CSE 106

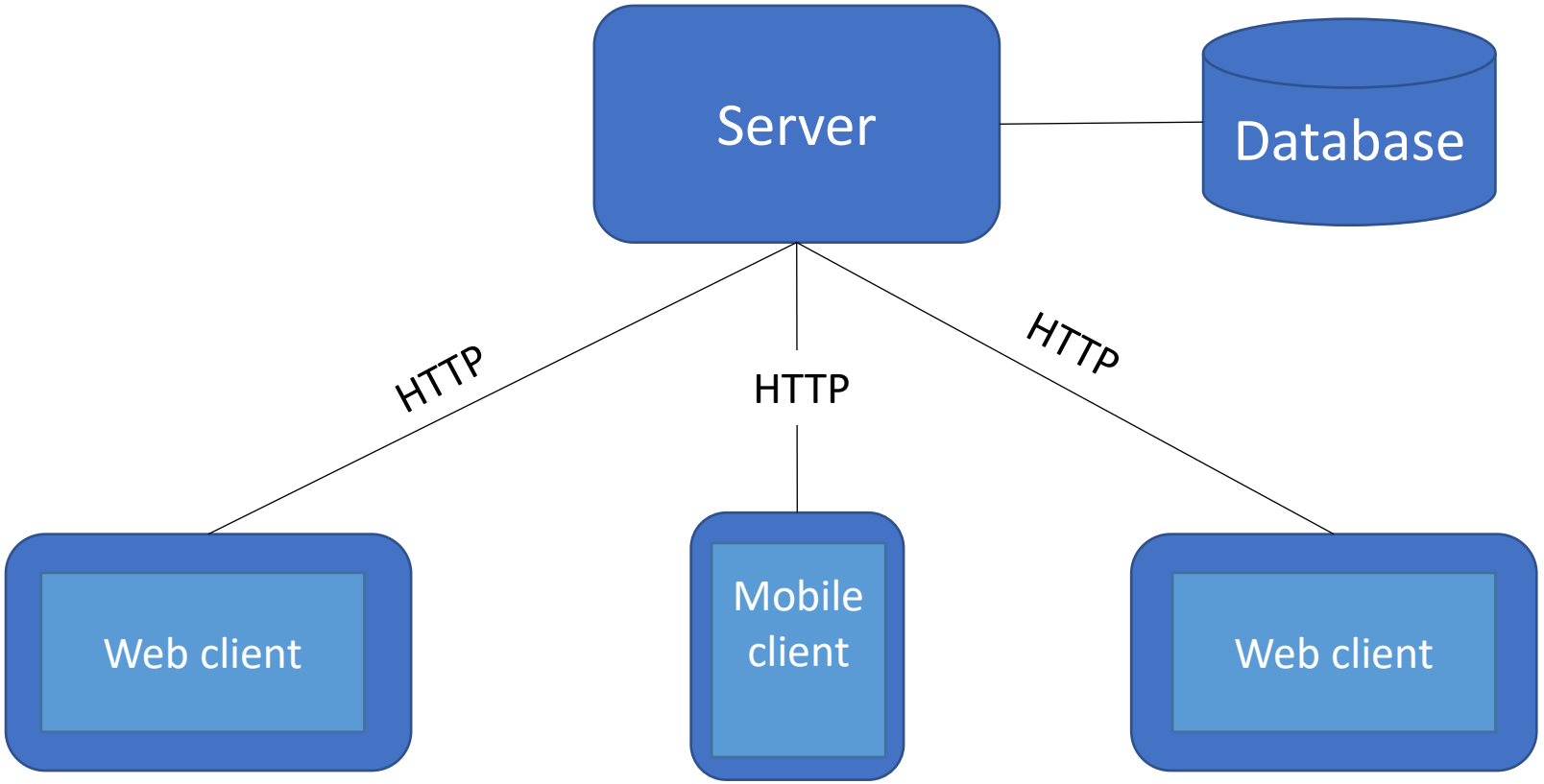
Lecture 10 – HTTP

Acknowledgement: w3schools.com, developer.mozilla.org, tutorialspoint.com

HTTP Overview

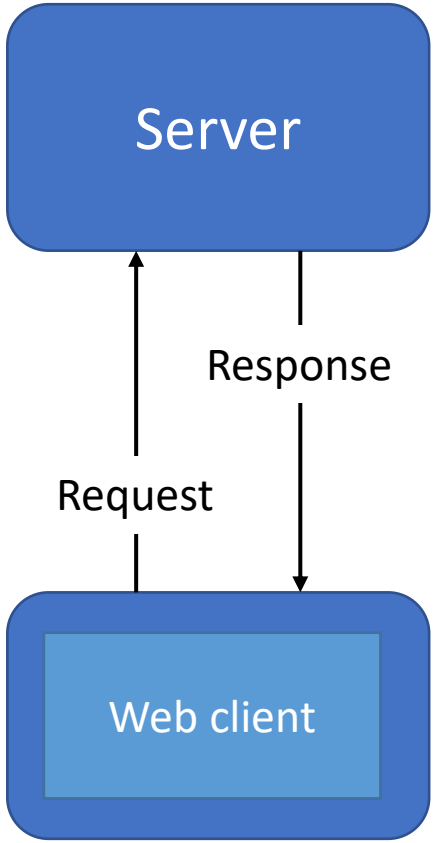
- HTTP stands for **H**yper **T**ext **T**ransfer **P**rotocol
- The protocol used for data exchange on the Web
- An application layer protocol that is sent over TCP, or over a TLS-encrypted TCP connection
- A client-server protocol where requests are made by a client and responded to by a server

HTTP Overview



HTTP Overview

- Client
  - Initiates the HTTP request
  - Interprets the response
  - Can be web browser, mobile app or other
- Server
  - Sends a HTTP response from a client request
  - Often a collection of servers (not just one)

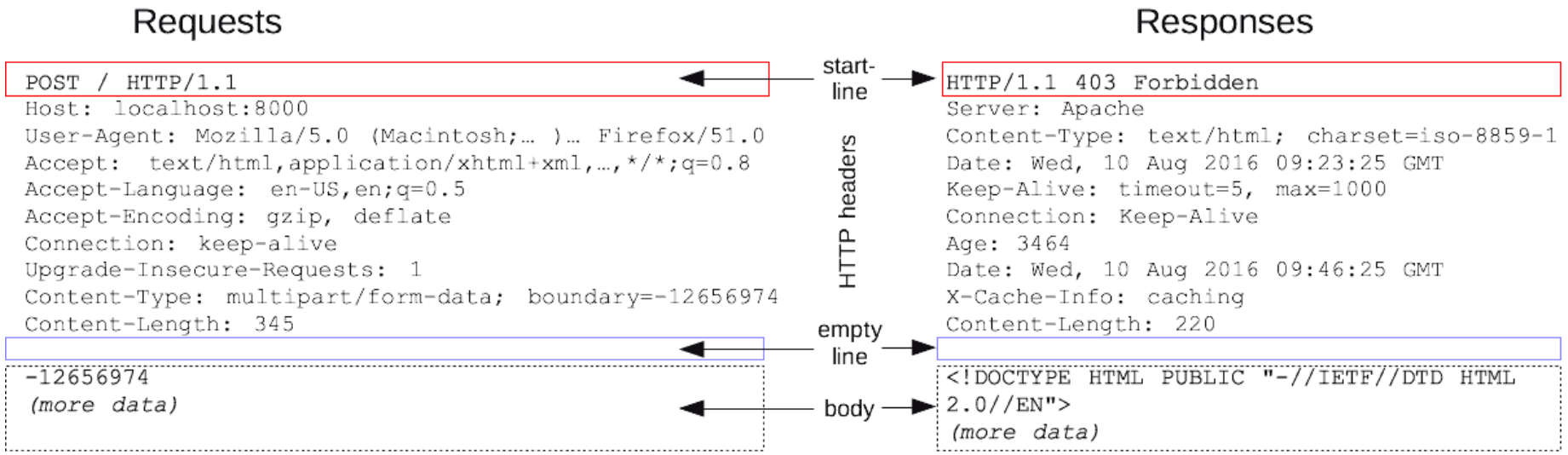


HTTP Overview

- HTTP is connectionless:
  - Client initiates HTTP request, disconnects from the server and waits for a response
  - The server processes the request and re-establishes the connection with the client to send response back
- HTTP is media independent:
  - Any type of data can be sent by HTTP
  - Content type is specified in the request header
- HTTP is stateless:
  - The server and client are aware of each other only during a current request, but then they forget about each other
  - Cookies allow for stateful sessions

HTTP Messages

- Requests and responses are sent via messages
- Messages include:
  - A start line (request or status)
  - Headers (zero or more)
  - An empty line
  - Optional message body



HTTP Requests

- Starts with the request line:
  - HTTP Method (GET, POST, PUT, DELETE, etc)
  - Request target, a URI (Uniform Resource Identifier)
  - HTTP Version

```
GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1
```

GET Method

- Used to retrieve information from a server
- The query string (name/value pairs) is sent in the URL of the request

```
GET /hello.html HTTP/1.1
```

```
GET /test/demo_form.php?name1=value1&name2=value2 HTTP/1.1
Host: w3schools.com
```

GET Method

- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions
- GET requests are only used to request data (not modify)

POST Method

- Used to send data to a server to **create a resource**
- The data sent to the server is stored in the body of the HTTP request
  - Body often sent as JSON

```
POST /test/demo_form.php HTTP/1.1
Host: w3schools.com
username=value1&password=value2
```

POST Method

- POST requests are never cached
- POST requests do not remain in the browser history
- POST requests cannot be bookmarked
- POST requests have no restrictions on data length
- POST requests are used when dealing with sensitive data

PUT Method

- Less common than post
- Used to send data to a server to **update a resource**
- The data sent to the server is stored in the body of the HTTP request

```
PUT /users/id HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2
```

DELETE Method

- Deletes a specified resource

```
DELETE /users/id HTTP/1.1
Host: w3schools.com
```

CRUD

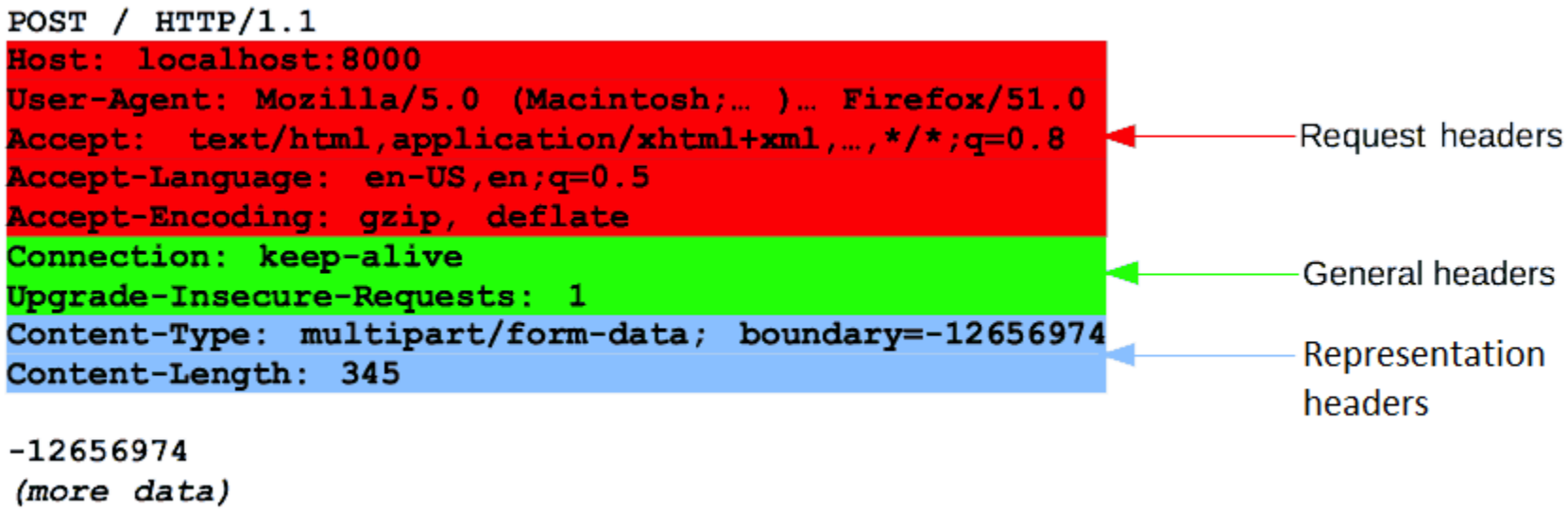
- The four basic operations of persistent storage
- Used in reference to databases or REST APIs
- CRUD maps to HTTP methods
  - Create - POST
  - Read - GET
  - Update - PUT
  - Delete - DELETE

Wake-up!

- https://youtu.be/Rq8A-tlJDew

HTTP Request Headers

- Provide required information about the request or response
- Request headers modify the request by specifying it further (like Accept-Language)
- General headers, like Connection, apply to the message as a whole
- Representation headers like Content-Type describe the original format of the message data and any encoding applied (only present if the message has a body)



HTTP Request Body

- Not all requests have a body
- GET, HEAD, DELETE, or OPTIONS, usually don't need one
- POST request may have a body if it contains HTML form data or other data to send in the request

HTTP Responses

- Starts with a status line which contains:
  - The protocol version, usually HTTP/1.1.
  - A status code, indicating success or failure of the request. Common status codes are 200, 404, or 302
  - A status text: A brief, purely informational, textual description of the status code to help a human understand the HTTP message

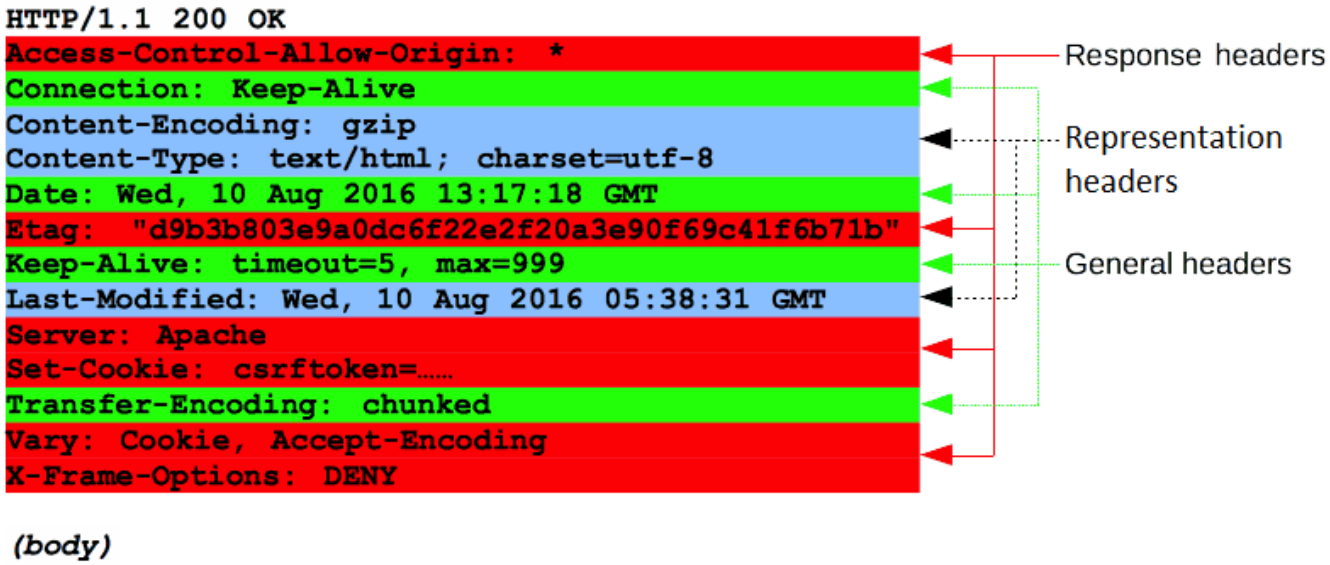
```
HTTP/1.1 404 Not Found
```

HTTP Status Codes

- 1xx: Informational
  - The request is received and continuing process (e.g. 100 Continue)
- 2xx: Success
  - The action was successfully received, understood, and accepted (e.g. 200 OK)
- 3xx: Redirection
  - Further action must be taken in order to complete the request (e.g. 302 Found)
- 4xx: Client Error
  - The request contains bad syntax or cannot be fulfilled (e.g. 404 Not Found)
- 5xx: Server Error
  - The server failed to fulfill an apparently valid request (e.g. 500 Internal Server Error)

HTTP Response Headers

- General headers, like Date, apply to the whole message
- Response headers, like Server, give additional information about the server which doesn't fit in the status line
- Representation headers like Content-Type that describe the original format of the message data and any encoding applied



HTTP Response Body

- Contains the requested information in the format specified by the Accept field in the request header
- Often HTML or JSON

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

HTTPS

- Takes care of secure communication between a client and server
- Handles sensitive data like login and credit card transactions
- Uses Secure Sockets Layer (SSL) or Transport Layer Security (TLS)
- The SSL protocol is used to encrypt data for secure data transmission
- The TLS protocol is a newer and more secure version of SSL

How this can fit into a web app (one example)