Your Quick Guide to HTTP

Aram Tchekrekjian

@AramT87

Topics

- HTTP
- REST & RESTful Services
- HTTP Request Methods
- Content-Type
- HTTP Headers
- HTTP Statuses
- HTTP Cookies
- HTTPS
- HTTP/2
- HTTP/3

HTTP

- Hyper-Text Transfer Protocol, is the communication protocol on the web that is used to transmit data
- Foundation of the Web
- Simple: Human readable format
- Extensible using the Headers to send/receive extra information
- Stateless, doesn’t maintain state unless HTTP Cookies are introduced to hold the communication session or state

REST & RESTful Services

- Representational State Transfer or REST is an architectural style to build Programming Interfaces (APIs) for data manipulation through HTTP

- RESTful Services are the web services built using the REST style

- Hosted under domain endpoints

- Allows clients to communicate and access resources

- Uses HTTP as the communication protocol

Request Methods

- **GET**: Used to retrieve data, any parameter should be passed via the query string

- **POST**: Used to submit data within the request body, this is usually used to pass personal or confidential data

- **PUT**: Used to edit record in resource server without creating new record

- **DELETE**: Used to delete a record in server

- Other Methods include: PATCH, OPTIONS, TRACE, HEAD, TUNNEL

Content Types

- **plain:** Data will be sent ‘as-is’ in plain text without any serialization, encryption or encoding.

- **json:** Data will be serialized in JSON format when sent from POST or PUT request body

- **form-url-encoded:** This is represented as a key-value pair (dictionary) of request parameters that are sent as part of the request body. Use when sending small amounts of data

- **form-data:** Used when uploading form fields that include file upload, it uploads the data in multiple parts. Use it when sending (binary) or large payloads

HTTP Headers

- A collection of key,value pairs (or dictionary) of meta-data that can be passed with each request or response
- Headers are categorized by context:
    - **Request Headers**: such as Accept-Language, Authorization
    - **Response Headers**: such as Connection, Server
    - **Representation Headers**: such as content-type, content-language
    - **Payload Headers**: such as content-length, transfer-encoding

HTTP Statuses

- Http status represents the status of the RESTful service after HTTP Request is completed

- Status codes are represented as 3 digits, where the first digit represents the category:

- **1xx**: Request received and under processing

- **2xx**: Successful

- **3xx**: Redirection (action to be taken by browser or user)

- **4xx**: Invalid request by client, data incomplete or invalid

- **5xx**: Server-side error (API crash, misconfigurations, app pool shutdown)

@AramT87

HTTP Cookies

- Used to maintain the state or session between multiple HTTP communications

- A Cookie is a small piece of data passed from server to user's browser via the set-cookie response header

- Cookies can also be restricted via the use of Secure attribute and HttpOnly Attribute, this is used to prevent Cross-site scripting attack (XSS)

- Cookies are used mainly for Session Management, Personalization and Tracking

HTTPS

- S stands for Secure, which means the HTTP communication between the client (like browser) and the website will happen via a secure channel, using an SSL/TLS encryption protocol
- TLS is the successor of SSL
- TLS v1.2 is the minimum recommended version of TLS that websites should use to maintain a secure website.
- TLS v1.3 is the latest version.
- Your site, including your web API must always use HTTPS

HTTP/2

- HTTP/2 is a major revision of the HTTP, introduced in 2015
- Its purpose is to improve the web performance by decreasing latency
- Over 97% of browsers now support HTTP/2
- Key Features Include: Multiplexing, Weighted Prioritization, Server Push, Headers Compression
- See the Next Slide for details

HTTP/2 - Key Features

- **Multiplexing**: Request and Response messages can be transmitted between client and server via bidirectional and concurrent streams (Over the same TCP Connection)

- **Weighted prioritization**: streams can be assigned weighted value and dependency for the client to display the responses from the streams accordingly

- **Sever Push**: When client requests a resource, the server can push extra resources to be cached on the client and used when needed

- **Headers Compression**: using HPACK specification to compress HTTP headers to optimize streams multiplexing

HTTP/3

- Though not officially announced, HTTP/3 is the 3rd major revision of the HTTP

- Introduces data transmission on a new transport protocol - Quic or (pronounced as Quick)

- Quick UDP Internet Connections or Quic relies on UDP protocol rather than TCP

- Quic on UDP provides a faster and more efficient communication than TCP which leads to improved web performance and user experience

- Over 75% of browsers now support HTTP/3, more and more sites have started adopting it

HTTP/3 - Key Features

- **Faster connection setup** and **reduced Round-Trip Time** by combining the cryptographic and transport handshakes

- With the use of **Connection IDs**, a communication can be maintained between client and server even when device's network switches to another

- Solves the **TCP head-of-line blocking issue**: If a packet is lost, the stream-aware Quic communication will know which stream is exactly loss and it will retransmit it

- Enhanced security with **transport-level default encryption**: which means connections will always be encrypted, which will include data and meta-data about the connection

Thank You

Follow me for more content

Aram Tchekrekjian

Microsoft® MVP Most Valuable Professional

AramT87

Get Free tips and Tutorials in .NET and C#

Join 700+ Readers

CodingSonata.com/newsletters

<Coding♪onata />