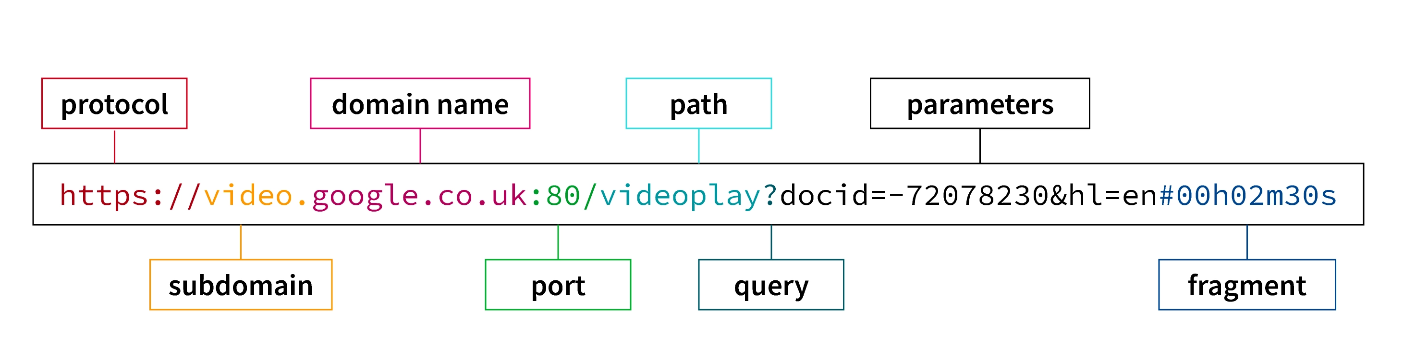
**URL module**

The URL module contains functions that help in parsing a URL. In other words, we can split the different parts of a URL easily with the help of utilities provided by the URL module.



The syntax for including the url module in your application:

**var url=require("url");**

The url.parse() method takes the url string as a parameter and parses it. The url module returns an object with each part of the url as property of the object.

**Syntax of url.parse() :**

url.parse(url\_string, parse\_query\_string, slashes\_host)

Description of the parameters :

* **url\_string :** <string> It is the URL string.
* **parse\_query\_string :** <boolean> It is a boolean value. By default, its value is false. If it is set to true, then the query string is also parsed into an object. Otherwise, the query string is returned as an unparsed string.
* **slashes\_host :** <boolean> It is a boolean value. By default, its value is false. If it is set to true, then the token in between // and first / is considered host.

var u=require("url");

var addr="http://localhost:8080/default.html?year=2025&month=feb";

var q=u.parse(addr,true);

console.log(q);

// console.log(q.host);

// console.log(q.pathname);

// console.log(q.search);

var qdata=q.query;

console.log(qdata.year);

if(qdata.year%4==0)

{

   console.log(“Its a leap year")

}

else{

   console.log("Its not a leap year")

}

=============================================================================

**Output:**

**Url {**

**protocol: 'http:',**

**slashes: true,**

**auth: null,**

**host: 'localhost:8080',**

**port: '8080',**

**hostname: 'localhost',**

**hash: null,**

**search: '?year=2025&month=feb',**

**query: [Object: null prototype] { year: '2025', month: 'feb' },**

**pathname: '/default.html',**

**path: '/default.html?year=2025&month=feb',**

**href: 'http://localhost:8080/default.html?year=2025&month=feb'**

**}**

**2025**

**Its not a leap year**

**Write a nodejs script to print query string of url on console as well as on file using ES6 callback.**

var u=require("url");

var ps=require("fs");

var adr1=" http://localhost:8080/default.html?year=2025&month=feb";

var q1=u.parse(adr1,true);

var qdata=q1.query;

console.log(qdata);

ps.writeFile("fsd2.txt",qdata.q,(err)=>

{

console.log("completed");

});

**HTTP Module**

HTTP module allows Node.js to transfer data over the Hyper Text Transfer Protocol (HTTP).

To include the HTTP module, use the require() method:

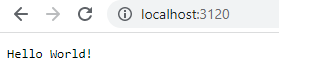
var http = require('http');

## **Node.js as a Web Server**

The HTTP module can create an HTTP server that listens to server ports and gives a response back to the client.

Use the createServer() method to create an HTTP server:

var http = require('http');  
  
//create a server object:  
http.createServer(function (req, res) {  
  res.write('Hello World!'); //write a response to the client  
  res.end(); //end the response  
}).listen(3020); //Server listening on port



## **Add an HTTP Header**

If the response from the HTTP server is supposed to be displayed as HTML, you should include an HTTP header with the correct content type:

var http = require('http');  
http.createServer(function (req, res) {  
**res.writeHead(200, {**'Content-Type'**:**'text/html'**});**  res.write('<h1>Hello World!</h1>');  
  res.end();  
}).listen(8180);



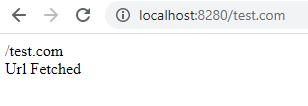
The first argument of the res.writeHead() method is the status code, 200 means that all is OK, the second argument is an object containing the response headers.

## **Read the Query String**

The function passed into the http.createServer() has a req argument that represents the request from the client, as an object (http.IncomingMessage object).

This object has a property called "url" which holds the part of the url that comes after the domain name:

var http = require('http');  
http.createServer(function (**req**, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write(**req.url**);  
  res.end();  
}).listen(8280);



## **Get the query string**

We can fetch the values from url query string as mentioned below using URL module.

1. Add static url in code and request server to display data of query string

var http = require('http');

var url = require('url');

var addr="http://localhost:8080/default.html?year=2024&month=feb";

http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/html'});

/\*Use the url module to get the querystring\*/

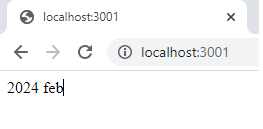
var q = url.parse(addr, true).query;

/\*Return the year and month from the query object:\*/

var txt = q.year + " " + q.month;

res.end(txt);

}).listen(3001);

****

1. Make changes in url at browser and request url to display data.

var http = require('http');

var url = require('url');

http.createServer(function (req, res) {

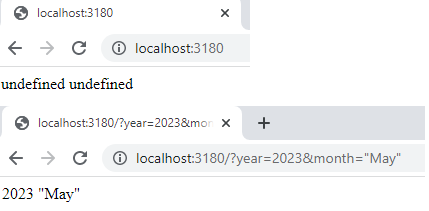
  res.writeHead(200, {'Content-Type': 'text/html'});

  var q = url.parse(req.url, true).query;

  var txt = q.year + " " + q.month;

  res.end(txt);

}).listen(3180);



**Write node js script to perform tasks as asked.**

1. **Create one page with two links (Home(/) and about(/about)).**
2. **Both pages must contain HTML type content and add required content on both the pages.**
3. **If user add any other URL path, then he/she will be redirected to page and plain message will be displayed of “Page not found”.**

var h=require("http");

var server=h.createServer(

    function(req,res)

    {

        if(req.url=="/")

        {

            res.writeHead(200,{"content-type":"text/html"});

            res.write("<h1> Home page </h1><div><ul><li><a href='/'>Home</a></li><li><a href='/about'>About</a></li></ul>");

            res.end();

        }

        else if(req.url=="/about")

        {

            res.writeHead(200,{"content-type":"text/html"});

            res.write("<h1> About Page </h1>");

            res.end();

        }

        else

        {

            res.writeHead(404,{"content-type":"text/plain"});

            res.write("Page not found");

            res.end("\nPlease check the url");

        /\* res.write("Bye");\*/ //display nothing if you add any content after res.end

        }

    });

server.listen(5051);

console.log("Thanks!");

**Write node js script to request server to display JSON data on browser**

var http=require("http");

var server=http.createServer(

    function(req,res)

    {

    if(req.url=="/")

    {

        const a={"Name":"ABC", "Age":35};

        res.writeHead(200,{"content-type":"application/json"});

        res.write("Thank you..!");

        res.write(JSON.stringify(a));

        res.end();

    }

});

server.listen(6001);

**Output:** Thank you..!{"Name":"ABC","Age":35}