Introduction to IT

Internet

На русском для тех, кто хочет перейти в ИТ сферу

Two main reasons

The Internet started in the 1960s as a way for government researchers to share information. Computers in the '60s were large and immobile and in order to make use of information stored in any one computer, one had to either travel to the site of the computer or have magnetic computer tapes sent through the conventional postal system.

Another catalyst in the formation of the Internet was the heating up of the Cold War. The Soviet Union's launch of the Sputnik satellite spurred the U.S. Defense Department to consider ways information could still be disseminated even after a nuclear attack. This eventually led to the formation of the ARPANET (Advanced Research Projects Agency Network), the network that ultimately evolved into what we now know as the Internet. ARPANET was a great success but membership was limited to certain academic and research organizations who had contracts with the Defense Department. In response to this, other networks were created to provide information sharing.

Birthday of the Internet

January 1, 1983 is considered the official birthday of the Internet. Prior to this, the various computer networks did not have a standard way to communicate with each other. A new communications protocol was established called Transfer Control Protocol/Internetwork Protocol (TCP/IP). This allowed different kinds of computers on different networks to "talk" to each other. ARPANET and the Defense Data Network officially changed to the TCP/IP standard on January 1, 1983, hence the birth of the Internet. All networks could now be connected by a universal language.

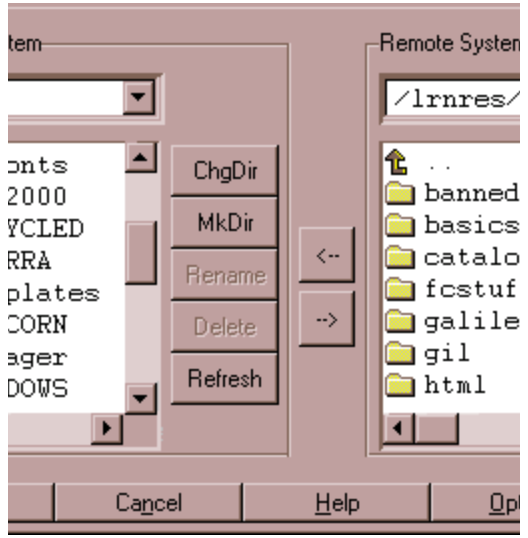
UNIVAC

The image above is a scale model of the UNIVAC I (the name stood for Universal Automatic Computer) which was delivered to the Census Bureau in 1951. It weighed some 16,000 pounds, used 5,000 vacuum tubes, and could perform about 1,000 calculations per second. It was the first American commercial computer, as well as the first computer designed for business use. (Business computers like the UNIVAC processed data more slowly than the IAS-type machines, but were designed for fast input and output.) The first few sales were to government agencies, the A.C. Nielsen Company, and the Prudential Insurance Company. The first UNIVAC for business applications was installed at the General Electric Appliance Division, to do payroll, in 1954. By 1957 Remington-Rand (which had purchased the Eckert-Mauchly Computer Corporation in 1950) had sold forty-six machines.

Types of Internet Protocols

When we think of the Internet we often think only of the World Wide Web. The Web is one of several ways to retrieve information from the Internet. These different types of Internet connections are known as protocols.

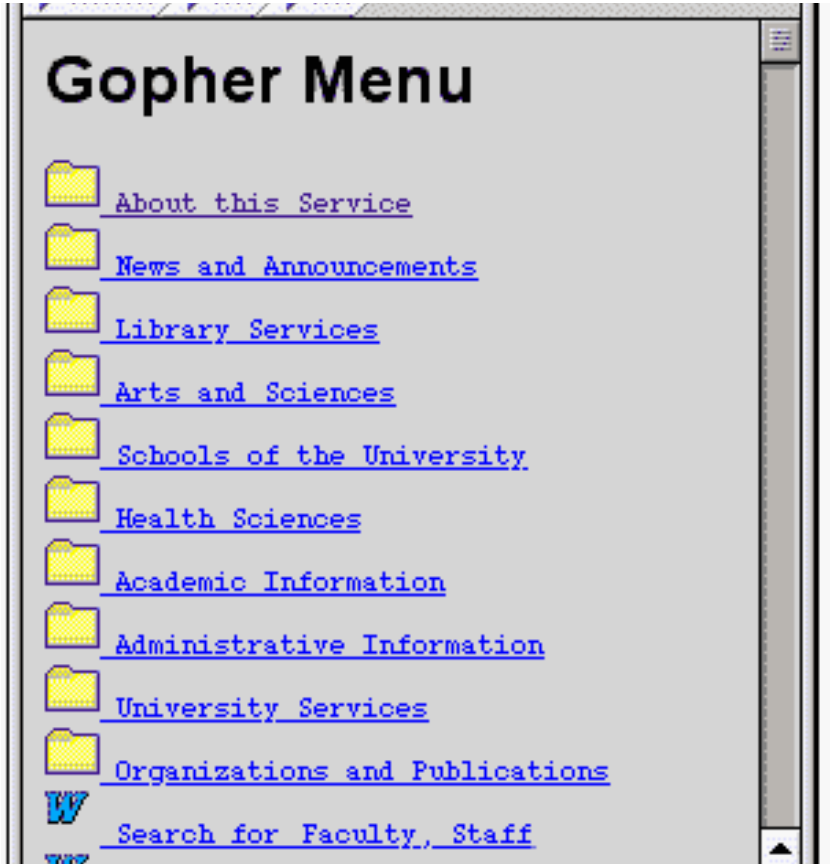
File retrieval protocols
This type of service was one of the earliest ways of retrieving information from computers connected to the Internet. You could view the names of the files stored on the serving computer, but you didn't have any type of graphics and sometimes no description of a file's content. You would need to have advanced knowledge of which files contained the information you sought.



FTP (File Transfer Protocol)
This was one of the first Internet services developed and it allows users to move files from one computer to another. Using the FTP program, a user can logon to a remote computer, browse through its files, and either download or upload files (if the remote computer allows). These can be any type of file, but the user is only allowed to see the file name; no description of the file content is included. You might encounter the FTP protocol if you try to download any software applications from the World Wide Web. Many sites that offer downloadable applications use the FTP protocol.

Types of Internet Protocols

- **Gopher**
  Gopher offers downloadable files with some content description to make it easier to find the file you need. The files are arranged on the remote computer in a hierarchical manner, much like the files on your computer's hard drive are arranged. This protocol isn't widely used anymore, but you can still find some operational gopher sites.
- **Telnet**
  You can connect to and use a remote computer program by using the telnet protocol. Generally you would telnet into a specific application housed on a serving computer that would allow you to use that application as if it were on your own computer. Again, using this protocol requires special software.



Communications Protocols

- **Email**
  This method of Internet communication has become the standard. A main computer acts as a "post office" by sending and receiving mail for those who have accounts. This mail can be retrieved through any number of email software applications (MS Outlook, Eudora, etc.) or from Web based email accounts (Yahoo, Hotmail). Email is an example of asynchronous Internet communication.

Email also provides the ability to access email lists. You can subscribe to an email list covering any number of topics or interests and will receive messages posted by other subscribers. Email communities evolve from interaction between subscribers who have similar interests or obsessions. Gmail, yahoo mail.

- **Usenet**
  Usenet is something like a bulletin board or an email list without the subscription. Anyone can post a message to or browse through a Usenet newsgroup. Usenet messages are retained on the serving computer only for a predetermined length of time and then are automatically deleted, whereas email list messages are retained on the serving computer until the account holder downloads them. Many email applications, as well as Web browsers, allow you to set up Usenet newsgroup accounts.
- **IRC (Internet Relay Chat)**
  This protocol allows for synchronous communication: users on different computers anywhere in the world can communicate in "real time" or simultaneously. You can instantly see a response to a typed message by several people at the same time. This protocol requires a special software application that can be downloaded from the Web, generally for free.

Browsers

Your transports around the World Wide Web

A browser is an application you use to view files on the World Wide Web. There are text or terminal-based browsers (such as Lynx) that allow you to view only the text of a file on the Web. Most browsers now are graphical browsers that can be used to view text, graphics, and other multimedia information.

There are many types of Web browsers available, but the most widely used are MS Internet Explorer and Netscape. Both claim to be better and faster than the other, but the choice of which one to use usually becomes a personal one. Because some Web pages are created for specific browsers, it can be important which browser you use. Web pages may look different when accessed by different browsers.

Microsoft Internet Explorer
Internet Explorer holds the lion's share of the browser usage today, but it came into the game later than its main competitor.

Netscape
Netscape was one of the first commercial browsers on the scene and dominated the browser market until Microsoft got serious about Internet Explorer. There are some Internet users who are fiercely loyal to Netscape and there are sites on the Web that are best viewed using Netscape.

Chrome Web browser from Google, **Firefox**, **Opera**, **Edge.**

Uniform Resource Locator (URL)

The Uniform Resource Locator or URL is the "address" of a computer connected to the Internet. While surfing the Web, you'll notice that there is an address or location box at the top of your browser. It's here that you'll see an individual site's address displayed. This address allows you to find the site again, should you forget to bookmark it. You can simply type the URL into the address box, press the Enter key on your computer keyboard and you'll be taken back to the site of the address.The general format of a Web address is as follows: **http://www.whitehouse.gov**

The http in the address stands for "hypertext transfer protocol", the protocol for the World Wide Web, and it tells your browser to look for a site on the Web. A URL could also appear as:

ftp://12.456.789 or **gopher://gopher.uzxy.edu**

The first part of the URL before the colon tells the browser what type of protocol to use. The colon and two forward slashes are standard to all URLs. Commonly, the letters WWW (World Wide Web) appear after the two forward slashes in many Web addresses, but other letters are also used.

After the first dot, or period, in the URL, is the name of the particular computer followed by another dot and what is known as the domain (.com, .edu, .gov, etc.). The domain indicates the type of group or organization using the address. For instance, all educational institutions have a URL that ends with the domain of .edu.

Internet Search Services

Loosely organizing the 'net

The vast amount of information available on the Internet can be dizzying. Some authorities estimate the number of documents on the Internet to be in the range of 800 million. Others say the number is unknowable. Fortunately, there are tools available that will sort through the mass of information: search engines or search directories.

Search engines collect information from Web sites and then, more or less, just dump that information into a database. There's more information to choose from in a search engine, but it's more difficult to retrieve relevant information.

Search directories try to impose some sense of order on the information they collect and you're more likely to find information relevant to your research topic, but they don't offer the massive amounts of information that you would find with a search engine. The sites collected are viewed by humans who make decisions about what subject categories the sites might fit into.

Search engines

Search engines are really just massive databases in which information from Internet documents are stored. The information in these databases is collected using a computer program (called a "spider" or a "robot") that scans the Internet and gathers information about individual documents. These special programs work automatically to find documents or they are asked by a creator of a Web site to visit the site to be included in a database.

When you do a search in a search engine, the order in which the results are listed also varies between search engines. Many search engines list the results using relevance ranking. Factors such as:

how often your search terms are on the Web page;

where they are located on the page; and,

how many other Web pages link to the page

...influence how high on the list of hits a page is listed. Many search engines allow Web sites to pay to have their pages listed higher in the results.

There are hundreds of these search engines available on the Web, but they all work in unique ways to collect and organize the information found. The information from Web sites might be gathered from all the words in a site, just the first few sentences in the body of a site, or only from the title or metatags (hidden descriptors of a site's content). Different search engines collect different information, that's why you'll get different results from the same search from different search engines.

"*dot com*" "*dot gov*" — Domain suffix

The term "dot.com" has become a ubiquitous phrase in the English language. The "dot.com" really refers to the domain of a Web site. Sites on the Web are grouped by their URLs according to the type of organization providing the information on the site. For example, any commercial enterprise or corporation that has a Web site will have a domain suffix of .com, which means it is a commercial entity.

The domain suffix provides you with a clue about the purpose or audience of a Web site. The domain suffix might also give you a clue about the geographic origin of a Web site. Many sites from the United Kingdom will have a domain suffix of .uk.

Here follows a list of the most common domain suffixes and the types of organizations that would use them.

.com Commercial site. The information provided by commercial interests is generally going to shed a positive light on the product it promotes.

.edu
Educational institution. Sites using this domain name are schools ranging from kindergarten to higher education.

.gov
Government. If you come across a site with this domain, then you're viewing a federal government site.

Country domain suffixes

.au	Australia
.in	India
.br	Brazil
.it	Italy
.ca	Canada
.mx	Mexico
.fr	France
.tw	Taiwan
.il	Israel
.uk	United Kingdom

URL vs DNS

URL stands for Uniform Resource Locator. URL is the address of the website which you can find in the address bar of your web browser. It is a reference to a resource on the internet, be it images, hypertext pages, audio/video files, etc.

Example : https://itexpertschool.com/

DNS is short for Domain Name System. Like a phonebook, DNS maintains and maps the name of the website, i.e. URL, and particular IP address it links to. Every URL on the internet has a unique IP address which is of the computer which hosts the server of the website requested.

Example: itexpertschool.com (ip address `44.240.168.34`)

The major difference between both is that the **URL is a complete address**. URL tells about the method through which information should exchange, the path after reaching that website. Whereas the **domain name is part of a URL**.

Domain name in a URL

- At it’s most basic, it’s composed of a top-level domain (for example, .com), plus a second-level domain (in this case, “example”). But often, it will also include a subdomain, most commonly www. Though it could also be “shop,” “help,” “docs,” “news,” etc.
- This tells your browser where to look in the DNS system for the webpage (or other resource) you’re looking for (but more on that later).

Path in a URL

The path is the part of the URL after the domain name. It tells the server corresponding to the domain name what folder and filename you're looking for.

http://itexpertschool.com/qa-for-beginners-101-russian-edition/

qa-for-beginners-101-russian-edition - path

Protocol in a URL

- Before the domain name, there will be a protocol. Usually, this is either http or https. This tells your browser the protocol to use to establish communication with a server.

http://itexpertschool.com

http is a protocol

Domain name system

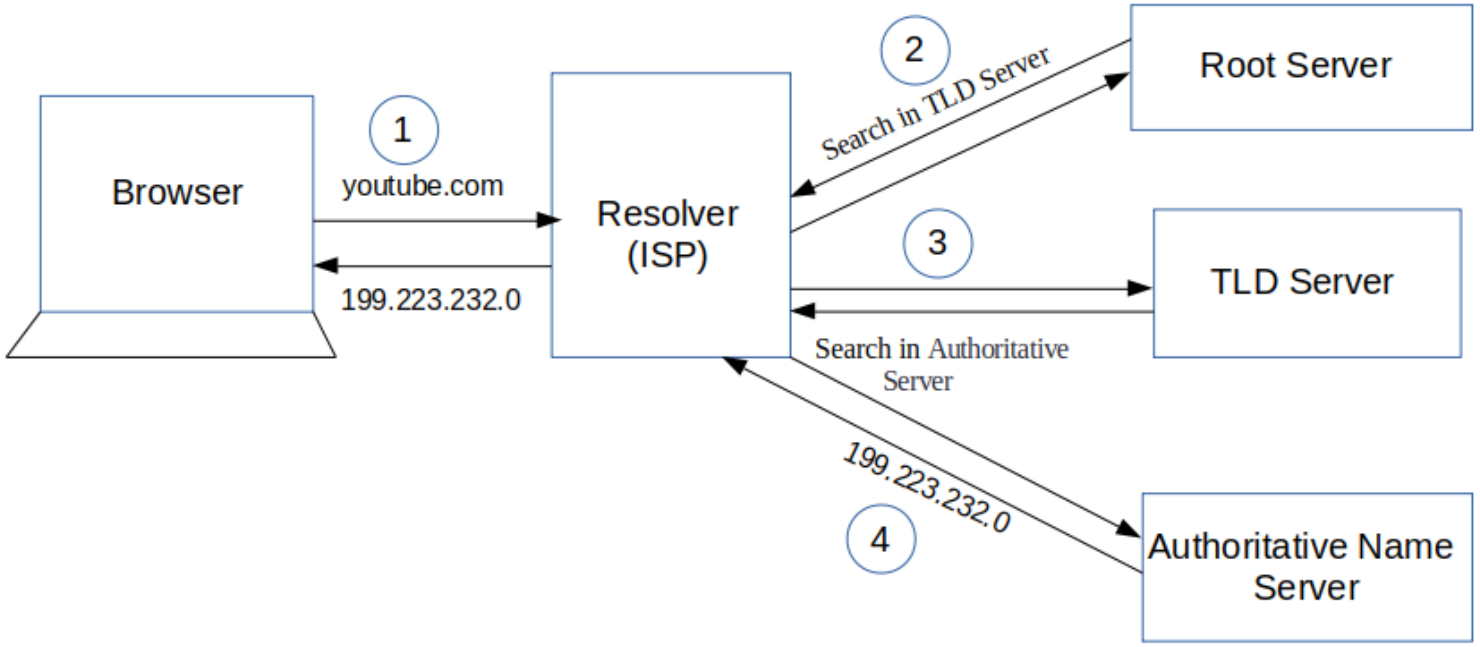
- You can relate it to your contact list. You can’t remember every person’s number, but you can remember their name. Same concept applies here as well. You **can’t remember** those scary **IP addresses**, but you can easily **remember domain names**.
- This huge amount of data is maintained in a database where the domain name with its IP address is stored. A system that stores domain names with its corresponding IP address is known as **DNS (Domain name system)** (I believe you must have heard about it).

Steps for what happens when we enter a URL :

1. Browser checks cache for DNS entry to find the corresponding IP address of website.
It looks for following cache. If not found in one, then continues checking to the next until found.
    1. Browser Cache
    2. Operating Systems Cache
    3. Router Cache
    4. ISP Cache
1. If not found in cache, ISP's (Internet Service Provider) DNS server initiates a DNS query to find IP address of server that hosts the domain name.
   The requests are sent using small data packets that contain information content of request and IP address it is destined for.
2. Browser initiates a TCP (Transfer Control Protocol) connection with the server using synchronize(SYN) and acknowledge(ACK) messages.
3. Browser sends an HTTP request to the web server. GET or POST request.
4. Server on the host computer handles that request and sends back a response. It assembles a response in some format like JSON, XML and HTML.
5. Server sends out an HTTP response along with the status of response.
6. Browser displays HTML content

DNS Lookup

- After hitting the URL, the first thing that needs to happen is to resolve IP address associated with the domain name. DNS helps in resolving this. **DNS is like a phone book** and **helps us to provide the IP address** that is associated with the domain name just like our phone book gives a mobile number which is associated with the person's name.



Steps

1. After hitting the URL, the **browser cache** is checked. As browser maintains its DNS records for some amount of time for the websites you have visited earlier. Hence, firstly, DNS query runs here to find the IP address associated with the domain name.

2. The second place where DNS query runs in **OS cache** followed by **router cache**.

3. If in the above steps, a DNS query does not get resolved, then it takes the help of resolver server. Resolver server is nothing but your ISP (Internet service provider). The query is sent to ISP where DNS query runs in **ISP cache.**

4. If in 3rd steps as well, no results found, then request sends to **top or root server** of the DNS hierarchy. There it never happens that it says no results found, but actually it tells, from where this information you can get. If you are searching IP address of the top level domain (.com,.net,.Gov,. org). It tells the resolver server to search **TLD server** (Top level domain).

Last steps

5. Now, resolver asks TLD server to give IP address of our domain name. TLD stores address information of domain name. It tells the resolver to ask it to **Authoritative Name server.**

6. The authoritative name server is responsible for knowing everything about the domain name. Finally, resolver (ISP) gets the IP address associated with the domain name and sends it back to the browser.

After getting an IP address, resolver stores it in its cache so that next time, if the same query comes then it does not have to go to all these steps again. It can now provide IP address from their cache.

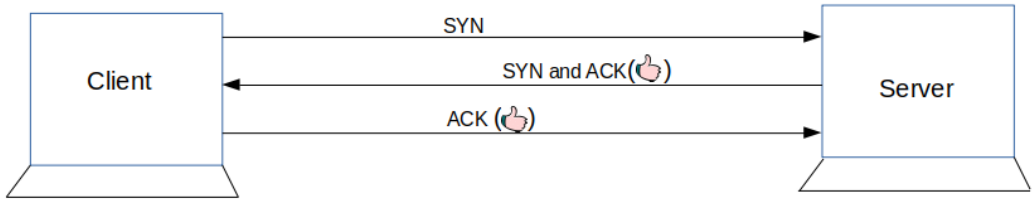
TCP connection initiates with the server by Browser

Once the **IP address** of the computer (where your website information is there) is **found**, it **initiates connection** with it. To communicate over the network, **internet protocol** is followed. **TCP/IP** is most common protocol. A connection is built between two using a process called **'TCP 3-way handshake'**. Let's understand the process in brief:

1. A client computer sends a **SYN message** means, whether second computer is open for new connection or not.

2. Then **another computer**, if open for new connection, it sends **acknowledge message** with SYN message as well.

3. After this, **first computer** receives its message and acknowledge by **sending** an **ACK message.**



Communication Starts (Request Response Process)

Finally, the connection is built between client and server. Now, they both can communicate with each other and share information. After successful connection, **browser (client)** sends a **request** to a **server** that I want this content. The server knows everything of what response it should send for every request. Hence, the **server responds back.** This response contains every information that you requested like web page, status-code, cache-control, etc. Now, the browser renders the content that has been requested. Content usually is HTML page.

What is HTML?

```
<!DOCTYPE html>
<html lang="en">

<meta charset="utf-8">
<title>Page Title</title>

<body>
  <h1>This is a Heading</h1>
  <p>This is a paragraph.</p>
  <p>This is another paragraph.</p>
</body>

</html>
```

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML is the **standard markup** language for Web pages
- HTML **elements** are the building blocks of HTML pages
- HTML elements are represented by **<> tags**

HTML Document Structure

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

HTTP Request / Response

- Communication between clients and servers is done by **requests** and **responses**:

1. A client (a browser) sends an **HTTP request** to the web
2. A web server receives the request
3. The server runs an application to process the request
4. The server returns an **HTTP response** (output) to the browser
5. The client (the browser) receives the response

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.

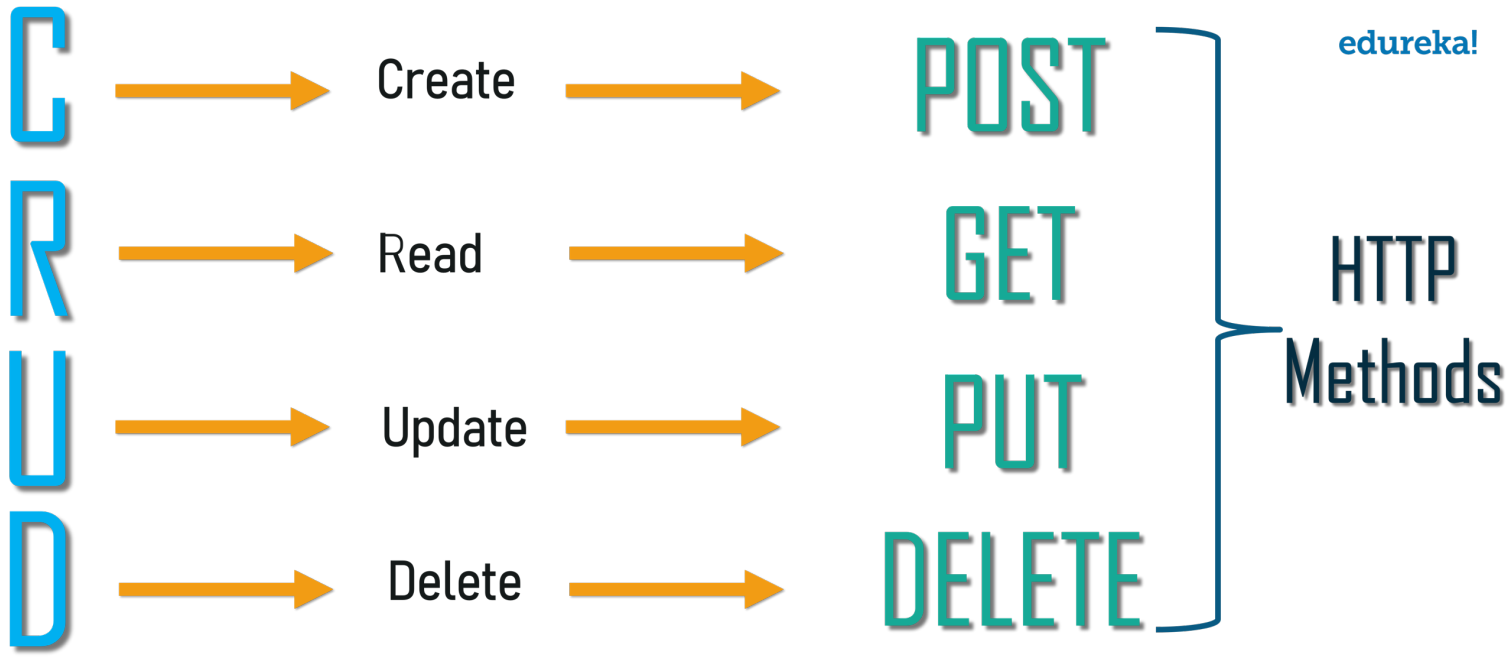
Example: A client (browser) sends an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

HTTP Methods

- **GET**
- **POST**
- **PUT**
- **HEAD**
- **DELETE**
- **PATCH**
- **OPTIONS**
- **CONNECT**
- **TRACE**

The two most common HTTP methods are: GET and POST.

Usage of HTTP methods



The GET Method

GET is used to request data from a specified resource.

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 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)
- Browsers use GET method

The POST Method

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: itexpertschoo.com

name1=value1&name2=value2
```

Some 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
- You could send post request use special tool like Postmen

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 Status 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.

Status 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 Early Hints	Used with the Link header to allow the browser to start preloading resources while the server prepares a response

Status 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

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
307 Temporary Redirect	The requested page has moved temporarily to a new URL
308 Permanent Redirect	The requested page has moved permanently to a new URL

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

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

REST

REST (**Representational State Transfer**) API is a software architectural style that determines how web services communicate with each other through HyperText Transfer Protocol.

- The REST stands for **REpresentational State Transfer**
- **State** means data
- **REpresentational** means formats (such as XML, JSON, YAML, HTML, etc)
- **Transfer** means carrying data between consumer and provider using the HTTP protocol

A REST API is an intermediary Application Programming Interface that enables two applications to communicate with each other over HTTP, much like how servers communicate to browsers.

The REST architectural style has quickly become very popular over the world for designing and architecting applications that can communicate

Server could return not only HTML page

Rest API end point

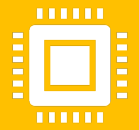
1. **An Endpoint URL**. An application implementing a RESTful API will define one or more URL endpoints with a domain, port, path, and/or query string — for example, https://mydomain/user/123?format=json.
2. **The HTTP method**. Differing HTTP methods can be used on any endpoint which map to application create, read, update, and delete (CRUD) operations

HTTP method	CRUD	Action
GET	read	returns requested data
POST	create	creates a new record
PUT or PATCH	update	updates an existing record

REST APIs and the Rest

Various data communication standards have evolved over the years. You may have encountered options including CORBA, SOAP, or XML-RPC. Most established strict messaging rules.

REST was defined in 2000 by Roy Fielding and is considerably simpler than the others. It's not a standard but a set of recommendations and constraints for RESTful web services.



Client-Server: System A makes an HTTP request to a URL hosted by System B, which returns a response. It's identical to how a browser works. A browser makes a request for a specific URL. The request is routed to a web server which typically returns an HTML page. That page may contain references to images, style sheets, and JavaScript, which incur further requests and responses. Rest API could return any type of data (JSON, XML, text)

Rest API examples

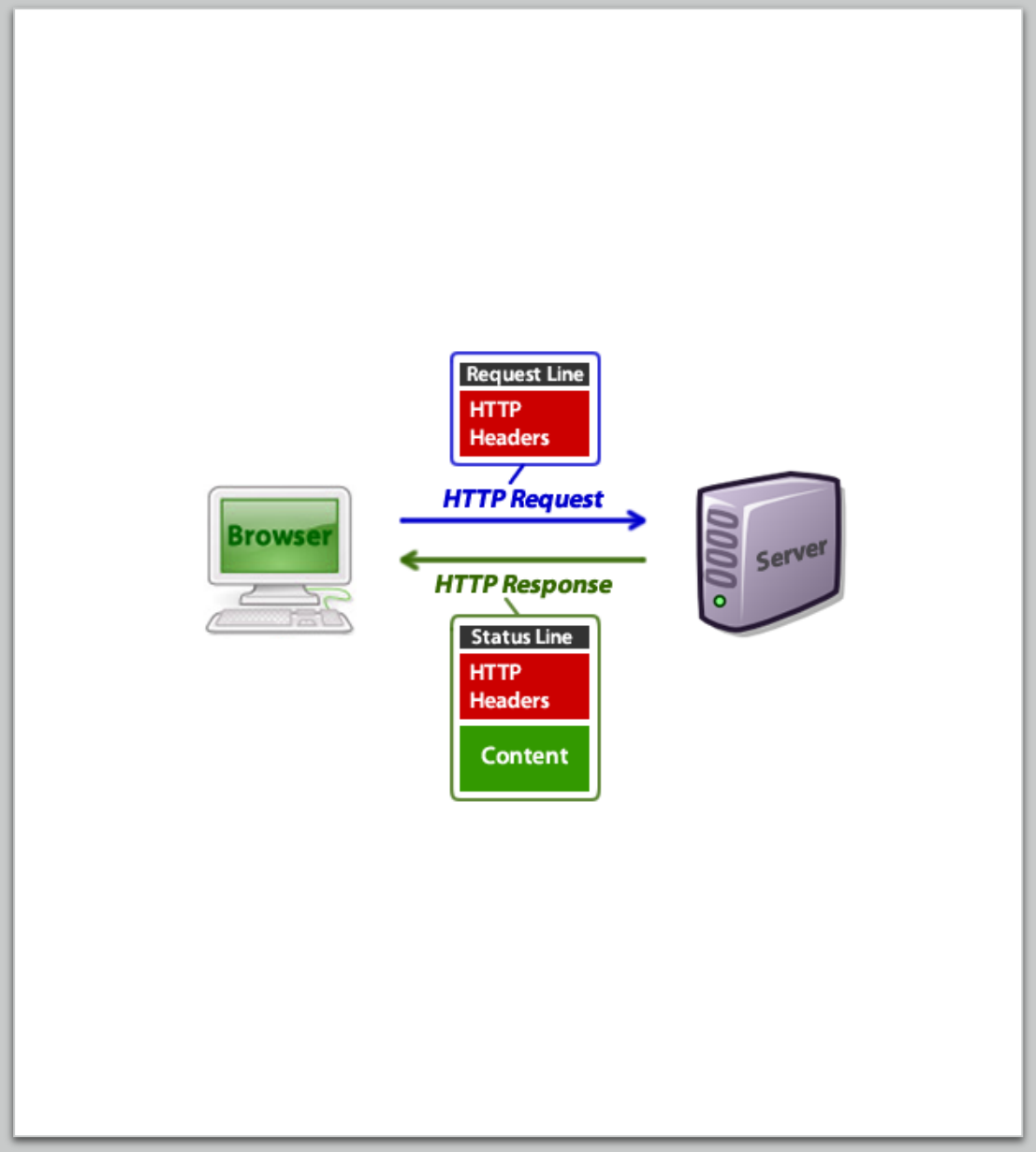
Examples:

- a GET request to /user/ returns a list of registered users on a system
- a POST request to /user/ creates a user with the ID 123 using the body data (see 4. below). The response returns the ID.
- a PUT request to /user/123 updates user 123 with the body data (see 4. below)
- a GET request to /user/123 returns the details of user 123
- a DELETE request to /user/123 deletes user 123

HTTP headers

HTTP headers. Information such as authentication tokens or cookies can be contained in the HTTP request header.

HTTP Headers are an important part of the API request and response as they **represent the meta-data associated with the API request and response**. Headers carry information for: Request and Response Body. Request Authorization.



HTTP Body

HTTP Body Data is the data bytes transmitted in an HTTP transaction message immediately following the headers if there is any (in the case of HTTP/0.9 no headers are transmitted).

Most HTTP requests are GET requests without bodies. However, simulating requests with bodies is important to properly stress the proxy code and to test various hooks working with such requests. Most HTTP requests with bodies use POST or PUT request method.

Message Body

The message body part is optional for an HTTP message but if it is available then it is used to carry the entity-body associated with the request or response. If entity body is associated then usually Content-Type and Content-Length headers lines specify the nature of the body associated.

A message body is the one which carries actual HTTP request data (including form data and uploaded etc.) and HTTP response data from the server (including files, images etc). Following is a simple content of a message body:

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

Structure of a Request

Every request requires some information to work:

Method: The method tells the server what the client wants it to do.

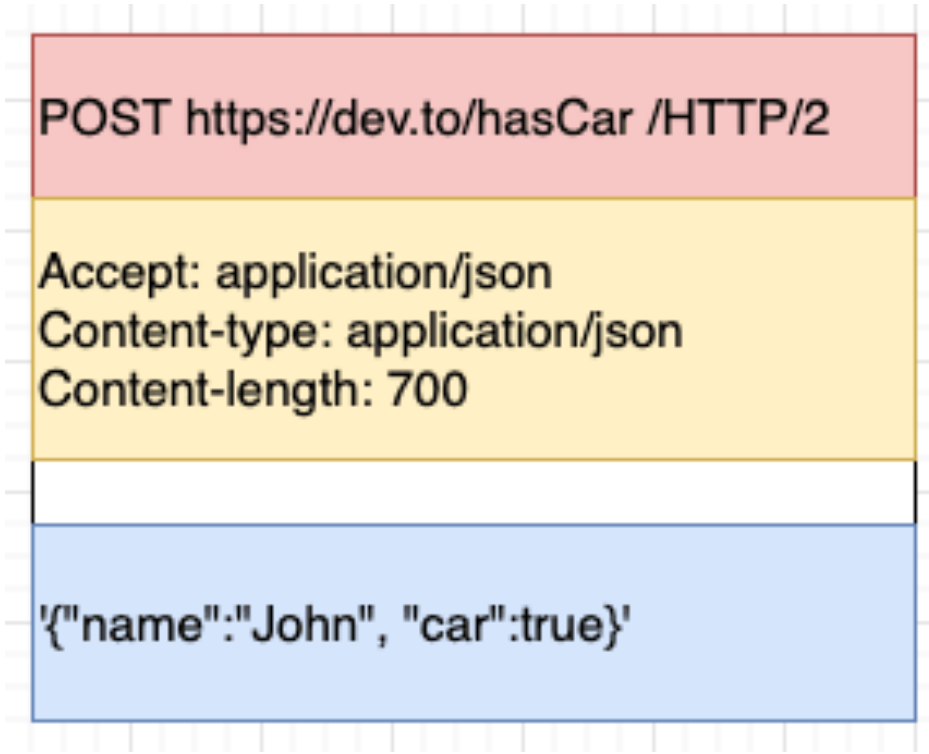
URL: The URL tells the HTTP tool where to send the request.

Protocol: Set by the HTTP tool used.

Headers: Headers give the server more information about the request itself.

The URL in the HTTP request message works just like when you type in a URL to go to a webpage in your browserThere is also an optional part:

Body: If a request uses a method that sends data to the server, the data is included in the body, right after the headers.



Structure of a Response

After a client sends an HTTP request, the server sends back an HTTP response. Every response sends back some information:

Protocol: Set by the HTTP tool being used.

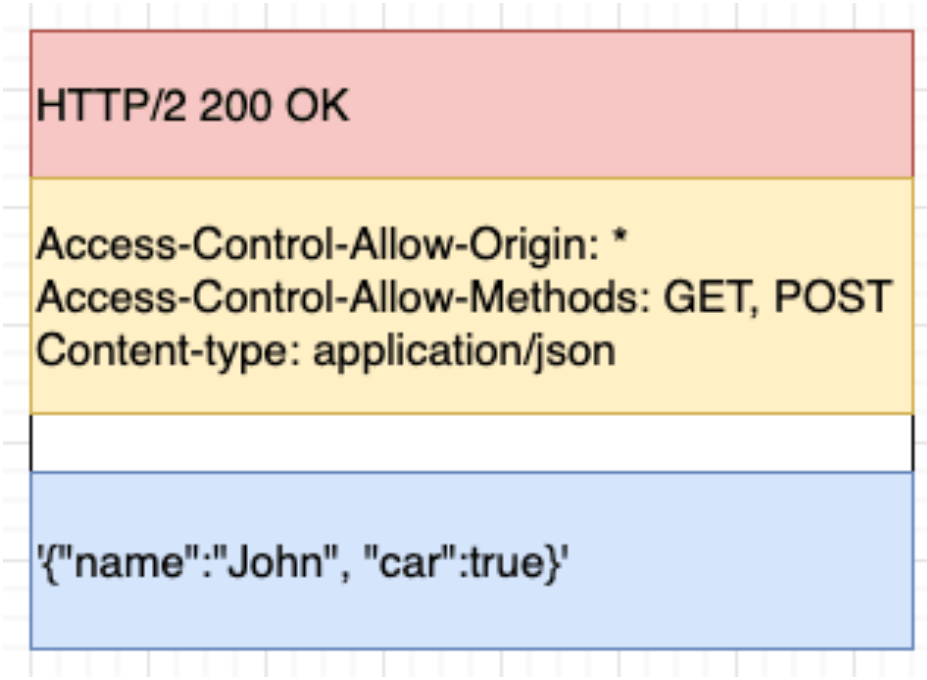
Status Code: A set of numbers that will tell you how the process from request to response went.

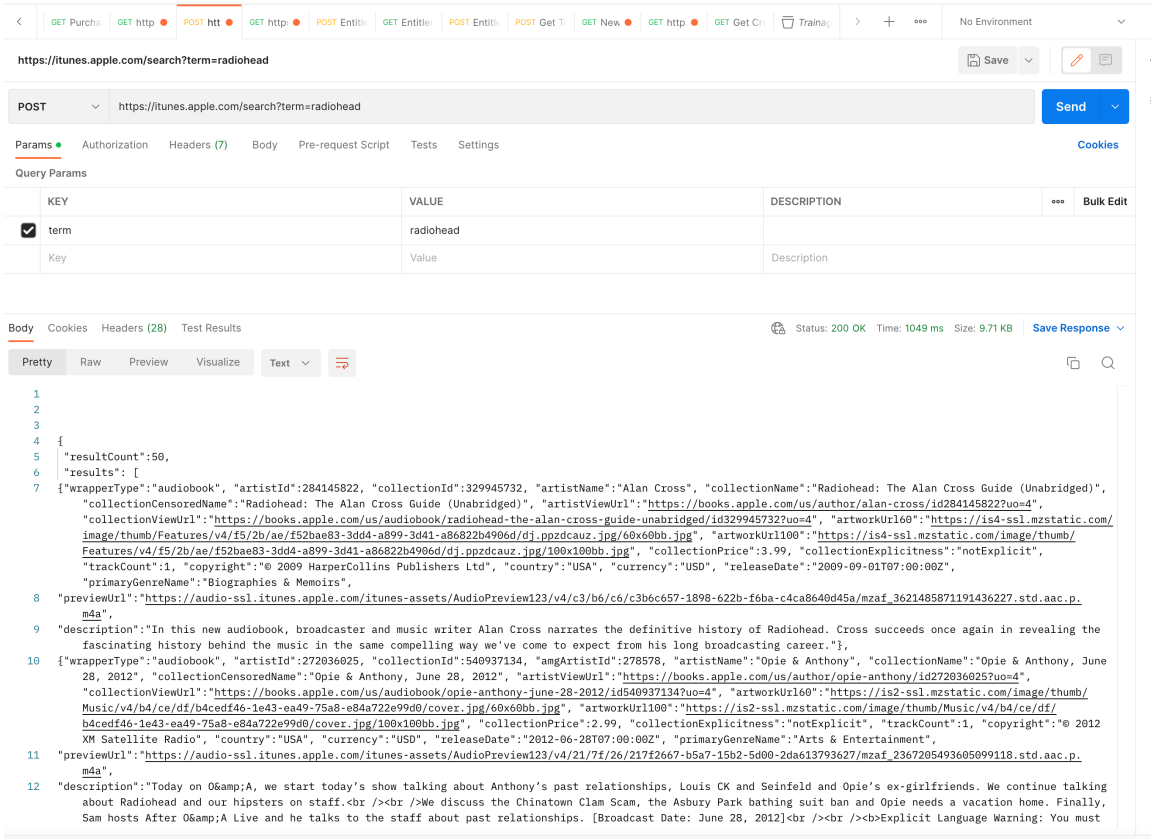
Status Message: A human readable description that will tell you how the process from request to response went.

Headers: Gives the client more information about the response itself.

There is also an optional part:

Body: If the response contains data from the server, it will be included here. Request and response bodies use the same formats.





Postman for requests

https://itunes.apple.com/search?term=radiohead

JSON - Introduction

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is a **text format** for storing and transporting data
- JSON is "self-describing" and easy to understand

```
{"name":"John", "age":30, "car":null}
```

It defines an object with 3 properties:

- name
- age
- car

Each property has a value.

What is JSON?

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is a lightweight data-interchange format
- JSON is plain text written in JavaScript object notation
- JSON is used to send data between computers
- JSON is language independent *

Why Use JSON?

The JSON format is syntactically similar to the code for creating JavaScript objects. Because of this, a JavaScript program can easily convert JSON data into JavaScript objects.

Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.

JavaScript has a built in function for converting JSON strings into JavaScript objects:

JSON.parse()

JavaScript also has a built in function for converting an object into a JSON string:

JSON.stringify()

JSON Syntax Rules

JSON syntax is derived from JavaScript object notation syntax:

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

JSON Data - A Name and a Value

JSON data is written as name/value pairs (aka key/value pairs).

A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value: `"name":"John"`

THANKS

UNIVERCITY OF GEORGIA

https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml

https://www.freecodecamp.org/news/what-happens-when-you-hit-url-in-your-browser/

https://www.w3schools.com