP?

HTTP requests from your program

- To understand how your program sends and receives HTTP requests.
- To debug and optimize HTTP requests from your program.
- To design better APIs and handle client–server communication effectively.

Web Client/Server Architecture

(3) Send “contents” (HTML + CSS + JS & other data)

(4) Present “contents”

(5) Run Code

(6) Handle user input

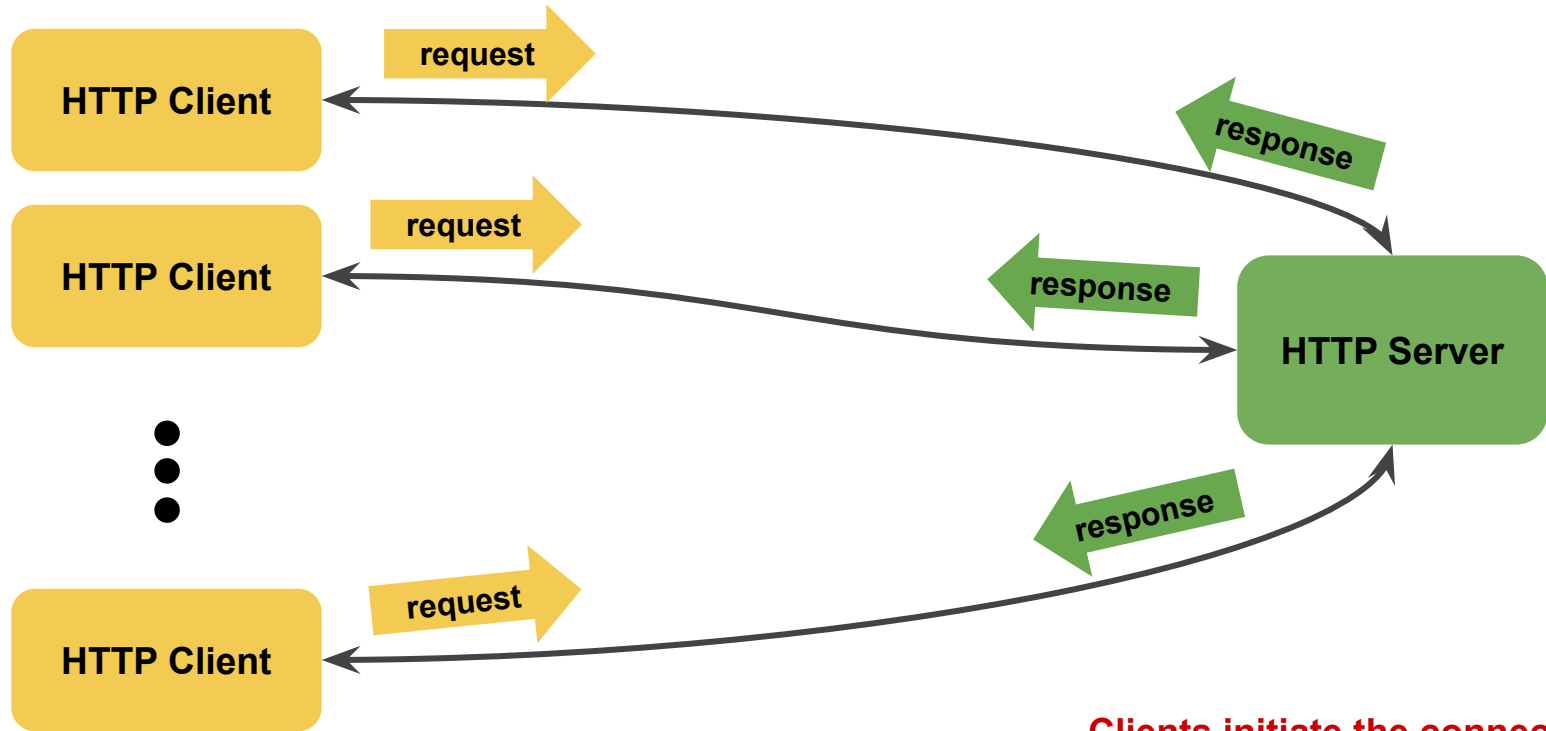


(2) Run Code



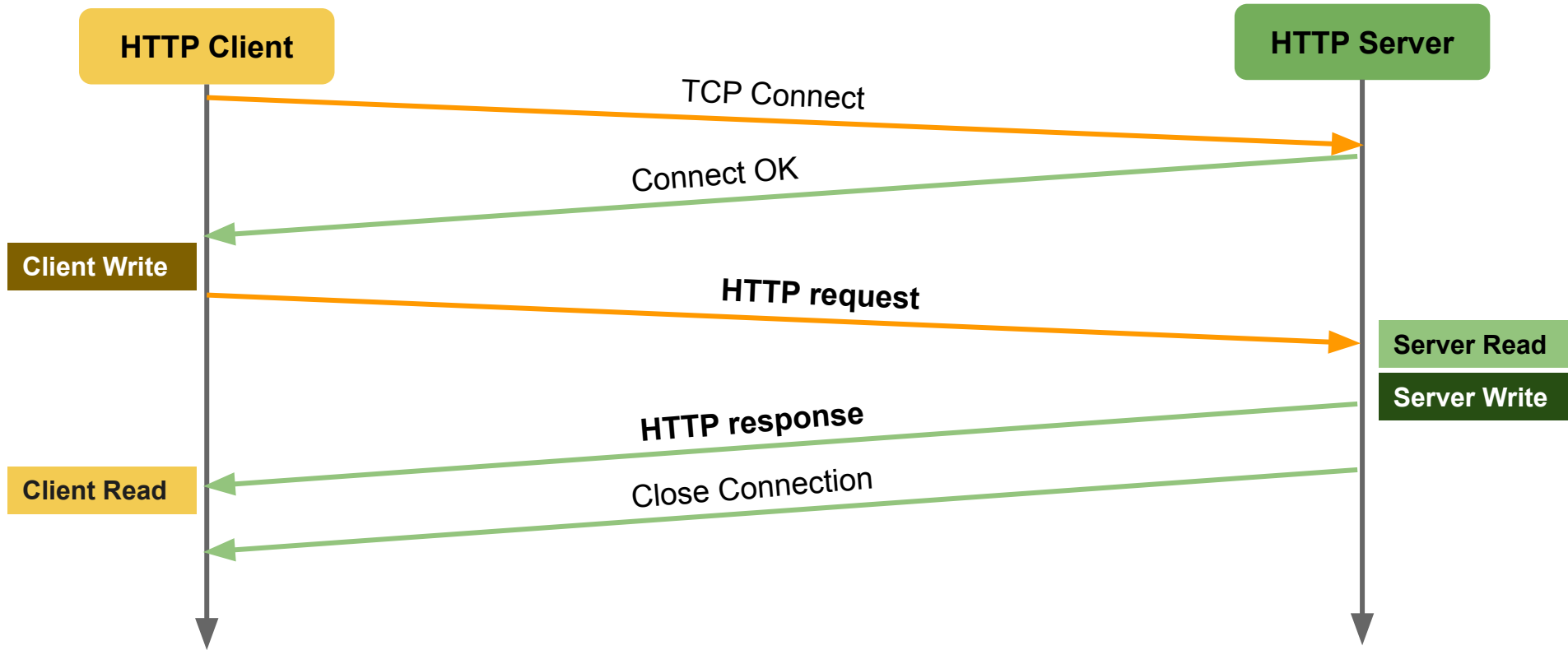
(1) Send “user input”

HTTP Communication Model



Clients initiate the connection!!!

Transaction Timeline (TCP Sockets)

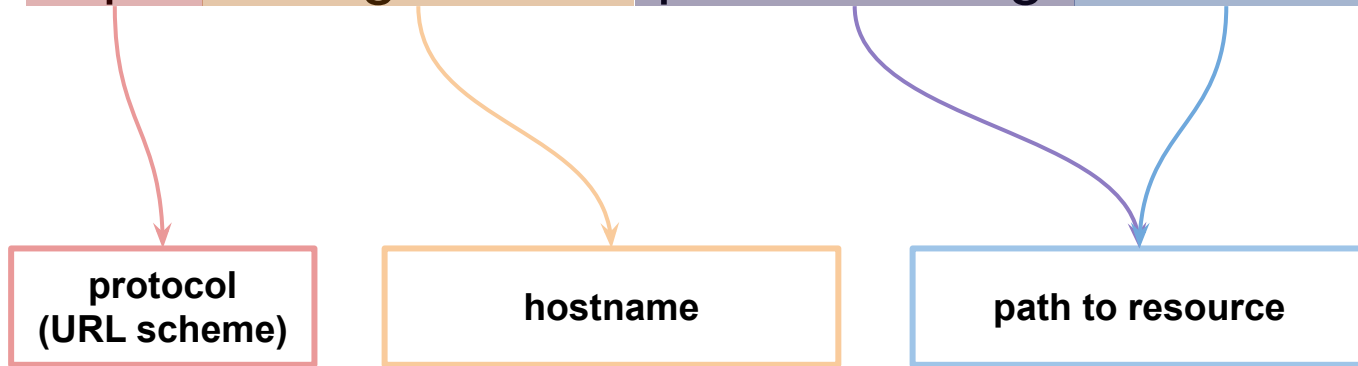


HTTP URL: Uniform Resource Locator

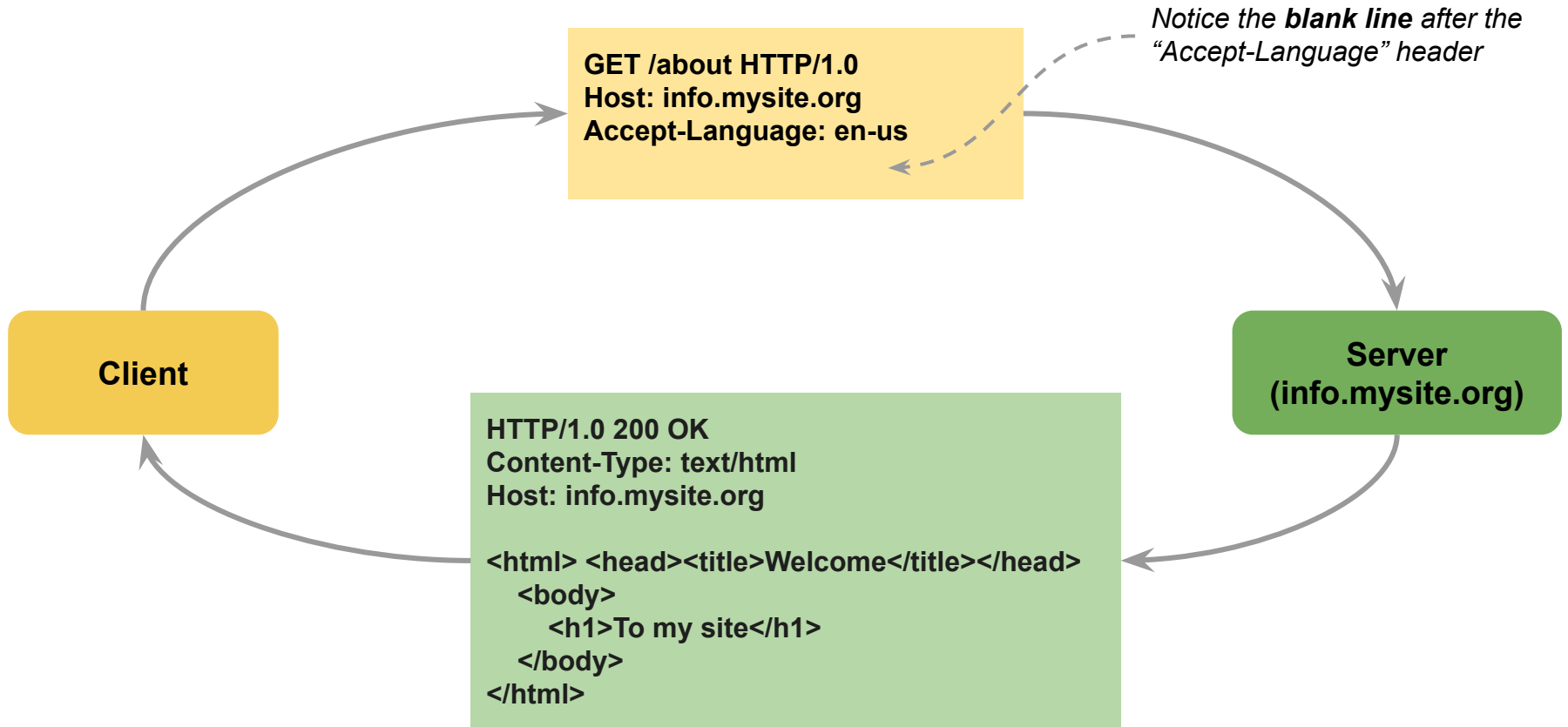
http:// www.gvsu.edu /files/registrar/622GX/7155/ admission.pdf

http:// www.gvsu.edu /files/img/article/frontpag/ 5123FG73A.jpg

http:// www.gvsu.edu /pcec/advising/ index.html



http://info.mysite.org/about/



Web Browser DevTools (Network Tab)

`http://info.cern.ch`

The screenshot displays the Network Tab in Chrome DevTools. The top toolbar includes icons for Inspector, Console, Debugger, Network (highlighted with a red box), Style Editor, Performance, and a menu icon. Below the toolbar is a search bar for Filter URLs and a 'No Throttling' dropdown. The main area shows a list of network requests, with the first request selected: a GET request to `http://info.cern.ch/` (646 B). The details pane for this request is expanded, showing the following information:

- Status:** 200 OK
- Version:** HTTP/1.1
- Transferred:** 646 B (646 B size)
- Request Priority:** Highest
- DNS Resolution:** System
- Response Headers (232 B):**
 - Accept-Ranges: bytes
 - Connection: close
 - Content-Length: 646
 - Content-Type: text/html
 - Date: Mon, 11 Sep 2023 14:25:55 GMT
 - ETag: "286-4f1aadb3105c0"
 - Last-Modified: Wed, 05 Feb 2014 16:00:31 GMT
 - Server: Apache
- Request Headers (346 B):**
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
 - Accept-Encoding: gzip, deflate
 - Accept-Language: en-US,en;q=0.5
 - Connection: keep-alive
 - Host: info.cern.ch
 - Upgrade-Insecure-Requests: 1
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0

Red arrows point to the request URL, the 'Response Headers' section, and the 'Request Headers' section.

Fetch the content via the command line

```
curl --verbose http://info.cern.ch
```

(On Linux/OSX/Windows 10 WSL)

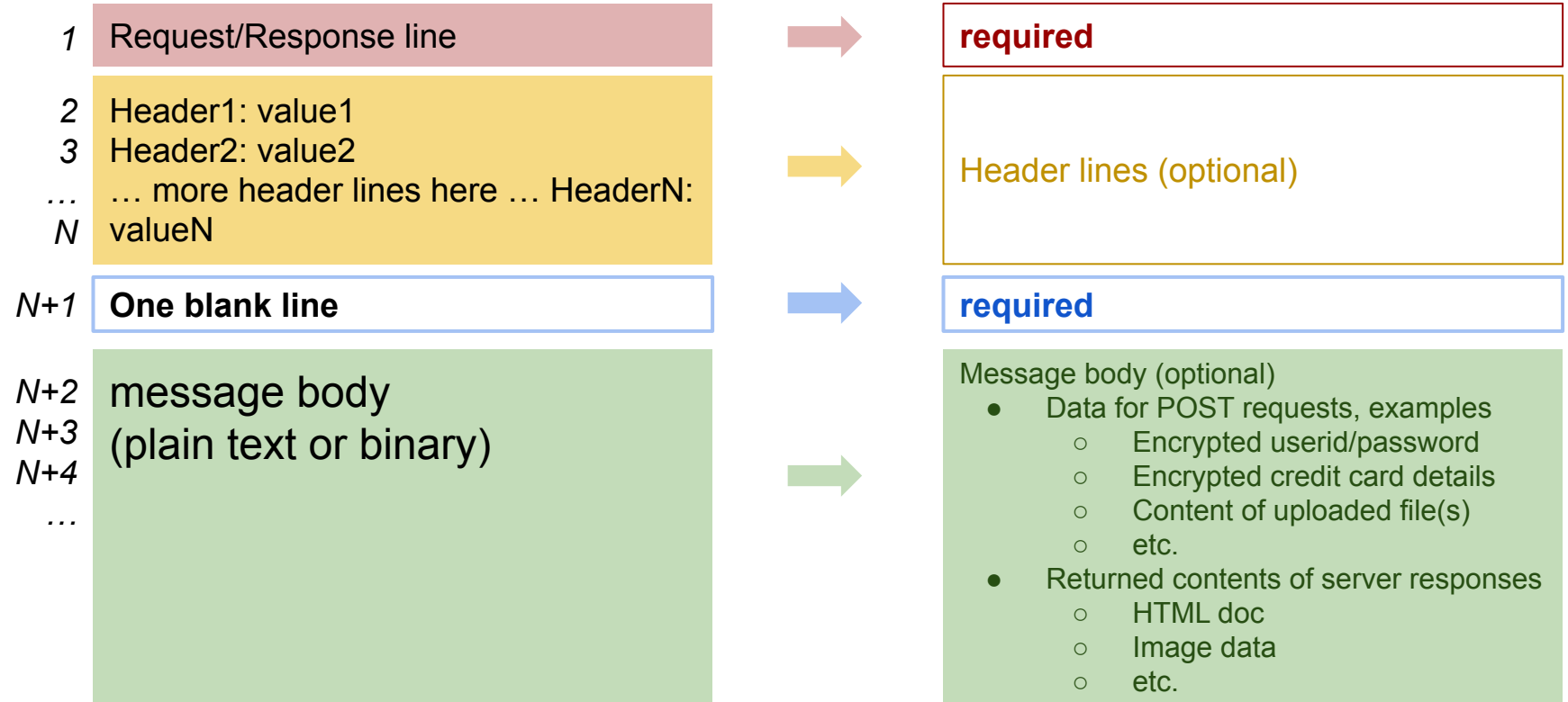
Fetch the content via the command line

```
iwr http://info.cern.ch -UseBasicParsing
```

(On Windows PowerShell)

HTTP Request/Response

line



HTTP headers of interest to web developers

Header	Description	Example
Accept	Inform server media-type to respond	Accept: image/jpg
Accept-Language	Inform the server which languages the client is able to understand	Accept-Language: en-US; en-UK
Content-Type	Media type of the returned content	Content-Type: plain/text
Content-Language	The languages of the content	Content-Language: en-US
Date	Date and time of the message	Date: Mon, 21 Aug 2017 18:14:36 GMT
ETag	Identifier used by caching algorithms	ETag: ""8a9-291e721905000"
Host	Specify the domain name of the intended server (mainly for Virtual Hosting)	Host: www.personal.me:5555

HTTP 1.0 Commands (Request Methods)

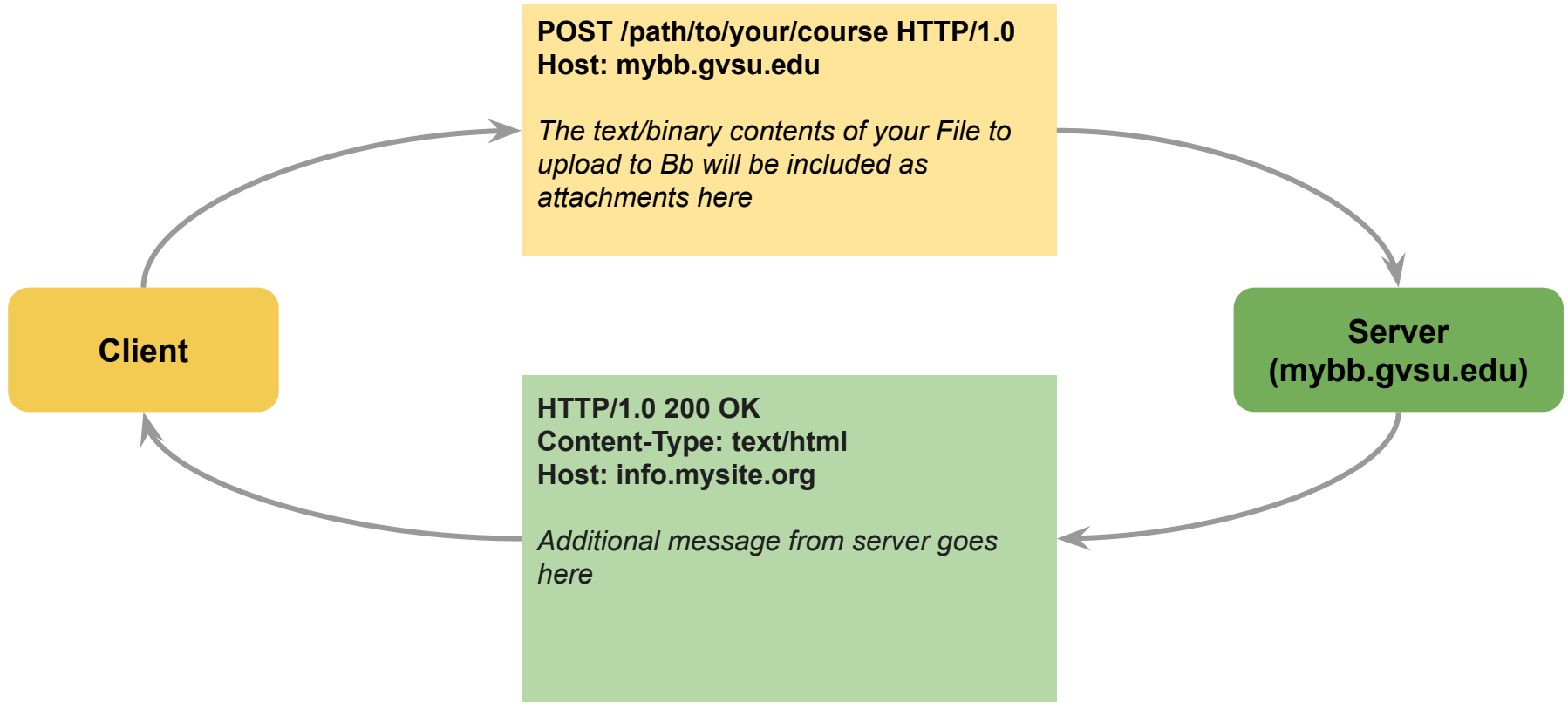
- GET
 - POST
 - HEAD
- More-frequently used

(like GET but the server responds only with header, no data)

- PUT
 - DELETE
 - OPTIONS
- Less-frequently used

Operation	HTTP Request
Create	POST
Read	GET
Update	PUT
Delete	DELETE

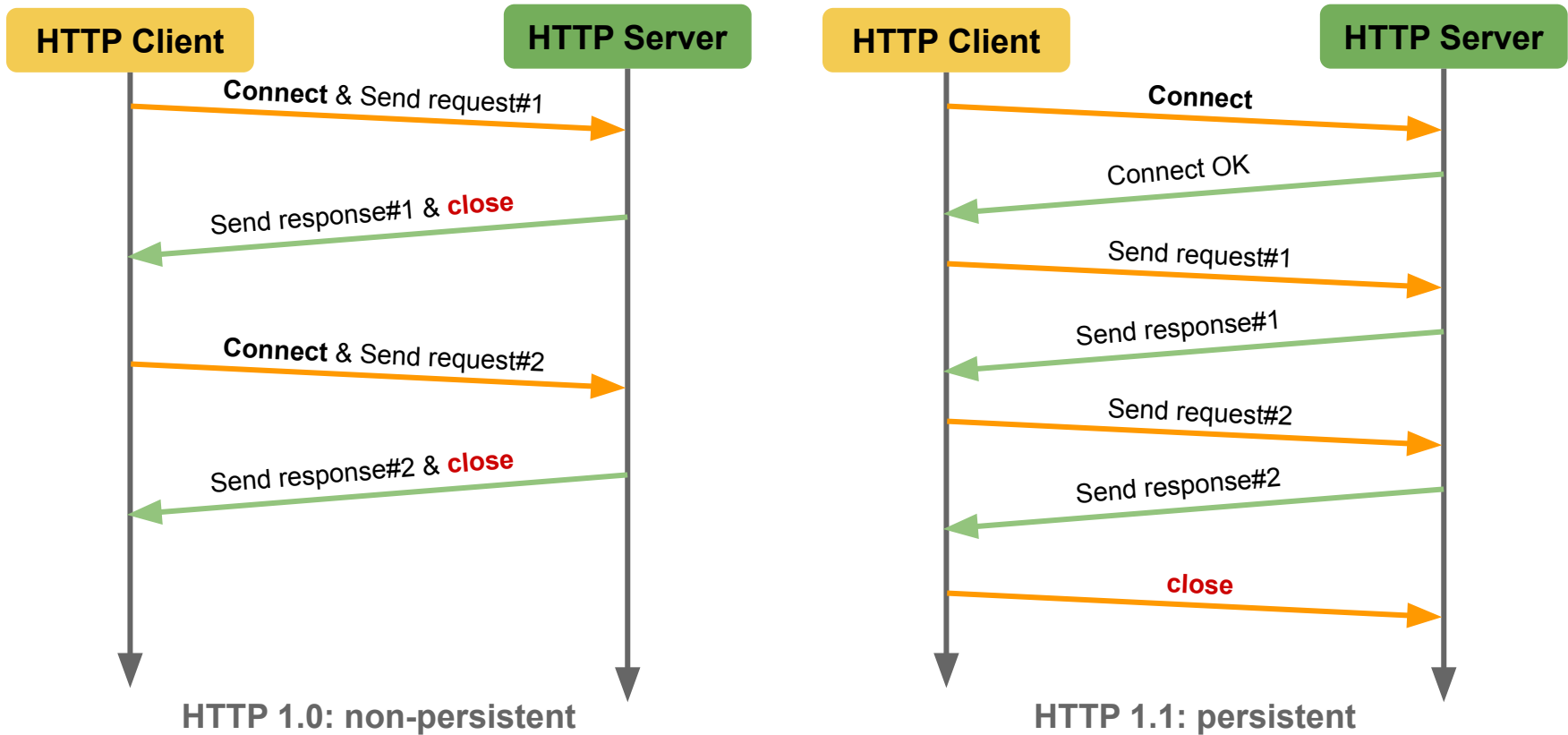
POST: upload file to Bb



HTTP Status Code

Status Code	Description
1xx	Informational messages
2xx	Success messages
3xx	Redirect message
4xx	Error on the client's behalf
5xx	Error on the server's behalf

HTTP Connections: Persistence



HTTP 1.0 vs. HTTP 1.1

HTTP 1.0

- One request per connection (non-persistent)
- Cache control is **timestamp based** with one-second resolution (inaccurate)
- Client cannot request a portion of a resource
- Responses are delivered in one big chunk

HTTP 1.1

- N requests per connection (persistent)
- Response can be delivered in chunk
- Cache control is **content based**, responses include entity tag (Etag), similar to hash value
- Clients can request **partial content**
 - “Range:” header line in HTTP request
- Responses may be delivered in many small chunks

HTTP 1.1 vs. HTTP 2

HTTP 1.1

- HTTP messages encoded in text format
- Require multiple connections to achieve concurrency
- Uncompressed response headers
- No resource prioritization

HTTP 2

- HTTP messages encoded in binary format
 - Message = request or response
- Multiple concurrent channels on a single connection
- Compressed response headers
- Resource prioritization (important requests complete more quickly)