

1. Indicate which are the parts of the following url: <https://backend.mega-app.com.co:8080/api/articles/search?docid=1020&hl=en#dayone>

https://: A scheme. The scheme identifies the protocol to be used to access the resource on the Internet. It can be HTTP (without SSL) or HTTPS (with SSL).

- The scheme is followed by a colon and two forward slashes.

backend.mega-app.com.co: A host. The host name identifies the host that holds the resource. For example, www.example.com. A server provides services in the name of the host, but hosts and servers do not have a one-to-one mapping.

:8080: A Port. To identify the data associated with each process, port numbers are used. Port numbers are 16-bit, and numbers up to 65535 are possible, although in practice only a small subset of these numbers is commonly used.

/api/articles/search: A Path. The path identifies the specific resource in the host that the web client wants to access.

?docid=1020&hl=en#dayone: A query string. If a query string is used, it follows the path component, and provides a string of information that the resource can use for some purpose (for example, as parameters for a search or as data to be processed).

Figure 1. Syntax of an HTTP URL

```

                                .-:80-----
>>-http://--+-host name--+-----+---/--path component----->
                        '-IP address-'   '-:--port-'

>--+-----+-----><
    '-?--query string-'
```

2. Define what is a Web API, Restful and what are the statusCode 200-, 400-, 500-

Web API

API stands for Application Programming Interface. API is actually some kind of interface which is having a set of functions. These set of functions will allow programmers to acquire some specific features or the data of an application.

Web API is an API as the name suggests, it can be accessed over the web using the HTTP protocol. It is a framework that helps you to create and develop HTTP based RESTFUL services.

Restful

RESTful web services are built to work best on the Web. Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. In the REST architectural style, data and functionality are considered resources and are accessed using **Uniform Resource Identifiers (URIs)**, typically links on the Web. The resources are acted upon by using a set of simple, well-defined operations. The REST architectural style constrains an architecture to a client/server architecture and is designed to use a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol.

Status Code

200 Ok: The request has succeeded.

400 Bad request: The request could not be understood by the server due to malformed syntax. The client **SHOULD NOT** repeat the request without modifications.

500 Internal server error: The server encountered an unexpected condition which prevented it from fulfilling the request.

3. When we talk about **CRUD**, what does it mean?

The acronym **CRUD** refers to the major operations which are implemented by databases. Each letter in the acronym can be mapped to a standard Structured Query Language (SQL) statement

C	Create
R	Read
U	Update
D	Delete