What are Node modules?

As you would have noticed, when we set up our Node server, we imported a **module** (the HTTP module) into our Node JavaScript file to help us.

What are modules?

You can think of modules as the same as JavaScript libraries. They give us a set of functions we want to include in our application.

The cool thing is that Node has a set of built-in modules that you can use without any further installation. To include a module, use the `require()` function with the name of the module:

HTTP module: `require('http')`

Node.js has a built-in module called HTTP, which allows Node.js to transfer data over the HyperText Transfer Protocol (HTTP).

As you have just seen in the example, we used the HTTP module to create an HTTP server that listens on port 8001 and gives a response back to the client (we used the `createServer()` method to create an HTTP server):

Querystring module: `require('querystring')`

In this example, we are **NOT** sending the user data as a query string (we will not send the input as key=value pairs).

Why not?

Because I want to show you that there are many ways in which we can achieve the same result when developing code.

BUT, assuming we had sent the data as a query string, we could have also used the built-in querystring module. Remember a few lectures back, PHP gave us access to the superglobal $_GET object? Well, you can think of the querystring module as giving us access to the same thing.

The Node querystring module provides methods for parsing and formatting URL query strings.

It has various methods and properties—such as decode() and encode()—but the one you will use in our example would be the `parse()` method. The `parse()` method is used to parse a URL query string into an object that contains the key and pair values of the query URL.

But yes, this is all theory.

Let’s get back to building our Node application.