**Node.js JavaScript Promises**

JavaScript is a single-threaded programming language - that means only one thing can happen at a time.

Before ES6, we used callbacks to handle asynchronous tasks such as network requests.

Using promises, we can avoid the infamous ‘callback hell’ and make our code cleaner, easier to read, and easier to understand.

**Promises**

A promise is an object that holds the future value of an async operation.

For example, if we are requesting some data from a server, the promise promises us to get that data which we can use in the future.

**States of a Promise**

A Promise in JavaScript just like in the real-world has 3 states.

It can be;

1. unresolved (pending),

2. resolved (fulfilled), or

3. rejected

Explanation:

> Unresolved or Pending: A Promise is pending if the result is not ready. That is, it’s waiting for something to finish (for example, an async operation).

> Resolved or Fulfilled: A Promise is resolved if the result is available. That is, something finished (for example, an async operation) and all went well.

> Rejected: A Promise is rejected if an error happened.

**A Promise has two parts;**

1. Promise Creation

2. Consuming a Promise

Most of the times you will be consuming promises rather than creating them, but it’s still important to know how to create them.

Compiled and presented by Vakindu Philliam.