Node.js

Lesson 04—Working with Asynchronous Programming

© Simplilearn. All rights reserved.

Lesson Overview

In this lesson, you will be able to understand the difference between Synchronous and Asynchronous programming, work with callback function, Promises, and work with request module.

©Simplilearn. All rights reserved.

Learning Objectives

✓ Defining Asynchronous Programming

✓ Understanding the Callback Function

✓ Working with Promises

✓ Getting Familiar with Nested Promises

✓ Working with the Request Module

©Simplilearn. All rights reserved.

Working with Asynchronous Programming

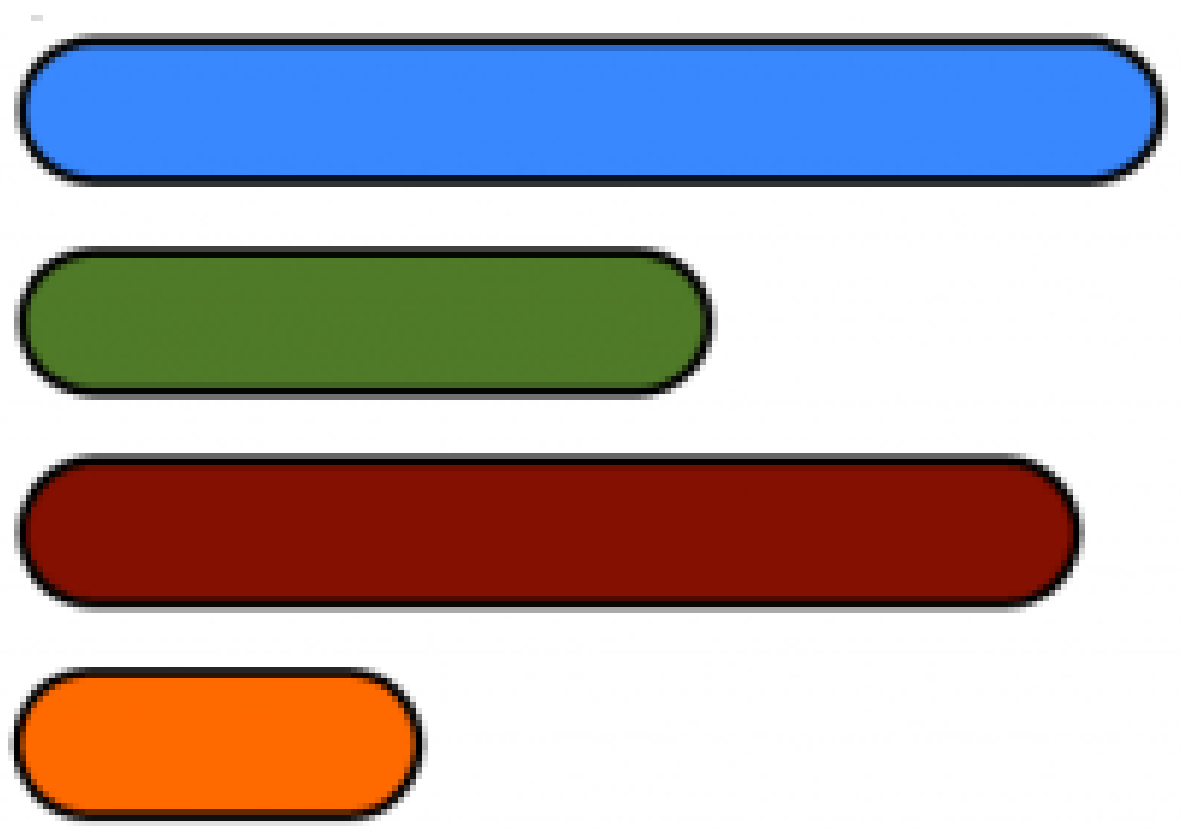
Topic 1—What is Asynchronous Programming?

©Simplilearn. All rights reserved.

What Is Asynchronous Programming?

INTRODUCTION

Asynchronous programming and Asynchronous actions are standard actions executed in a non-blocking fashion, which allows the main control flow of the program to continue the process.



©Simplilearn. All rights reserved.

What Is Asynchronous Programming?

FEATURES OF ASYNCHRONOUS PROGRAMMING

- Asynchronous programming allows Node developers to make callback events on a single thread. This not only improves the performance of an application but also reduces the input/output wait time between two requests.
- If you want to be effective at programming with Node, you must have a thorough understanding of the asynchronous model.
- For a functional programmer, it may be difficult to understand asynchronous programming. However, if a programmer has experience in multithreading, it is simple.
- To achieve Asynchronous model, we rely on callback functions.

©Simplilearn. All rights reserved.

What Is Asynchronous Programming?

ASYNCHRONOUS VS SYNCHRONOUS MODEL

Asynchronous Model

- Asynchronous programming enforces parallel programming.
- A group of tasks runs independently from the main thread and later notifies the calling thread about its status, failure or progress, which in turn improves application performance and responsiveness.

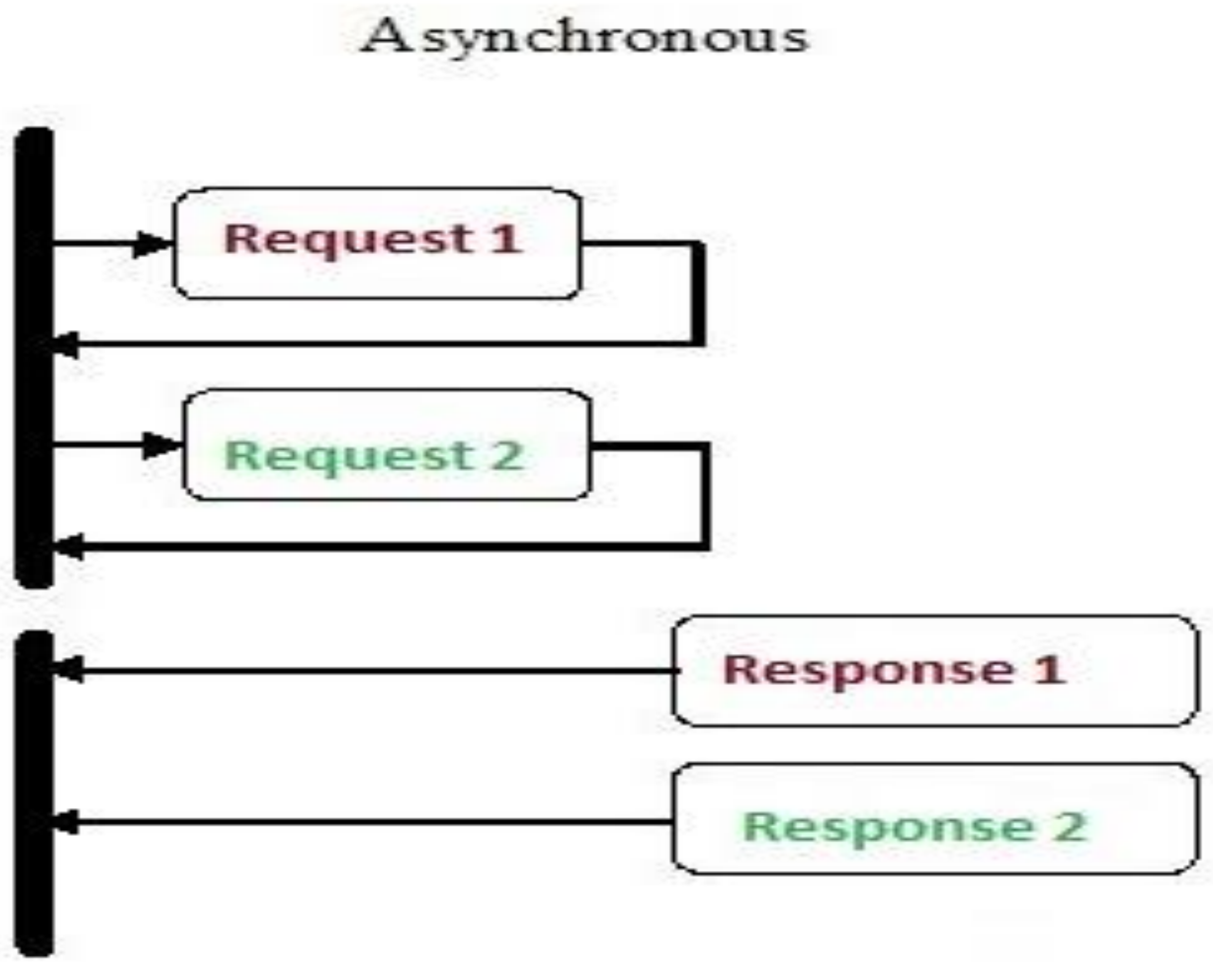
Synchronous Model

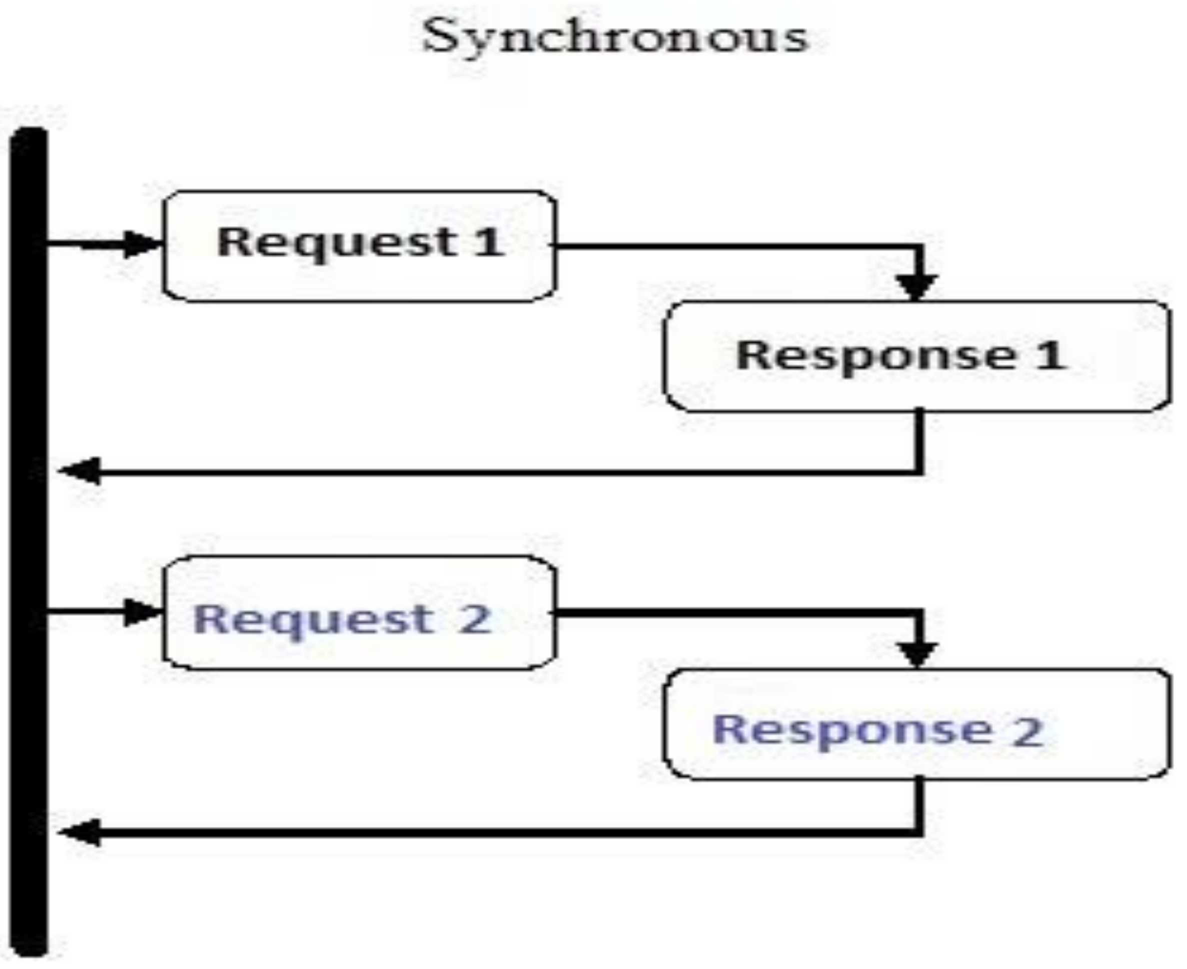
- In synchronous programming, the code is executed line by line (i.e., one line at a time).
- Each time only one function is executed, and program execution must wait until that current function returns before continuing to the next line of code.

©Simplilearn. All rights reserved.

What Is Asynchronous Programming?

ASYNCHRONOUS VS SYNCHRONOUS FLOW





©Simplilearn. All rights reserved.

Working with Asynchronous Programming

Topic 2—Callback Function

©Simplilearn. All rights reserved.

Callback Function

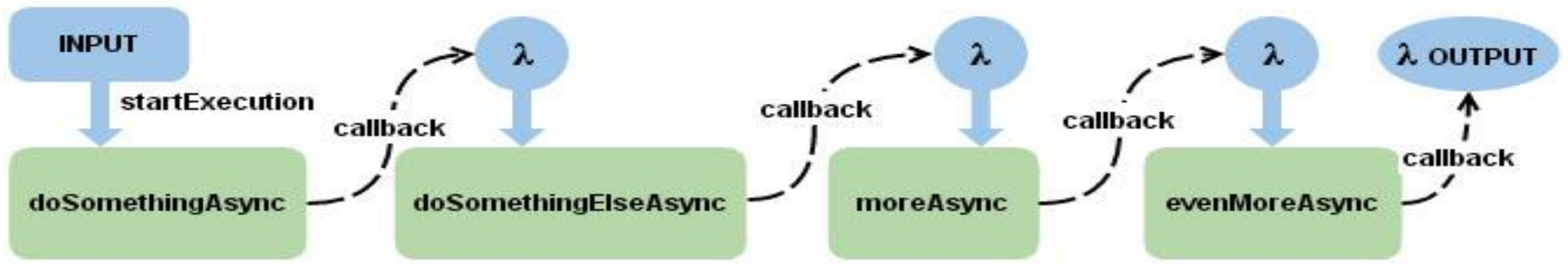
INTRODUCTION

Callback is a kind of asynchronous model for a function.

A callback function is called after the completion of every task.

In Node.js programming, callbacks are frequently used and hence all node modules are written to supports callbacks.

Using a callback function, you can pass a function (x) as a parameter to another function (y) and can call (x) within (y) when you finish your task.



©Simplilearn. All rights reserved.

Callback Function

EXAMPLES

Without Callback

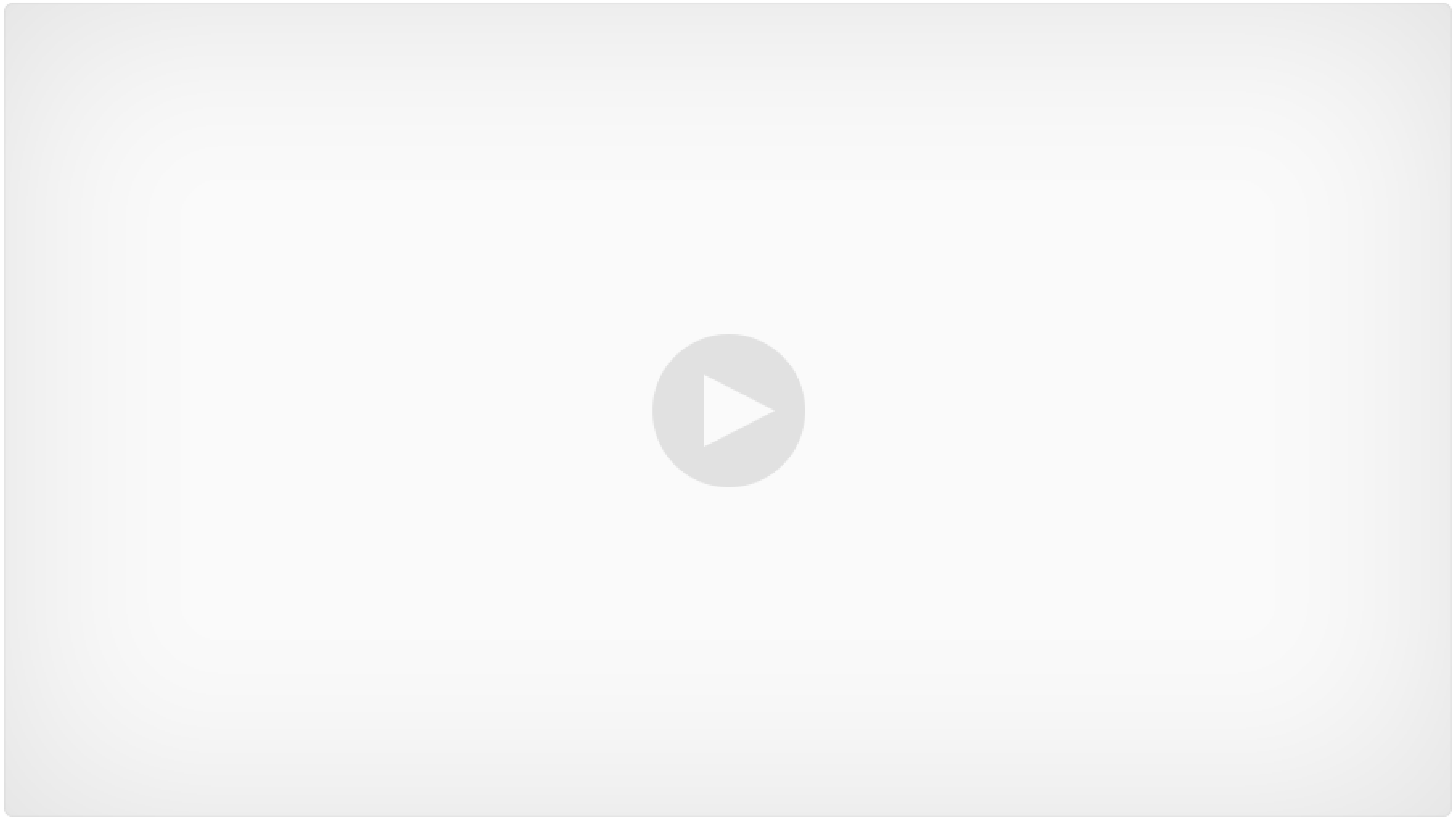
```
var fs = require("fs");
var userData =
fs.readFileSync(‘employee.txt');
console.log(userData.toString());
console.log("Program Finished");
```

With Callback

```
var fs = require("fs");
fs.readFile(employee.txt', function (err, data)
{
 if (err) return console.error(err);
 console.log(data.toString());
});
console.log("Program Finished");
```

©Simplilearn. All rights reserved.

Demo for Callback Function



©Simplilearn. All rights reserved.

Working with Asynchronous Programming

Topic 3—Promises

©Simplilearn. All rights reserved.

Promises

PROMISES IN A NUTSHELL

Imagine you are a **kid**. Your mom **promises** you that she'll get you a **new phone** next week.

You don't know if you will get that phone until next week. Your mom can either buy you a brand new phone or withhold the phone if she is not happy.

Three types of Promises

A promise that is **pending**: You don't know if you will get that phone until next week.

A promise that is **resolved**: Your mom buys you a brand new phone.

A promise that is **rejected**: You don't get a new phone because your mom is not happy.

©Simplilearn. All rights reserved.

Promises

CALLBACKS WITH PROMISES

- There may be a situation where we have to nest multiple callback functions together. This approach shall lead you to a difficult situation where the code would be hard to maintain, understand, and debug.
- A promise is an enhancement to callback functions that looks toward alleviating problems in Node.js.

Example:

```
var Promise = require('promise’);    ←------use the promise module
var MongoClient = require('mongodb').MongoClient;
var url = 'mongodb://localhost/EmployeeDB';
MongoClient.connect(url)
 .then(function(err, db) {    ←---the then function now available
 db.collection('Employee').updateOne({
  "EmployeeName": “Admin"
 }, {      block of code to update the employee
  $set: {
  "EmployeeName": “martin“}
```

©Simplilearn. All rights reserved.

Demo for Promises

©Simplilearn. All rights reserved.

Working with Asynchronous Programming

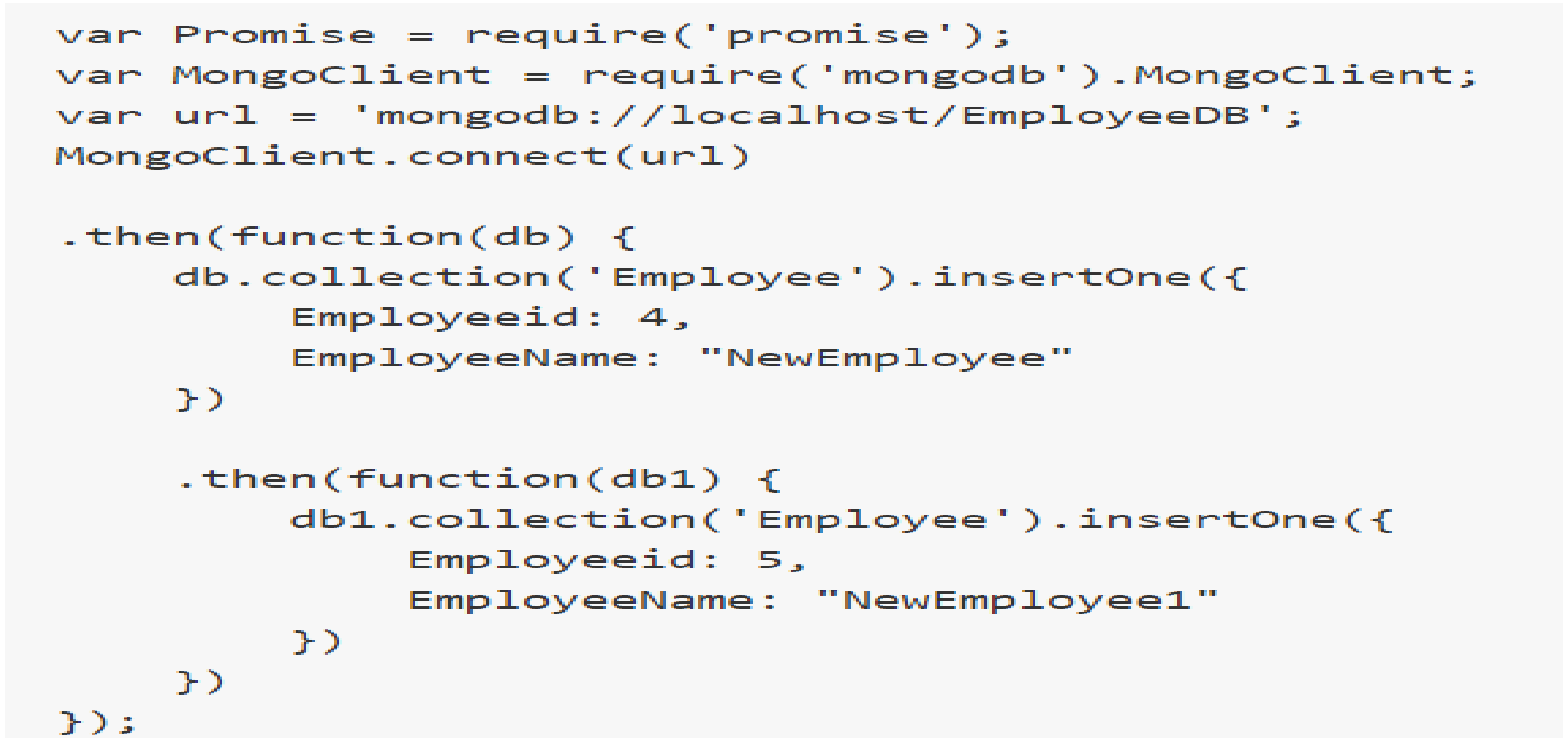
Topic 4—Nested Promises

©Simplilearn. All rights reserved.

Nested Promises

INTRODUCTION

When we define a promise, it is important to note that the "then" method returns a promise; therefore, promises can be nested or chained to each other.



```
var Promise = require('promise');
var MongoClient = require('mongodb').MongoClient;
var url = 'mongodb://localhost/EmployeeDB';
MongoClient.connect(url)

.then(function(db) {
    db.collection('Employee').insertOne({
        Employeeid: 4,
        EmployeeName: "NewEmployee"
    })

    .then(function(db1) {
        db1.collection('Employee').insertOne({
            Employeeid: 5,
            EmployeeName: "NewEmployee1"
        })
    })
});
```

©Simplilearn. All rights reserved.

Demo for Nested Promises

©Simplilearn. All rights reserved.

Working with Asynchronous Programming

Topic 5—Request Module API

©Simplilearn. All rights reserved.

Request Module API

INTRODUCTION

Request Module API

- Nowadays, web-based applications communicate with other services (REST service, Twitter, or images from Yahoo).
- Node is one of the popular platforms to handle many such applications.
- Users will be executing HTTP requests, which means the developer shall need a solid API to build a manageable code.
- Node.js “request” module is very famous among the developers for making HTTP requests.
- It’s a wrapper over Node's built-in HTTP module; therefore, you can achieve all features (GET, POST, PUT, DELETE).

©Simplilearn. All rights reserved.

Request Module API

EXAMPLE

```
var request = require('request');
request('http://www.google.com', function (error, response, body)
{
 console.log('error:', error); // Print the error if one occurred

console.log('statusCode:', response && response.statusCode); //Print the response status code if a response was re
ceived
 console.log('body:', body); // Print the HTML for the Google homepage.
});
```

Code Explanation:

url: URL of the HTTP request
method: method to be used (GET, POST, PUT, DELETE)

©Simplilearn. All rights reserved.

Demo for Request Module API

©Simplilearn. All rights reserved.

Quiz

©Simplilearn. All rights reserved.

QUIZ 1

Node.js by default supports ____ model.

a. Synchronous model

b. Asynchronous model

QUIZ 1

Node.js by default supports ____ model.

a. Synchronous model

b. Asynchronous model

The correct answer is **b. Asynchronous model.**

Node.js by default supports asynchronous model.

©Simplilearn. All rights reserved.

QUIZ

2

Which are the options Node.js uses to support asynchronous programming?

a. Callback

b. Promise

c. Observables

d. All of the above

©Simplilearn. All rights reserved.

QUIZ
2

Which are the options that Node.js uses to support asynchronous programming?

a. Callback

b. Promise

c. Observables

d. All of the above

The correct answer is **d. All of the above.**

Callback, promise, and observables are the options that Node.js uses to support asynchronous programming.

©Simplilearn. All rights reserved.

QUIZ

3

Which method is valid for http?

a. get

b. post

c. put

d. All of the above

©Simplilearn. All rights reserved.

QUIZ
3

Which method is valid for http?

a. get

b. post

c. put

d. All of the above

The correct answer is **d. All of the above.**

Get, post, and put are the methods valid for http.

©Simplilearn. All rights reserved.

QUIZ
4

Which of the following is true about RESTful webservices?

a. Web services based on REST Architecture are known as RESTful web services

b. Web services use HTTP methods to implement the concept of REST architecture

c. Both A & B

d. None of the above

©Simplilearn. All rights reserved.

QUIZ 4

Which of the following is true about RESTful web services?

a. Web services based on REST Architecture are known as RESTful web services

b. Web services use HTTP methods to implement the concept of REST architecture

c. Both A & B

d. None of the above

The correct answer is **c. Both A & B.**

Web services based on REST Architecture are known as RESTful web services, and web services use HTTP methods to implement the concept of REST architecture.

©Simplilearn. All rights reserved.

Key Takeaways

- ✓ Asynchronous programming and Asynchronous actions are standard actions executed in a non-blocking fashion, which allows the main control flow of the program to continue the process.
- ✓ Callback is a kind of asynchronous model for a function which is called after the completion of every task.
- ✓ A promise is an enhancement to callback functions in Node.js.
- ✓ It is important to note that the "then" method returns a promise and therefore promises can be nested or chained to each other.
- ✓ Using Request Module API, you can achieve all features (GET, POST, PUT, DELETE) with ease in Node.js.

©Simplilearn. All rights reserved.

Thank You