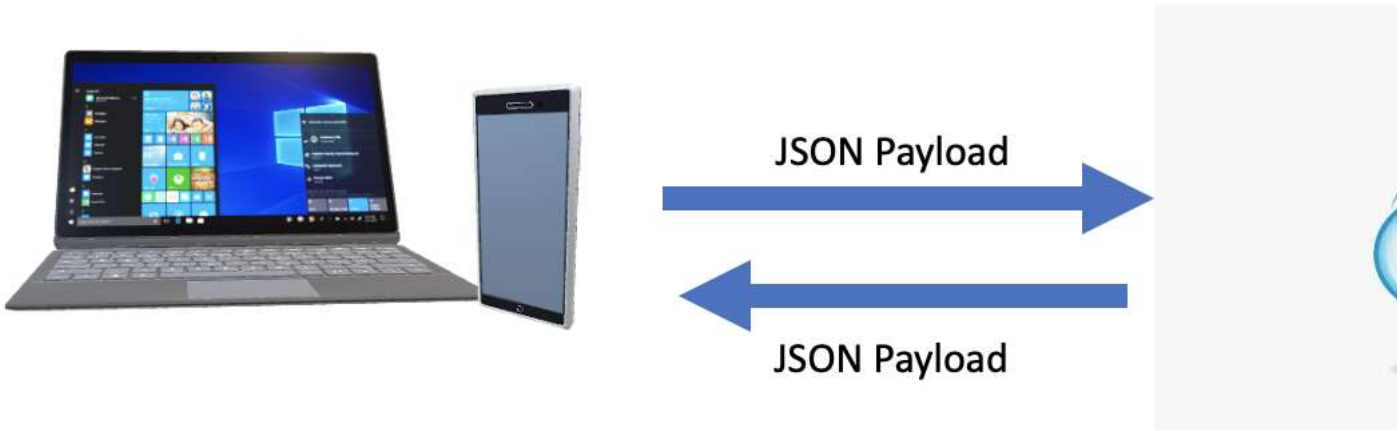
Getting started with Node.js

Skills Network

Full-stack application

When we talk about full-stack application, it includes:

- The client side
  - The website and the mobile application that are user facing
- The server side
  - Processes any request from the client side and sends a response to the client. In today's world, the cloud hosts the web server, application server, and database.



Open Source and cross-platform

Javascript is an ideal choice on the client side to perform validation of the HTML pages. Given the ease of use, Javascript was extended to be used with server-side coding. This is **Node.js**. Node.js is a run-time environment. You don't need any special licenses to use Node.js and there are many packages and libraries written for use with Node.js, as it open source. Node.js code, once written, can run on Linux, Windows, and Mac OSX.

V8 Engine

Any code that you write needs to be processed and converted to a machine-readable form. The Node.js Javascript code uses the Google V8 engine. V8 is Google's open source high-performance engine. All the Google Chrome browsers come with the V8 engine. Other modern browsers such as Microsoft Edge and Firefox Node.js also now use the V8 engine.

Event-driven, Asynchronous, Non-blocking, Single-Threaded

Processes in a server can be "single-threaded" or "multi-threaded." Single-threaded is where only one command is processed at a given point in time. Multi-threaded is where multiple commands are processed simultaneously. Node.js is single-threaded, which means it can only do one process at one time. That might make it sound like it

is not appropriate for server-side coding but Node.js is asynchronous and non-blocking. This means, while a process is being executed, the program doesn't have to wait until the process fininshes.

Node.js is event-driven. When Node.js performs an input/output (I/O) operation, like reading from the network, accessing a database or the file system, an event is triggered. Instead of blocking the thread and wasting the processor time waiting, Node.js will resume the operations when the response comes back or, in other words, the reseponse event occurs. During that time, the server is not blocked and can do other things, which makes it look like it is multi-threaded.

JSON Payload

JSON stands for JavaScript Object Notation. The JSON is formatted as "key-value pairs." The payload is the data transmitted between the client and the server. When the client needs to send data to the server, it sends it in the form a JSON object such as in the example below.

```
1. 1
2. 2
3. 3
4. 4
5. 5
```

```
1. {
2. "name":"John",
3. "age":"24",
4. "email":"johnparker@gmail.com"
5. }
```

Copied!

The above object is a JSON. When this data is sent as a part of the request, the values in it can be extracted simply by using, `request.name`and so on.

The response in also sent as JSON. An example is shown below.

```
1. 1
2. 2
3. 3
4. 4
```

```
1. {
2.     "status":"ok",
3.     "message":"Successfully added"
4. }
```

Copied!

Express Framework

While Node.js has packages to create a server, the Express framework makes it simple to create APIs and endpoints. An API endpoint is the specific point of entry for a request from client to the server.

Next steps

In this course you will learn how to do server-side coding with Node.js using the Express framework.