

Node.js HTTP Module

< Previous

Next >

The Built-in HTTP Module

Node.js has a built-in module called HTTP, which 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:

Example

Get your own Node.js Server

```
var http = require('http');

//create a server object:
http.createServer(function (req, res) {
   res.write('Hello World!'); //write a response to the client
```



The function passed into the http://nethod.nill.nethod, will be executed when someone tries to access the computer on port 8080.

Save the code above in a file called "demo_http.js", and initiate the file:

```
Initiate demo_http.js:
    C:\Users\Your Name>node demo_http.js
```

If you have followed the same steps on your computer, you will see the same result as the example: http://localhost:8080

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:

Example

```
var http = require('http');
http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write('Hello World!');
    res.end();
}).listen(8080);
```

Run example »

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.



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:

```
demo_http_url.js

var http = require('http');
http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(req.url);
    res.end();
}).listen(8080);
```

Save the code above in a file called "demo_http_url.js" and initiate the file:

```
Initiate demo_http_url.js:
```

```
C:\Users\Your Name>node demo_http_url.js
```

If you have followed the same steps on your computer, you should see two different results when opening these two addresses:

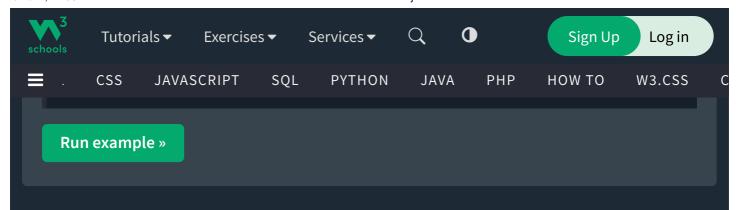
http://localhost:8080/summer

Will produce this result:

/summer

Run example »

http://localhost:8080/winter



Split the Query String

There are built-in modules to easily split the query string into readable parts, such as the URL module.

Example

Split the query string into readable parts:

```
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(8080);
```

Save the code above in a file called "demo_querystring.js" and initiate the file:

Initiate demo_querystring.js:

C:\Users\Your Name>node demo_querystring.js

The address:

http://localhost:8080/?year=2017&month=July