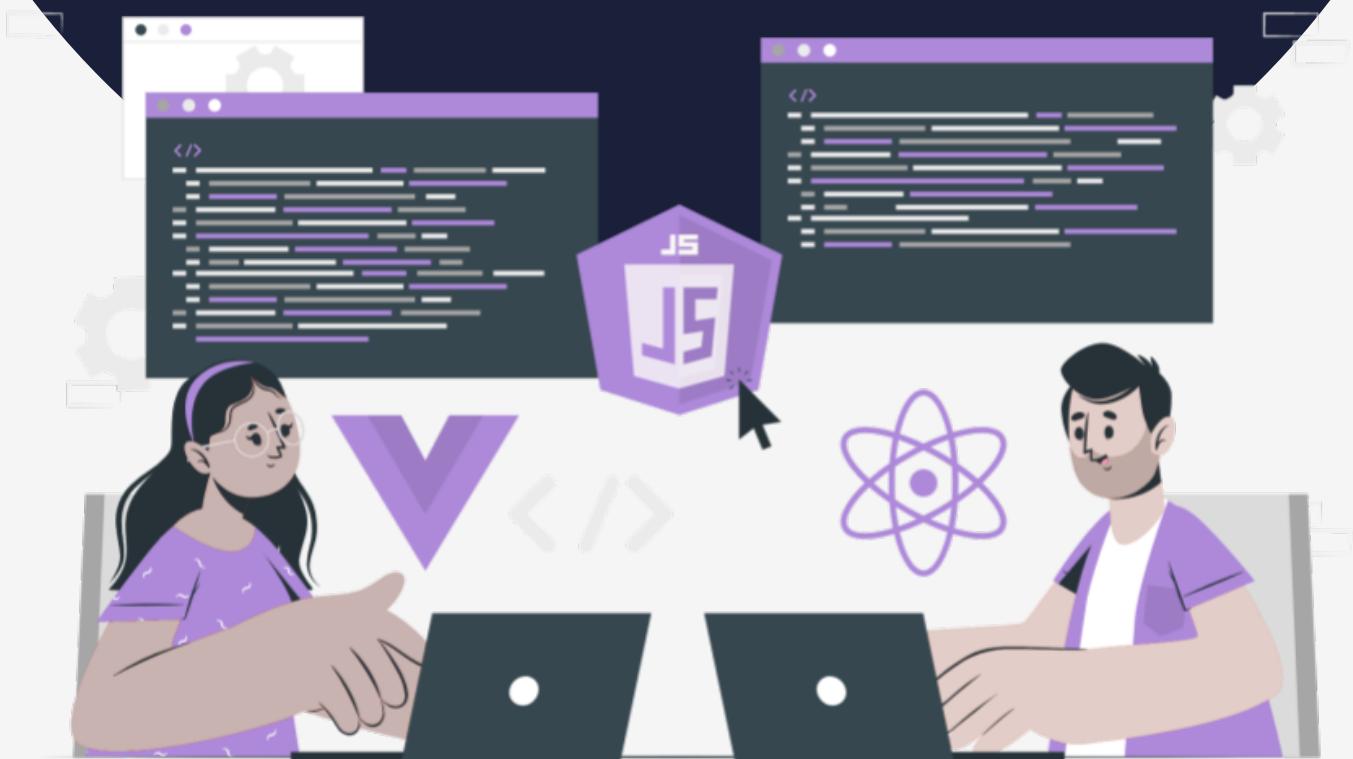


# Lesson:

# How to Handle Different URLs



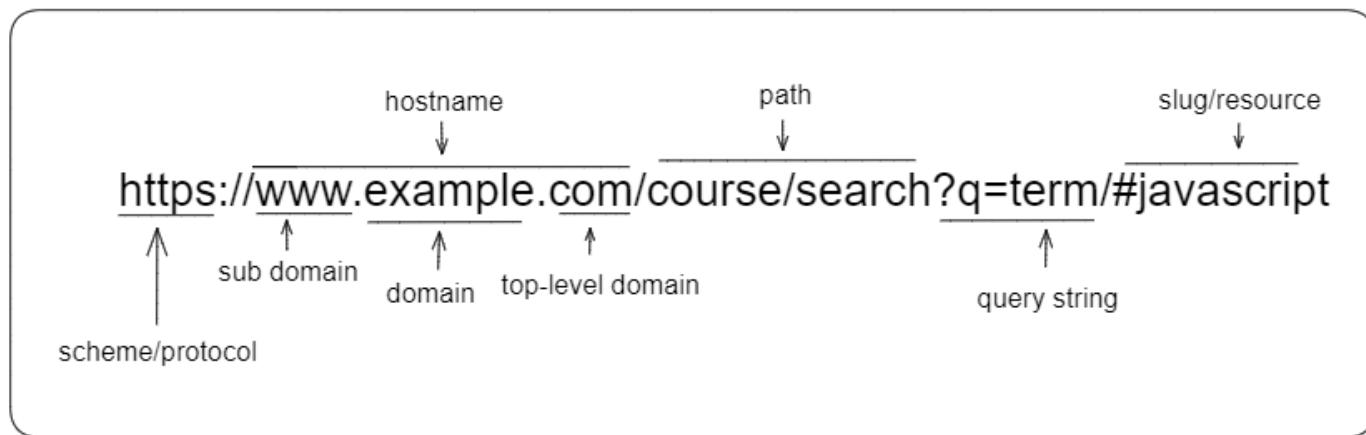
# Topics to be covered

1. What are URLs
2. What are the different types of URLs
3. How to handle different URLs in Node.js

## What are URLs

URLs stand for Uniform Resource Location, commonly termed as a web address, and is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A URL is a specific type of Uniform Resource Identifier (URI) although many people use the two terms interchangeably. URLs occur most commonly reference web pages (HTTP) but are also used for file transfer (FTP), email(mailto), database access, and much other application.

An URL comprises several parts, including the protocol (eg. HTTP or HTTPS), the domain name or IP address of the server hosting the resource, and the path to the resource on the server.



URLs are composed of several parts that help browsers and servers locate and retrieve the resource. Some of the main parts of a URL include:

1. Scheme – The scheme indicates the protocol used to access the resource it can be HTTP, HTTPS, FTP, and so on.
2. Hostname – the hostname is the name of the server hosting the resource, it consists of a domain, sub-domain, and top-level domain.
3. Port – the port number is an optional part of the URL that specifies the network port to use for the connection.
4. Path – it specifies the location of the resource on the server
5. Query – the query string is an optional part of the URL that contains data to be passed to the server.
6. Slug – it can be called a resource or fragment which is an optional part of the URL that identifies a specific portion of the resource to display.

# What are the different types of URLs

There are several types, each with its own unique format and purpose. Some of the common types of URLs include the following-

1. **HTTP URL:** A URL that uses HTTP (Hypertext Transfer Protocol) protocol to transfer data over the internet. This is the common type of URL used to access web pages and resources.

Example - <http://www.example.com>

2. **HTTPS URL:** A URL that uses the HTTPS (Hyper Transfer Protocol Secure) protocol to transfer data over the internet. HTTPS provides encryption and security features to protect data sensitive data, such as passwords and credit card numbers, from being intercepted by unauthorized parties.

Example - <https://www.example.com>

3. **FTP URL:** A URL that uses the FTP (File Transfer Protocol) protocol to transfer files over the internet. FTP is often used to upload and download large files, such as software updates or media files.

Example - <ftp://example.com>

4. **FTPS URL:** A URL that uses the FTPS(File Transfer Protocol Secure), it is the secure version of FTP

Example - <ftps://example.com>

5. **mailto URL:** A URL that launches an email client and populates the recipient, subject, and body fields of a new email message

Example - <mailto:user@example.com>

6. **Tel URL:** A URL that launches the phone app on a mobile device and dials a phone number

Example - <tel:1234567890>

would launch the phone app and dial the number  
"1234567890"

7. **File URL:** A URL that points to a local file on a computer or network.

Example - <file:///c:/Users/Document/example.pdf> would point to the example pdf.

A valid URL must therefore begin with either file:/path (no hostname), file:///path (empty hostname), or file://hostname/path

# How to handle different URLs in Node.js

In Node.js, different URLs can be handled in two ways i.e with the built-in module and a third-party package or library.

In the built-in module, the conditional statement is used i.e if-else and the switch statement is used to check the URL of the incoming request and handle each route accordingly.

## Example:

```

const http = require("http");

const port = 3000; // port declaration
const host = "localhost"; // host name declaration

// create server with http.createServer
const server = http.createServer((req, res) => {
  // conditional statement for handling route
  if (req.url === "/") {
    res.write("Hello, world!");
    res.end();
  } else if (req.url === "/about") {
    res.write("About page");
    res.end();
  } else if (req.url === "/contact") {
    res.write("Contact us");
    res.end();
  } else {
    res.write("Page not found");
    res.end();
  }
});
// server listen on port
server.listen(port, () => {
  console.log(`Server up at ${host}:${port}`);
});

// output -
http://localhost:3000/
hello world
http://localhost:3000/about
About page
http://localhost:3000/contact
Contact Us
http://localhost:3000/*
Page not found

```

Different URLs can also be handled using third party libraries or packages which we will look at further in the next topic, Express.