DRQ Final Exam Notes

HTTP Status Codes Cheatsheet

Code Cat.	Code	Name	Meaning
2XX			**Success**
	200	OK	Request succeeded
	201	Created	Resource created
	204	No Content	Request succeeded, but no content to return
3XX			**Redirection**
	301	Moved Permanently	Resource moved permanently to new URL
	302	Found/Temporary Redirect	Resource temporarily located at diff URL
	304	Not Modified	Not modified e.g. resource has not changed since last request
4XX			**Client Errors**
	400	Bad Request	Server couldn't understand the request
	401	Unauthorized	Authentication required
	403	Forbidden	Client does not have access rights
	404	Not Found	Resource does not exist
	429	Too Many Requests	More requests than allowed by server sent in a given time
5XX			**Server Errors**
	500	Internal Server Error	Generic server error
	502	Bad Gateway	Invalid response from upstream server
	503	Service Unavailable	Temporarily unable to handle request
	504	Gateway Timeout	Upstream server did not respond in time

HTTP Methods Cheat Sheet

- GET
  - Retrieves data on the server, asking for info without changing anything
- POST

    - Sends data to the server to create something new, ex. submitting a form or uploading a file. Signups almost always use this.
- PUT
    - Replaces existing data on the server, or adds it if it doesn't exist. Updates an entire resource, sort of like pasting a duplicate file into a folder. In HTML, an example would be replacing your profile picture.
- PATCH
    - Partially replaces data on the server. Modifies only part of a resource, like saving a new version of a script from a code editor. HTML example: changing your email address without touching any other details
- HEAD
    - Similar to GET, but only requests the header, not the body. Allows for checking metadata such as file size.

Example Q & As

What are HTTP status codes used for?

HTTP status codes are three digit numbers returned with every HTTP response to indicate the outcome of the request. They assist browsers to understand what happened with the request, and if it requires redirection, was successful, failed etc. Where necessary they are also sometimes displayed to the end user, typically in the event of an error.

What are the most common 2XX status codes?

200 OK, 201 Created, 204 No Content.

What are the most common 3XX status codes?

301 Moved Permanently, 302 Found/Temporary Redirect, 304 Not Modified, 307 Temporary Redirect (similar to 302, but retains HTTP method used)

What is REST?

Short for **Re**presentational **S**tate **T**ransfer, REST is an architectural style for designing networked (internet connected) applications. Key principles include:

- Uses exclusively HTTP methods ex. GET for reading, POST for creating
- Is stateless (each request contains all information needed to understand the intent)

- Exposes directory structure-like URLs (resource-based)
- Transfers data in XML or JSON (typically JSON nowadays). This makes APIs predictable and resource-oriented. Ex. `/books/15` would contain info for a book with an ID of 15.

What does stateless mean in an HTTP context?

Stateless means that each request made to a server must include all information necessary to understand and process the request being made. The server doesn't store session information between requests, reducing memory usage and increasing performance and consistency. Each request is independent and complete in itself.

What is the structure of an HTTP request?

1. Line 1 (Request line)
    1. Method (GET, POST)
    2. URL to which request is being made
    3. HTTP version (Ex. HTTP/1.1, HTTP/2.0)
2. Line 2 (Headers)
    1. Could include many things
        1. Set Cookie
        2. Accept (specify MIME types for desired content)
        3. User Agent
        4. Server
3. Empty line
4. Body (optional)
    1. This is pretty much only used when data is being sent, such as in a form submission

What is an HTTP API?

An HTTP API (Application Programming Interface) is a set of rules and endpoints that allow different applications to communicate over HTTP. It typically defines:

- Available endpoints (URLs pointing to specific API-controlled resources)
- Supported methods
- Expected request/response formats
- Auth methods

- Data structures

What is the difference between GET and PUT in terms of sending data?

GET sends its data through URL parameters, e.g. `GET /search?query=movies&page=5`. It is visible in the URL (like in a Google search) and is usually cached by browsers. PUT, on the other hand, sends its data in the request body. It is not visible in the URL or cached, and is used when the data is larger, or of a more sensitive nature (such as the input into a submitted form).

What is AJAX programming?

AJAX is short for **A**synchronous **J**avascript **A**nd **X**ML. It is a technique for:

- Making HTTP requests via JS without a page reload
- Updating select parts of a webpage dynamically, only as needed
- Creating interactive webapps using the aforementioned functionality.
- Modern frameworks often use `fetch()` or Axios instead of the more barebones `XMLHttpRequest` function.

What is the difference between XML and JSON?

Both are data formats often used to transmit information over HTTP. XML displays information using a tag-based structure, similar to HTML (both were developed near-simultaneously and are both 'application profiles' of *SGML*, or **S**tandardised **G**eneral **M**arkup **L**anguage.)
An example of XML:

```
<movie>
        <title>Star Wars</title>
        <director>George Lucas</director>
        <releaseyear>1977</releaseyear>
</movie>
```

An example of JSON:

```
{
        "title": "Star Wars",
        "director": "George Lucas",
```

```
        "releaseyear": 1977
}
```

What is the structure of JSON?

JSON (JavaScript Object Notation) stores information as objects (like the curly braces shown above). Data types include strings, numbers, bools, null, arrays, and objects. It uses square brackets (`[1, 2, 3]`) to denote arrays. Functions and undefined values are not allowed, and double quotes *must* be used for strings and keys.

How do JSON.parse and JSON.stringify work?

JSON.parse() takes in JSON in the form of a string, and converts it to a JavaScript object.

```
const obj = JSON.parse('{"name": "John"}');
// obj is now {name: "John"}
```

JSON.stringify() does essentially the reverse, taking in a JS object and converting it to a JSON string:

```
const str = JSON.stringify({name: "John"});
// str is now '{"name":"John"}'
```

What are cookies?

Cookies are small pieces of data stored on the client's computer. They're transmitted:

- Server sets cookies using Set-Cookie header in response
- Browser automatically sends cookies back in Cookie header of subsequent requests
  For example:
  `Set-Cookie: sessionId=abc123; Expires=Wed, 09 Jun 2024 10:18:14 GMT`

What is an HTTP session?

A session is a way to maintain state between multiple HTTP requests. It typically uses a unique ID, stores session data on the server, allows tracking user state across requests, and has an expiration time.

How do shopping carts utilise cookies?

Shopping carts use cookies:

- Generating a unique cart ID when the user starts shopping
- Storing a card ID in a cookie
- Server uses ID to look up cart contents
- Cart persists across page visits/browser closes

Do cookies break statelessness?

No, they do not, because the cookie is transmitted with each request, meaning that all required information is included. Because the cookie is stored by the client, the server is not maintaining state between requests, and the cookie just helps the client send consistent identification.

What is the difference between static and dynamic web resources?

Static Resources:

- Files served exactly as stored (HTML, images, CSS)
- Same content for every user
- Faster to serve
- Can be cached easily
  Dynamic Resources:
- Generated on-demand
- Can be personalized per user
- Can reflect real-time data
- Requires server processing

What is an SPA (Single Page Application)?

An SPA is a web app that loads a single, initial HTML page, updating content dynamically using JavaScript (in an AJAX manner) after this initial load. It does not require full-page reloads, and often uses client-side routing (Ex. Gmail, Facebook, other modern web apps)

Why would your browser show raw HTML instead of the rendered version?

Your browser might show raw HTML instead of rendering it when:

1. Content-Type header is incorrect (e.g., "text/plain" instead of "text/html")
2. Server configuration issues
3. File extension problems
4. Browser plugin/setting forcing source view
5. Malformed HTML causing browser to fail parsing

That covers all 21 questions! Would you like me to elaborate on any particular topics? Also, since this is for an exam, I can help you practice by:

1. Asking you some of these questions back in different ways
2. Creating some scenario-based questions
3. Helping you identify common trick questions or pitfalls