IT 311

Mobile Application Development 2

Learning objectives

At the end of this lesson, you should be able to:

01 Familiarize with JSON

02 JSON Syntax

03 HTTP Request and Response

DURATION: 3hrs

What is JSON?

JSON stands for JavaScript Object Notation

JSON is a lightweight format for storing and transporting data

JSON is often used when data is sent from a server to a web page

JSON is "self-describing" and easy to understand

Sample of JSON

```
{“student":[
    {
   “name":“jayson", “school":“Bulsu“, “age” : 22
    },
    {
        “name": “mark", “school":“LCUP“, “age” : 24
    },
    {
        “name": “josh", “school":“UST“, “age” : 25
    }
]
}
```

Sample of JSON

```
{
  "first_name": "Rajeev",
  "last_name": "Sharma",
  "email_address": "rajeev@ezeelive.com",
  "is_alive": true,
  "age": 30,
  "height_cm": 185.2,
  "billing_address": {
    "address": "502, Main Market, Evershine City, Evershine, Vasai East",
    "city": "Vasai Raod, Palghar",
    "state": "Maharashtra",
    "postal_code": "401208"
  },
  "shipping_address": {
    "address": "Ezeelive Technologies, A-4, Stattion Road, Oripada, Dahisar East",
    "city": "Mumbai",
    "state": "Maharashtra",
    "postal_code": "400058"
  },
  "phone_numbers": [
   {
   "type": "home",
   "number": "9975666694"
   },
   {
   "type": "office",
   "number": "9822117730"
   }
 ],
 "date_of_birth": null
}
```

ADVANTAGES of json

Less Verbose: JSON has a more compact style than XML, and it is often more readable. The lightweight approach of JSON can make significant improvements in RESTful APIs working with complex systems.

Faster: The XML software parsing process can take a long time. One reason for this problem is the DOM manipulation libraries that require more memory to handle large XML files. JSON uses less data overall, so you reduce the cost and increase the parsing speed.

ADVANTAGES of json

Readable: The JSON structure is straightforward and readable. You have an easier time mapping to domain objects, no matter what programming language you're working with.

Structure Matches the Data: JSON uses a map data structure rather than XML's tree. In some situations, key/value pairs can limit what you can do, but you get a predictable and easy-to-understand data model.

Objects Align in Code: JSON objects and code objects match, which is beneficial when quickly creating domain objects in dynamic languages.

DISADVANTAGES of json

While we saw the advantages, it is not devoid of disadvantages too. Following are the disadvantages:

1. It is not fully secure. “JSON is not fully secure as we think in the web world, Cross Site Request Forgery (CSRF) in JSON Array hack allow grab sensitive user data from an unsuspecting user.”
2. It is limited in terms of the supported datatypes.

JSON RULES

- Data is in key value pairs
- In JSON, keys must be strings, written with double quotes
- Commas are used to separate data
- Curly braces are used for objects
- Square brackets are used to declare an Array

JSON DATA

- String “Hello” “Jayson Batoon” “I”
- Numbers 10 50.2 -52
- Booleans true false
- Null null
- Arrays [1,2,3] [“Hello”, “Jayson”]
- Object {“key”:”value”} {“name”:”Jayson”}

NOTES

When receiving a data from a web server, the data is always a String JSON.parse() for the data to becomes a javaScript object.

When sending a data t a web server the data has to be a string. Convert a javaScript object into a string with JSON.stringyfy()

HTTP REQUEST METHODS

- What is HTTP?
- The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

- HTTP works as a request-response protocol between a client and server.

- Example: A client (browser) sends an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

HTTP REQUEST METHODS

1. GET
2. POST
3. PUT
4. HEAD
5. DELETE
6. PATCH
7. OPTIONS

HTTP METHODS

CONNECT

- The HTTP CONNECT method starts two-way communications with the requested resource. It can be used to open a tunnel.

For example, the CONNECT method can be used to access websites that use SSL (HTTPS). The client asks an HTTP Proxy server to tunnel the TCP connection to the desired destination. The server then proceeds to make the connection on behalf of the client. Once the connection has been established by the server, the Proxy server continues to proxy the TCP stream to and from the client.

CONNECT is a hop-by-hop method.

GET METHODS

GET is used to request data from a specified resource.
GET is one of the most common HTTP methods.
Note that the query string (name/value pairs) is sent in the URL of a GET request:

/test/demo_form.php?name1=value1&name2=value2

Some other notes on GET requests:
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 METHODS

POST is used to send data to a server to create/update a resource.
The data sent to the server with POST is stored in the request body of the HTTP request:

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

POST is one of the most common HTTP methods.

Some other notes on POST requests:

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

put METHODS

PUT is used to send data to a server to create/update a resource.

The difference between POST and PUT is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly have side effects of creating the same resource multiple times.

HEAD METHODS

HEAD is almost identical to GET, but without the response body.

In other words, if GET /users returns a list of users, then HEAD /users will make the same request but will not return the list of users.

HEAD requests are useful for checking what a GET request will return before actually making a GET request - like before downloading a large file or response body.

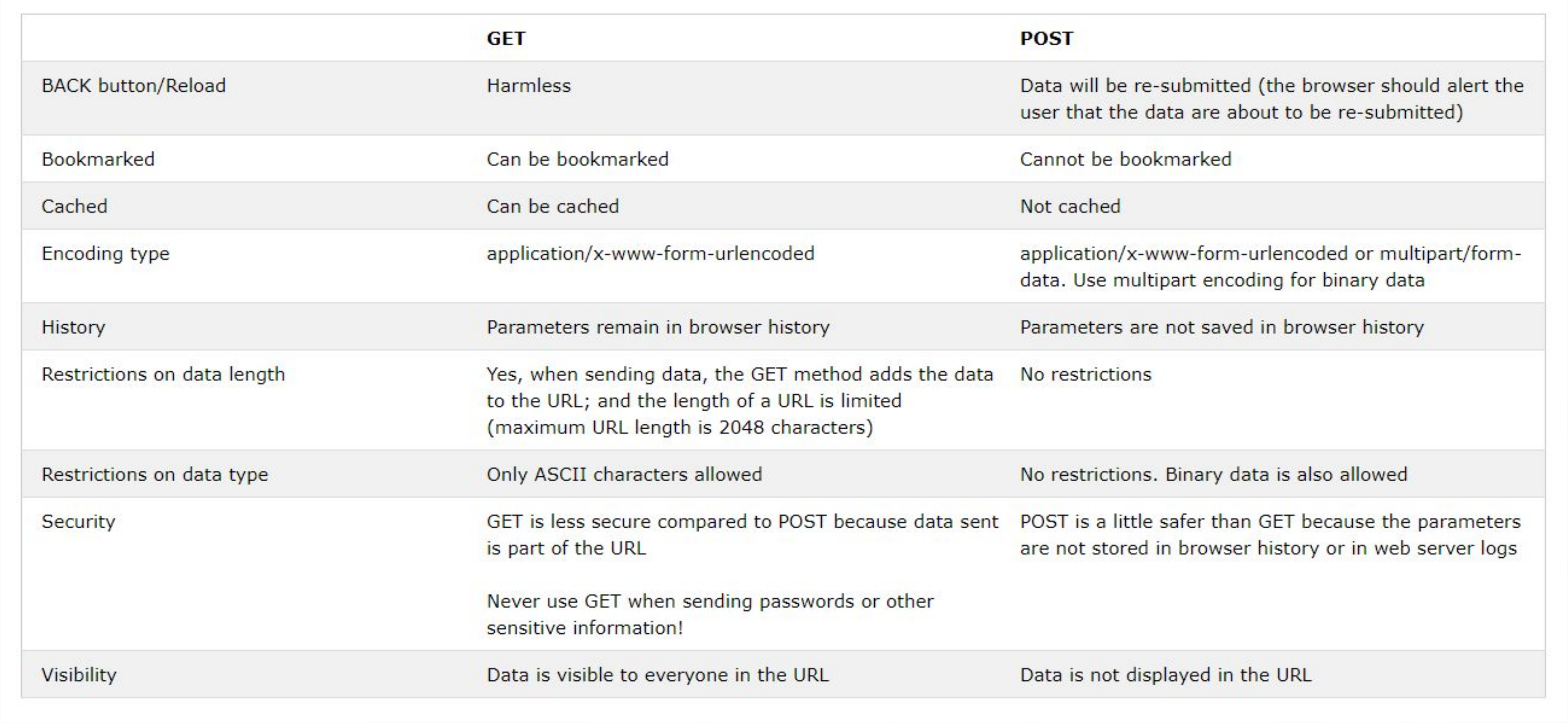
DELETE METHODS

The DELETE Method
The DELETE method deletes the specified resource.

OPTION METHODS

The OPTIONS method describes the communication options for the target resource.

GET vs. POST



	GET	**POST**
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

HTTP Responses

HTML Error Messages

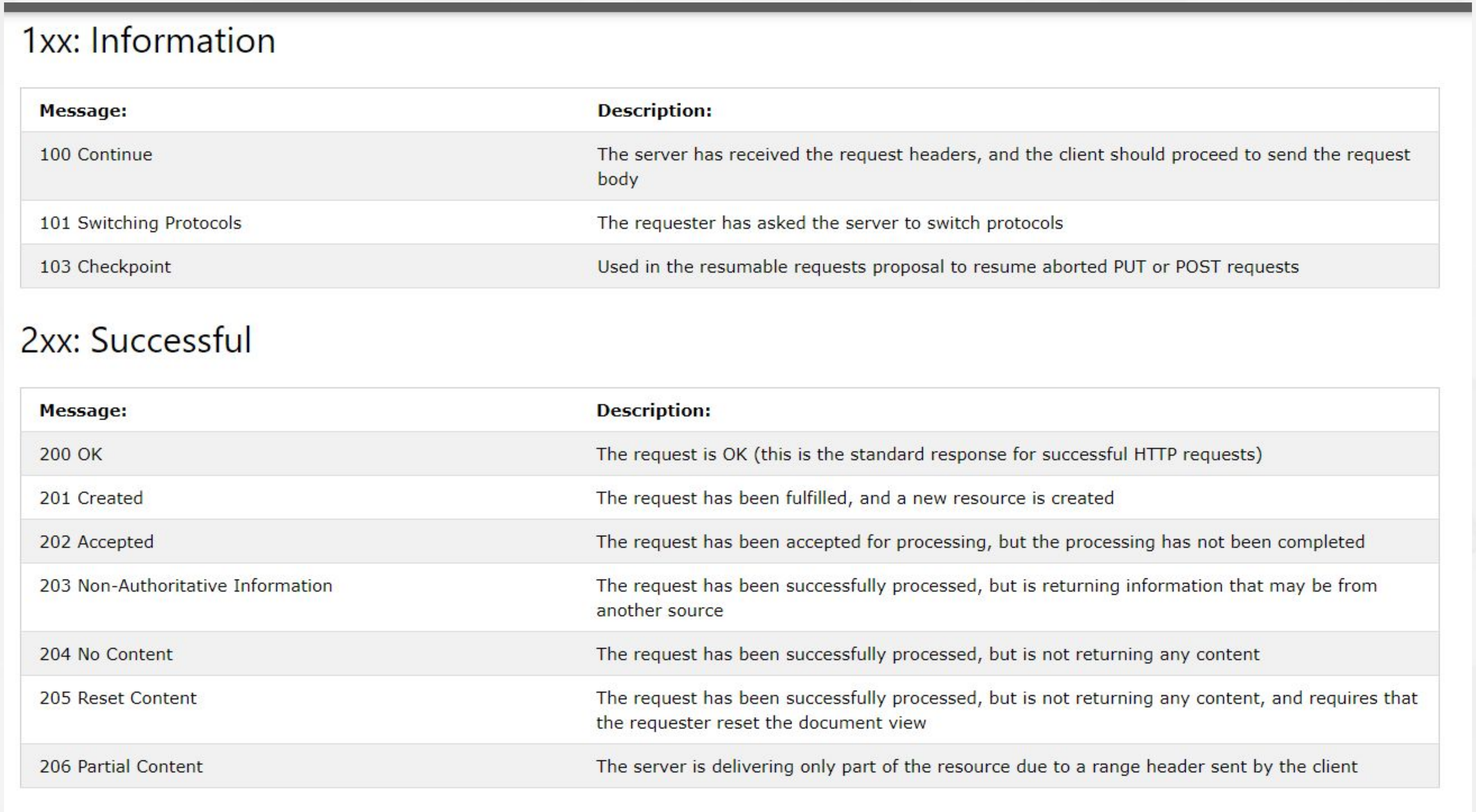
When a browser requests a service from a web server, an error might occur, and the server might return an error code like "404 Not Found".

It is common to name these errors HTML error messages.

But these messages are something called HTTP status messages. In fact, the server always returns a message for every request. The most common message is 200 OK.

Below is a list of HTTP status messages that might be returned:

HTTP Responses



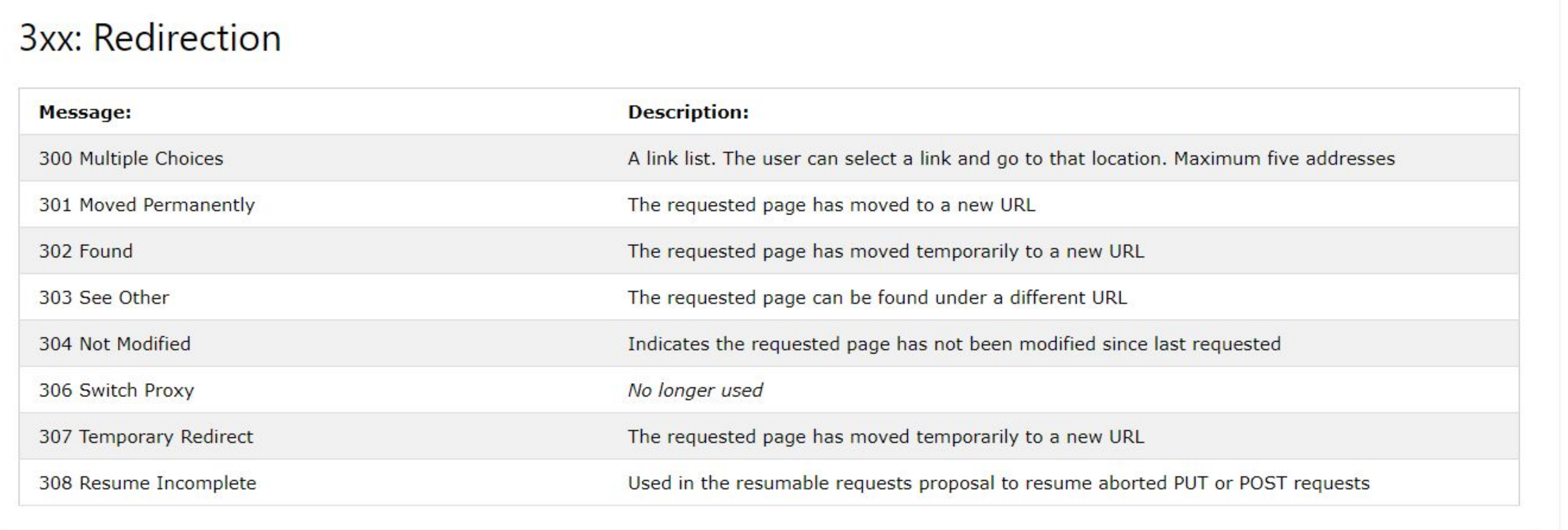
1xx: Information

Message:	Description:
100 Continue	The server has received the request headers, and the client should proceed to send the request body
101 Switching Protocols	The requester has asked the server to switch protocols
103 Checkpoint	Used in the resumable requests proposal to resume aborted PUT or POST requests

2xx: Successful

Message:	Description:
200 OK	The request is OK (this is the standard response for successful HTTP requests)
201 Created	The request has been fulfilled, and a new resource is created
202 Accepted	The request has been accepted for processing, but the processing has not been completed
203 Non-Authoritative Information	The request has been successfully processed, but is returning information that may be from another source
204 No Content	The request has been successfully processed, but is not returning any content
205 Reset Content	The request has been successfully processed, but is not returning any content, and requires that the requester reset the document view
206 Partial Content	The server is delivering only part of the resource due to a range header sent by the client

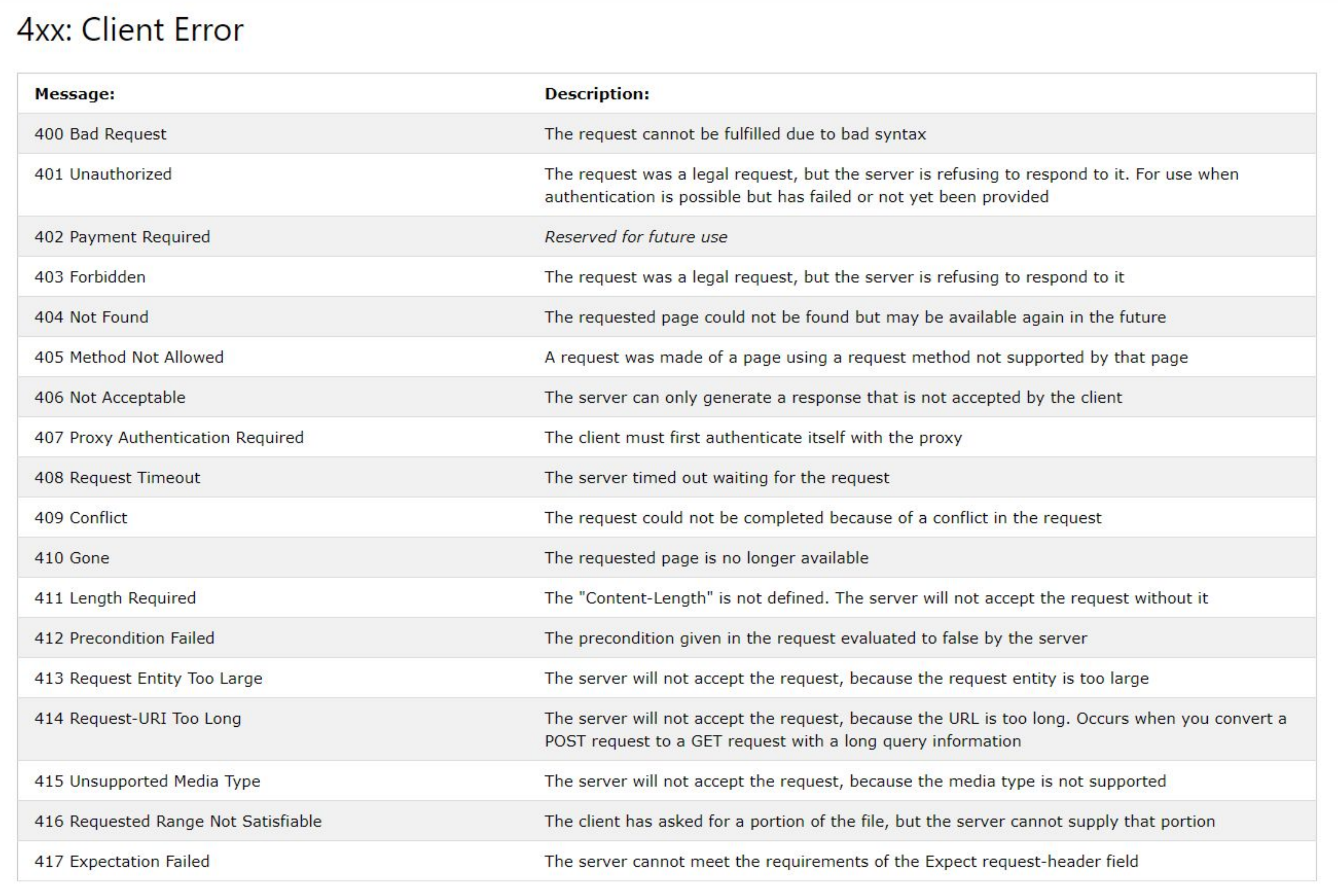
HTTP Responses



3xx: Redirection

Message:	Description:
300 Multiple Choices	A link list. The user can select a link and go to that location. Maximum five addresses
301 Moved Permanently	The requested page has moved to a new URL
302 Found	The requested page has moved temporarily to a new URL
303 See Other	The requested page can be found under a different URL
304 Not Modified	Indicates the requested page has not been modified since last requested
306 Switch Proxy	*No longer used*
307 Temporary Redirect	The requested page has moved temporarily to a new URL
308 Resume Incomplete	Used in the resumable requests proposal to resume aborted PUT or POST requests

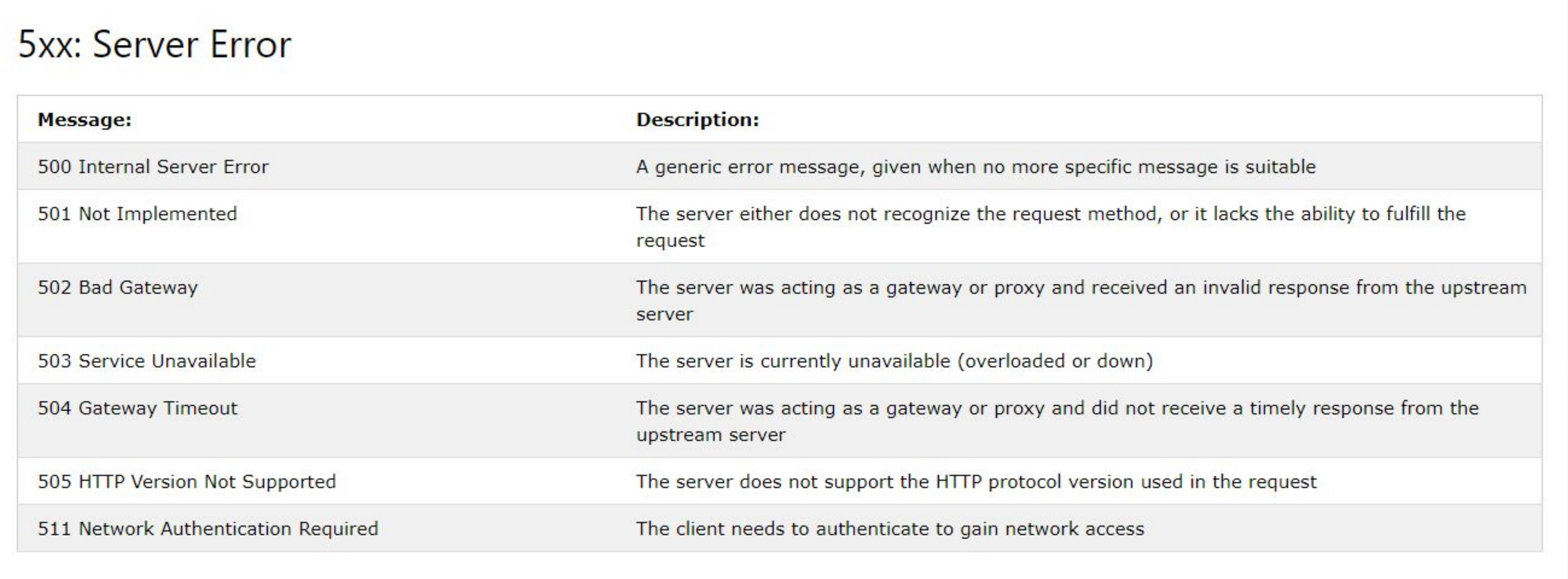
HTTP Responses



4xx: Client Error

Message:	Description:
400 Bad Request	The request cannot be fulfilled due to bad syntax
401 Unauthorized	The request was a legal request, but the server is refusing to respond to it. For use when authentication is possible but has failed or not yet been provided
402 Payment Required	*Reserved for future use*
403 Forbidden	The request was a legal request, but the server is refusing to respond to it
404 Not Found	The requested page could not be found but may be available again in the future
405 Method Not Allowed	A request was made of a page using a request method not supported by that page
406 Not Acceptable	The server can only generate a response that is not accepted by the client
407 Proxy Authentication Required	The client must first authenticate itself with the proxy
408 Request Timeout	The server timed out waiting for the request
409 Conflict	The request could not be completed because of a conflict in the request
410 Gone	The requested page is no longer available
411 Length Required	The "Content-Length" is not defined. The server will not accept the request without it
412 Precondition Failed	The precondition given in the request evaluated to false by the server
413 Request Entity Too Large	The server will not accept the request, because the request entity is too large
414 Request-URI Too Long	The server will not accept the request, because the URL is too long. Occurs when you convert a POST request to a GET request with a long query information
415 Unsupported Media Type	The server will not accept the request, because the media type is not supported
416 Requested Range Not Satisfiable	The client has asked for a portion of the file, but the server cannot supply that portion
417 Expectation Failed	The server cannot meet the requirements of the Expect request-header field

HTTP Responses



5xx: Server Error

Message:	Description:
500 Internal Server Error	A generic error message, given when no more specific message is suitable
501 Not Implemented	The server either does not recognize the request method, or it lacks the ability to fulfill the request
502 Bad Gateway	The server was acting as a gateway or proxy and received an invalid response from the upstream server
503 Service Unavailable	The server is currently unavailable (overloaded or down)
504 Gateway Timeout	The server was acting as a gateway or proxy and did not receive a timely response from the upstream server
505 HTTP Version Not Supported	The server does not support the HTTP protocol version used in the request
511 Network Authentication Required	The client needs to authenticate to gain network access

References

W3Schools. (2020) Retrieved from : https://www.w3schools.com/xml/xml_http.asp

Rajeev Sharma. (2021), Retrieved from : https://ezeelive.com/json-advantages-disadvantages/